```
main_function('strings4.wav',1,'IIR',1,1,1,2,2,2,3,3,3);
main_function('strings4.wav',1,'FIR Window',1,1,1,2,2,2,3,3,3);
main_function('strings4.wav',2,'FIR Window',1,1,1,2,2,2,3,3,3);
main_function('strings4.wav',0.5,'FIR Window',1,1,1,2,2,2,3,3,3);
main_function('strings4.wav',1,'FIR Window',12,0,0,0,0,0,0,0,0);
function composite_signal =
main_function(file_name,choose_rate,filter_type,gain_1,gain_2,gain_3,gain_4,gain_5,gain_6,gain_7,
gain_8,gain_9)
% Read wav File
%file_name = 'strings4.wav';
[x,Fs]=audioread(file_name);

% Choose rate value
 switch choose_rate
     case 1
        rate = Fs;
     case 2
        rate = (Fs)*2;
     case 0.5
        rate = (Fs)/2;
     otherwise
  end

% play Sound
sound(x, rate);

% plot Frequency Domain for original signal
figure;
Nfft = length(x);
f=linspace(0, Fs, Nfft);
G=abs(fft(x, Nfft));
plot(f(1:Nfft/2), G(1:Nfft/2));

% Plot Time Domain for original signal
figure;
t = linspace(0,length(x)/Fs,length(x));
plot(t,x);

switch filter_type
    case 'IIR'
       band_1 = IIR_ButterWorth(Fs,0,170);
       band_2 = IIR_ButterWorth(Fs,170,310);
       band_3 = IIR_ButterWorth(Fs,310,600);
       band_4 = IIR_ButterWorth(Fs,600,1000);
       band_5 = IIR_ButterWorth(Fs,1000,3000);
       band_6 = IIR_ButterWorth(Fs,3000,6000);
       band_7 = IIR_ButterWorth(Fs,6000,12000);
       band_8 = IIR_ButterWorth(Fs,12000,14000);
       band_9 = IIR_ButterWorth(Fs,14000,16000);
     case 'FIR Window'
       band_1 = FIR_Window(Fs,0,170);
       band_2 = FIR_Window(Fs,170,310);
```

```
        band_3 = FIR_Window(Fs,310,600);
        band_4 = FIR_Window(Fs,600,1000);
        band_5 = FIR_Window(Fs,1000,3000);
        band_6 = FIR_Window(Fs,3000,6000);
        band_7 = FIR_Window(Fs,6000,12000);
        band_8 = FIR_Window(Fs,12000,14000);
        band_9 = FIR_Window(Fs,14000,16000);
    otherwise
 end


% filter using digital filters generating 9 signal of 9 bands
signal_1=filter(band_1, x);
signal_2=filter(band_2, x);
signal_3=filter(band_3, x);
signal_4=filter(band_4, x);
signal_5=filter(band_5, x);
signal_6=filter(band_6, x);
signal_7=filter(band_7, x);
signal_8=filter(band_8, x);
signal_9=filter(band_9, x);

% Amplify the output signals using the user-defined gain
% Changing from db to magnitude
signal_1=signal_1*(10^(gain_1/20));
signal_2=signal_2*(10^(gain_2/20));
signal_3=signal_3*(10^(gain_3/20));
signal_4=signal_4*(10^(gain_4/20));
signal_5=signal_5*(10^(gain_5/20));
signal_6=signal_6*(10^(gain_6/20));
signal_7=signal_7*(10^(gain_7/20));
signal_8=signal_8*(10^(gain_8/20));
signal_9=signal_9*(10^(gain_9/20));

% Add the amplified-output signals in time domain to form composite signal.

composite_signal = signal_1 + signal_2 + signal_3 + signal_4 + signal_5 + signal_6 + signal_7 +
signal_8 + signal_9;

% time domain for composite signal
figure;
t = linspace(0,length(composite_signal)/Fs,length(composite_signal));
plot(t,composite_signal);
% frequency domain for composite signal
figure;
Nfft = length(composite_signal);
f=linspace(0, Fs, Nfft);
G=abs(fft(composite_signal, Nfft));
plot(f(1:Nfft/2), G(1:Nfft/2));
% play composite signal
sound(composite_signal, rate);
% save composite signal
audiowrite('file.wav',composite_signal,Fs);
```

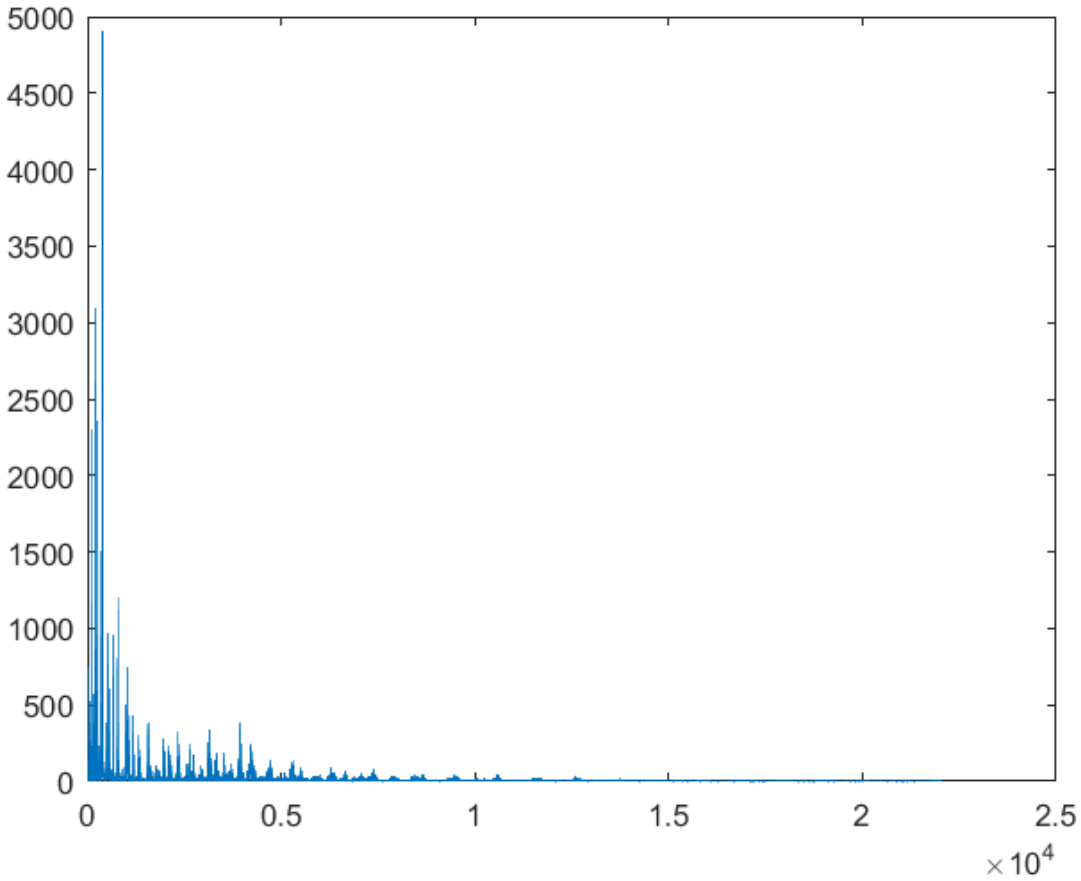
```
end
```

```
Warning: Data clipped when writing file.
Warning: Data clipped when writing file.
Warning: Data clipped when writing file.
Warning: Data clipped when writing file.
```

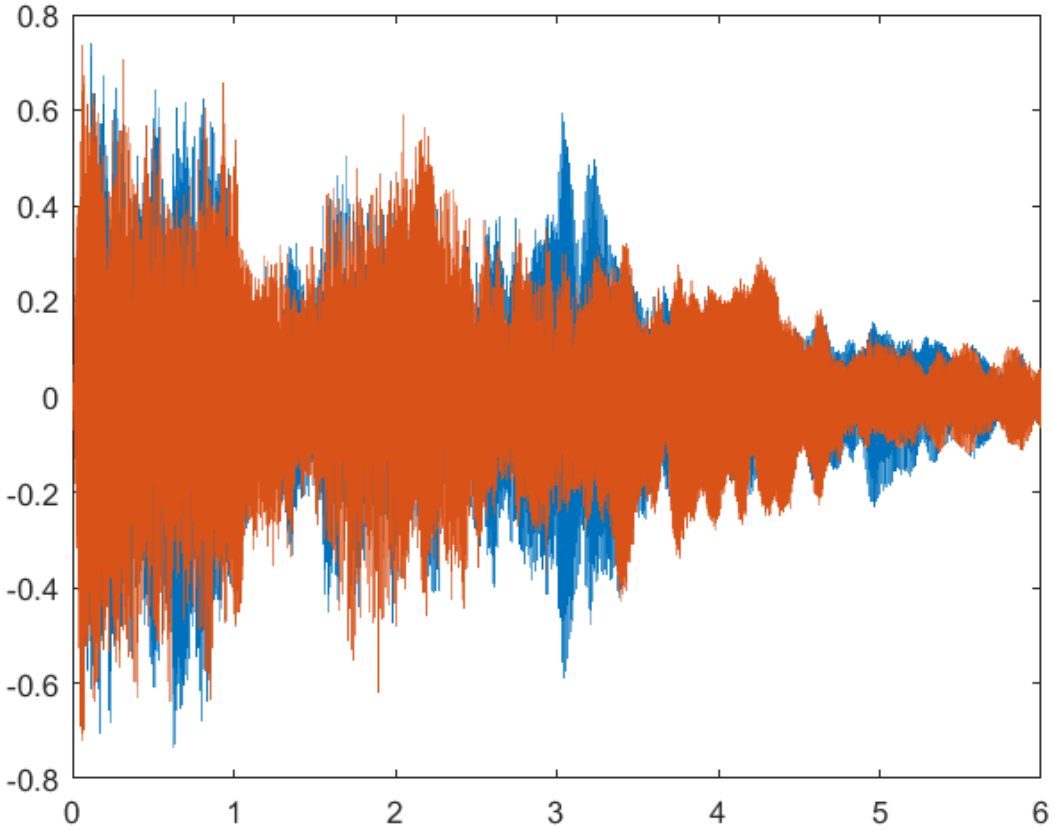
IIR Filter sample run:

```
main_function('strings4.wav',1,'IIR',1,1,1,2,2,2,3,3,3);
```

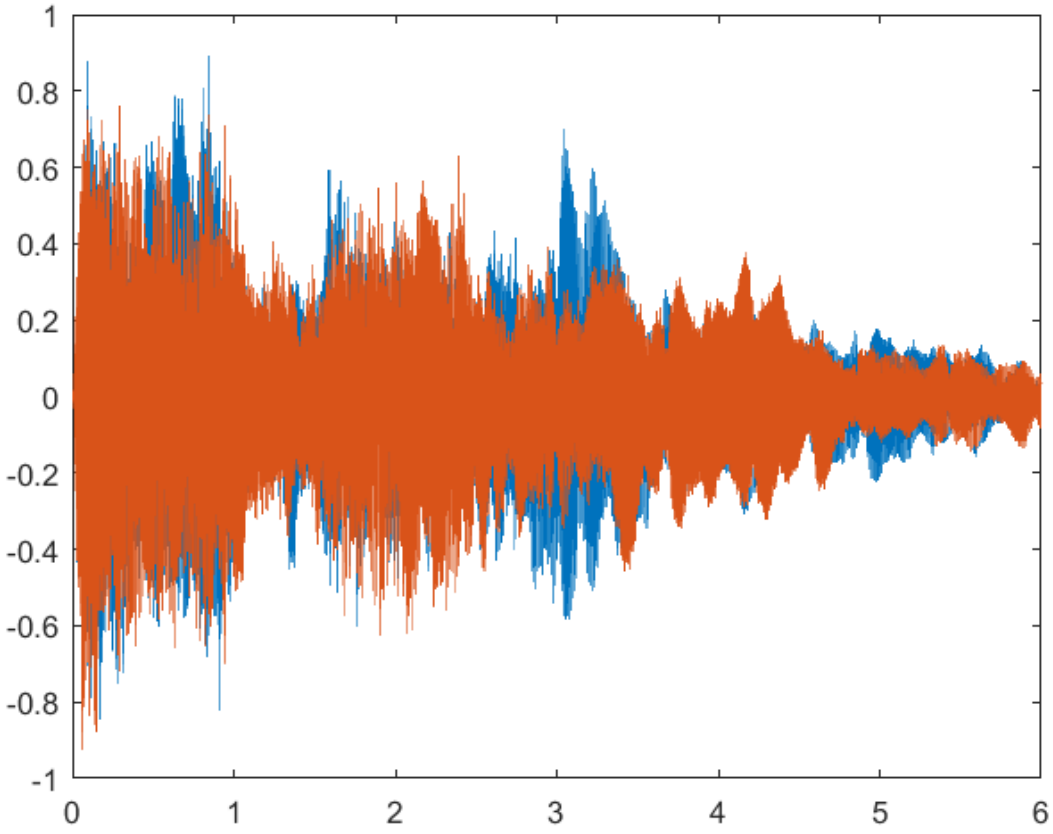
Frequency domain for Original signal



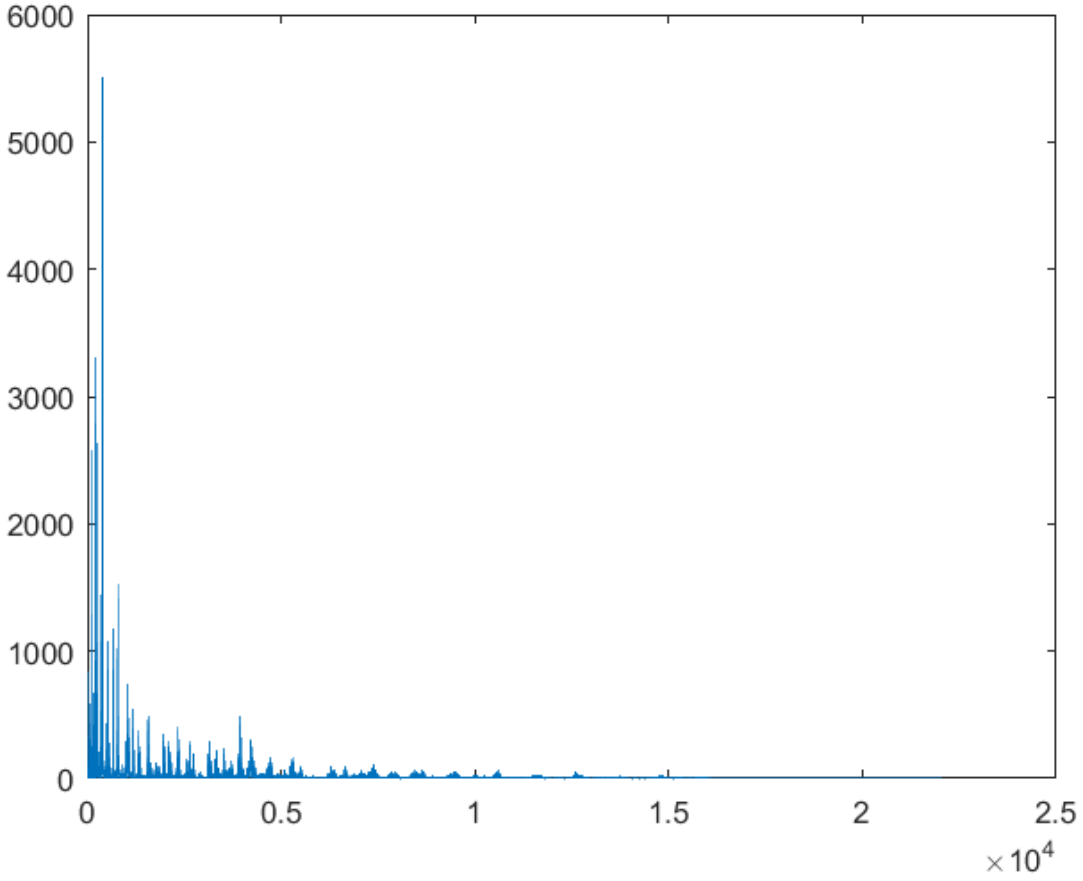
Time domain for Original signal



Time domain for Composite signal



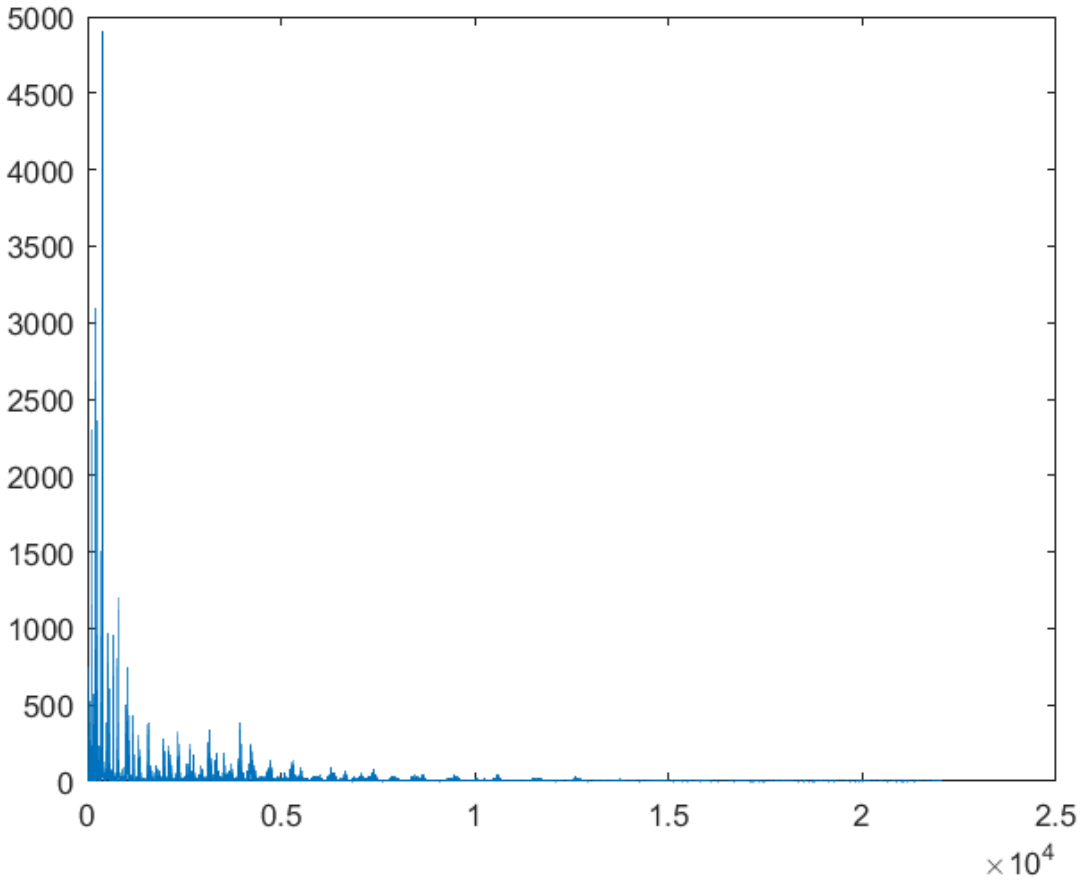
Frequency domain for Composite signal



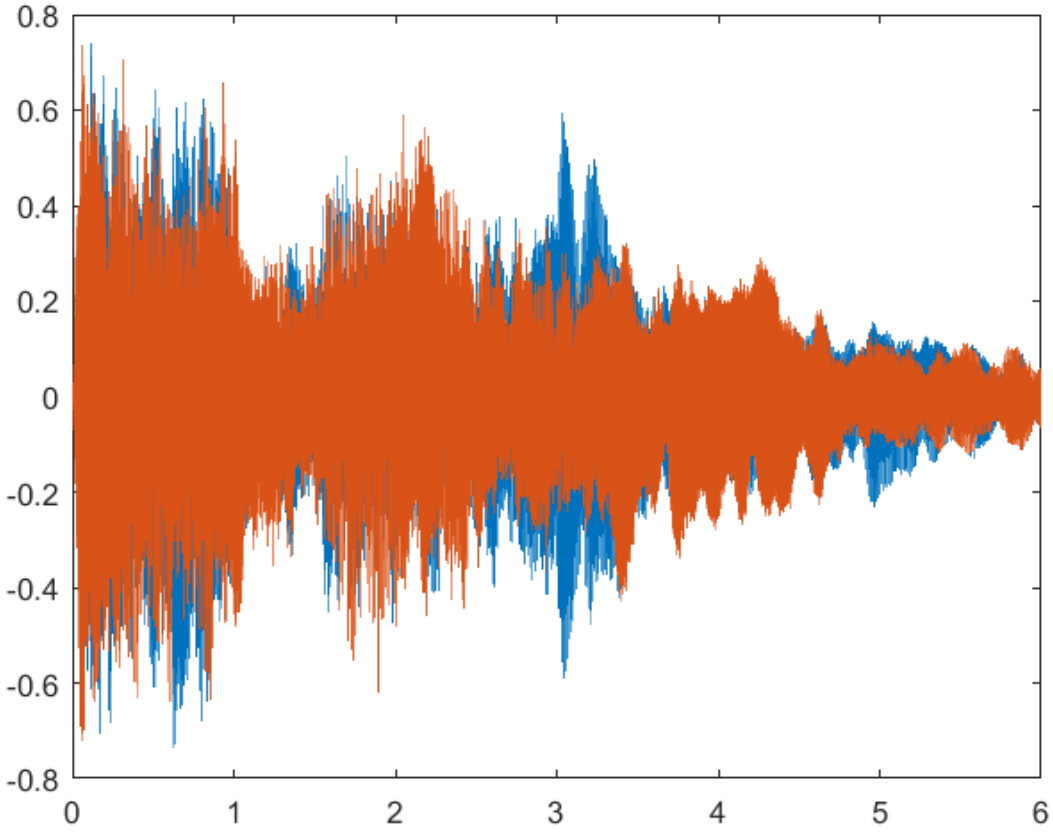
FIR Filter sample run:

```
main_function('strings4.wav',1,'FIR Window',1,1,1,2,2,2,3,3,3);
```

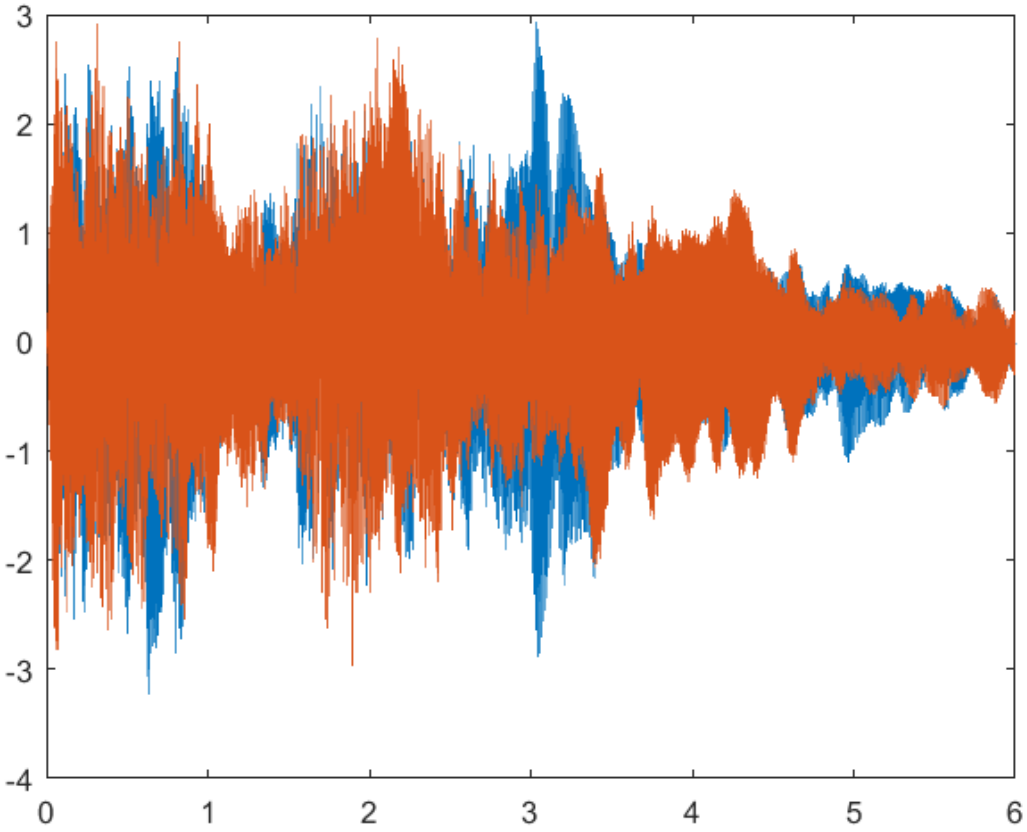
Frequency domain for Original signal



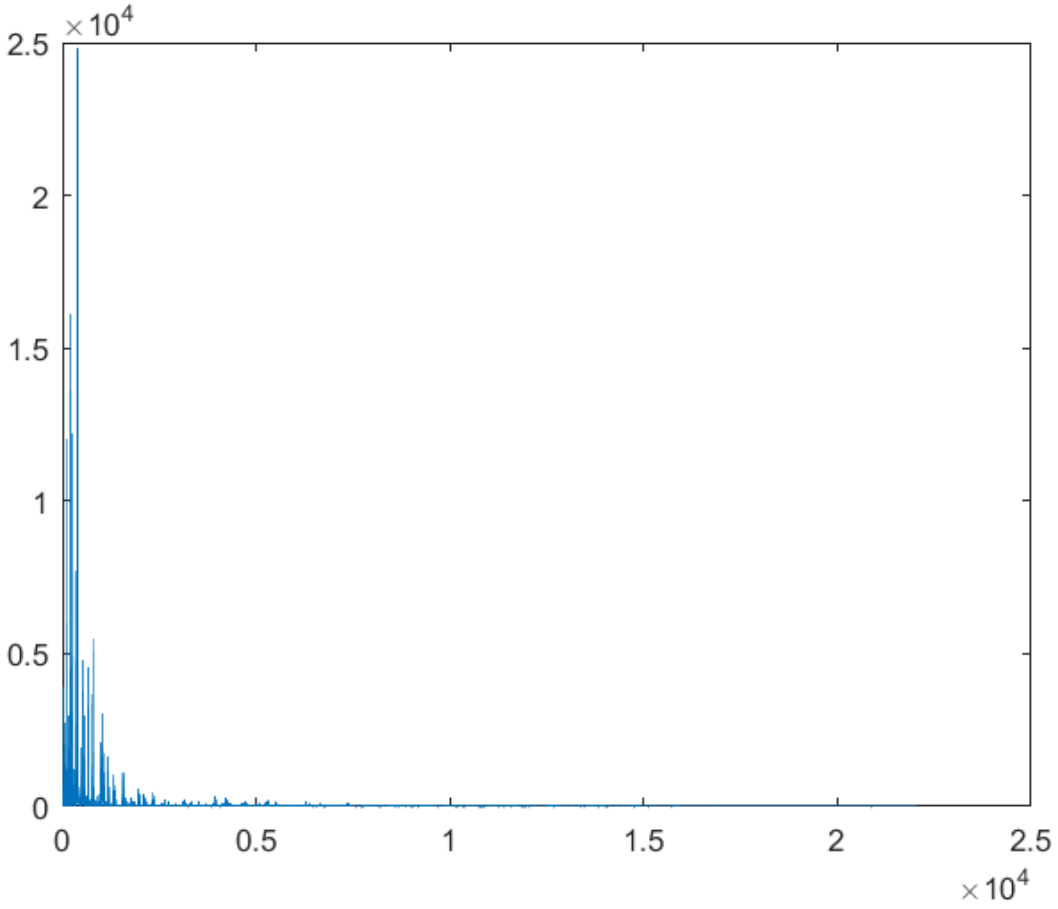
Time domain for Composite signal



Time domain for Composite signal



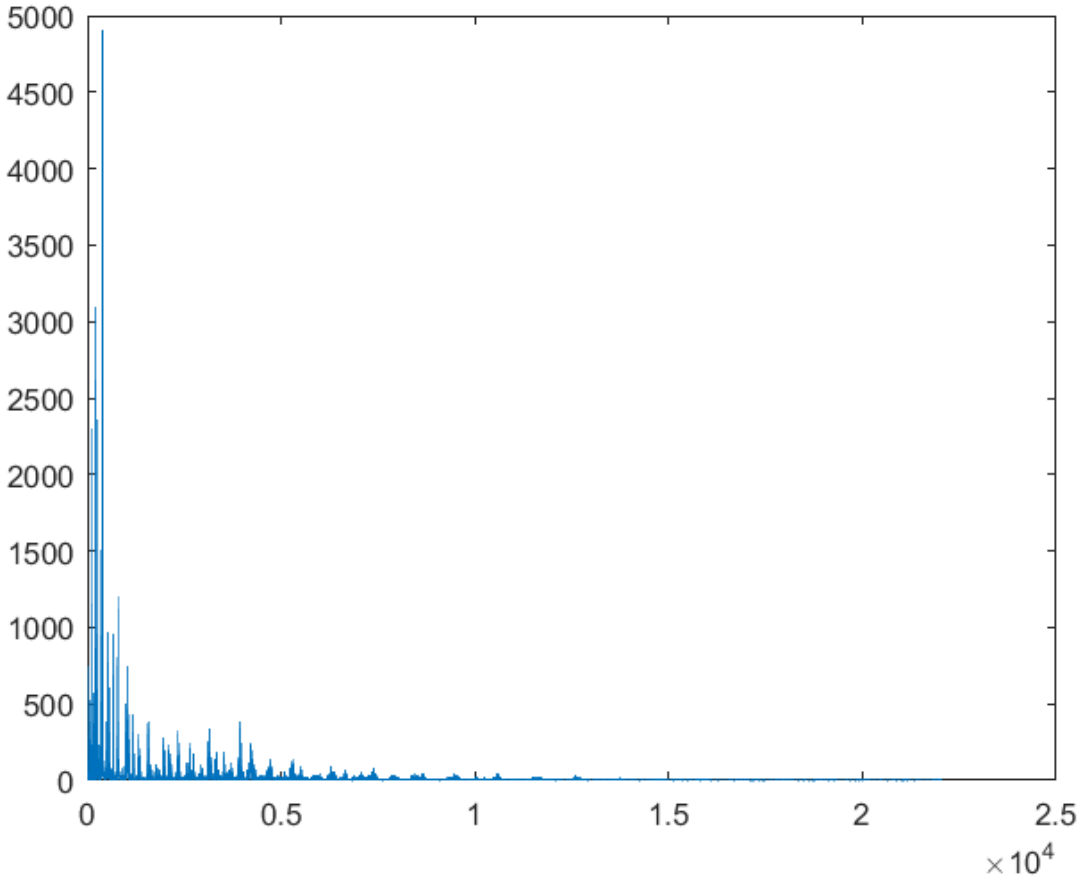
Frequency domain for Composite signal



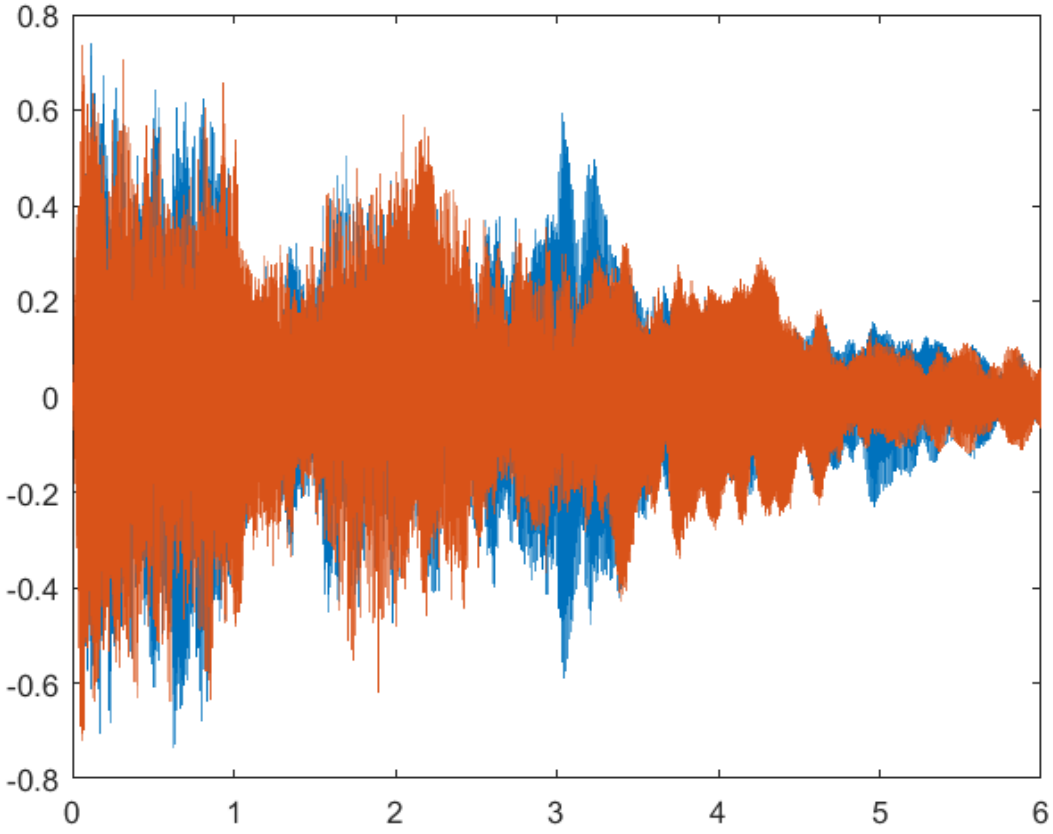
Double rate FIR sample run

```
main_function('strings4.wav',2,'FIR Window',1,1,1,2,2,2,3,3,3);
```

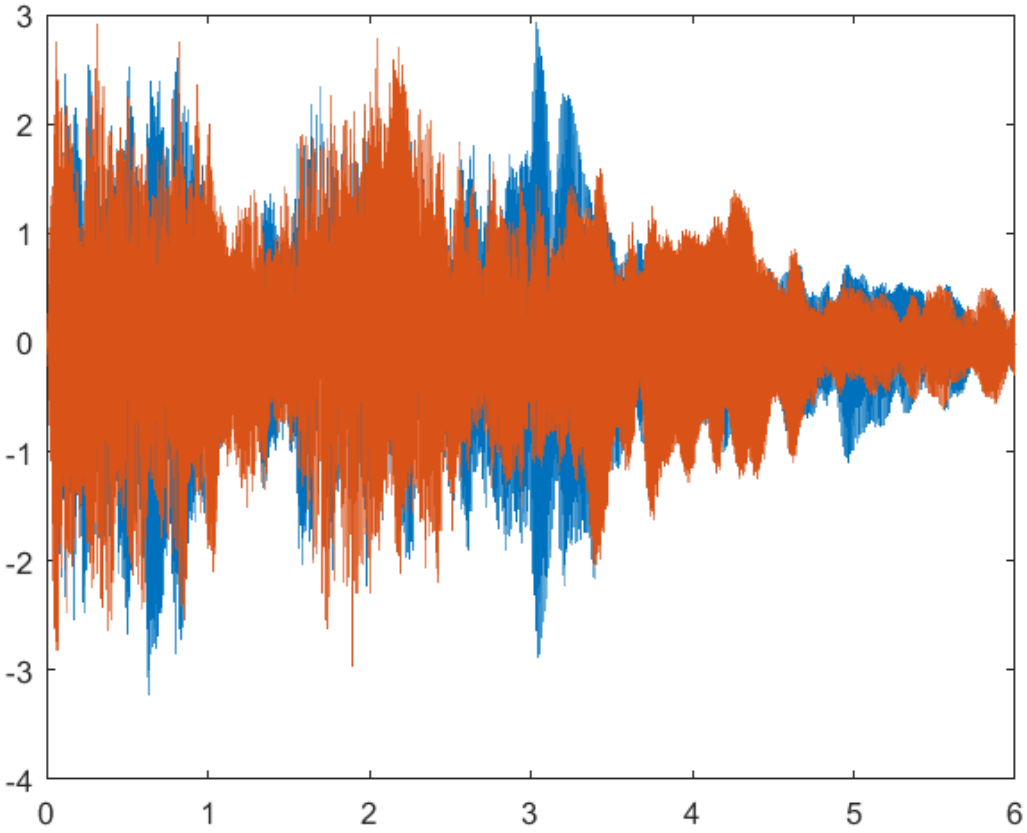
Frequency domain for Original signal



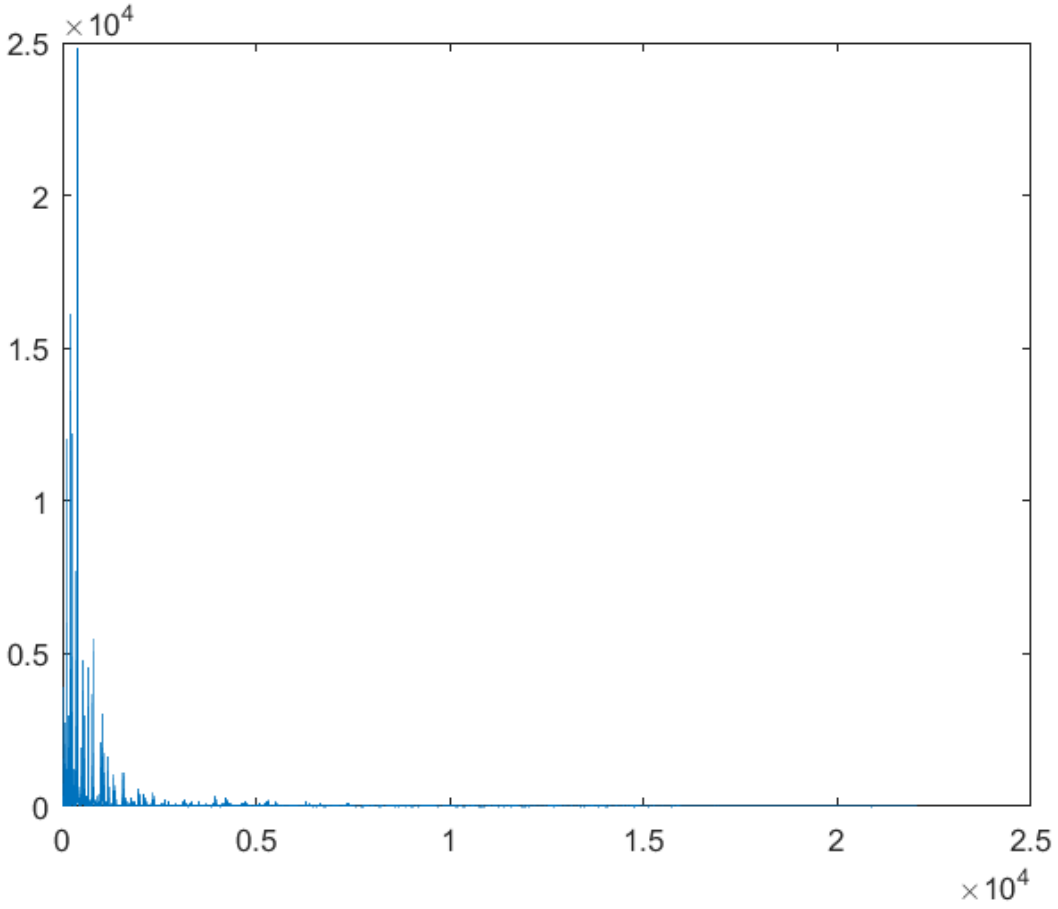
Time domain for Original signal



Time domain for Composite signal



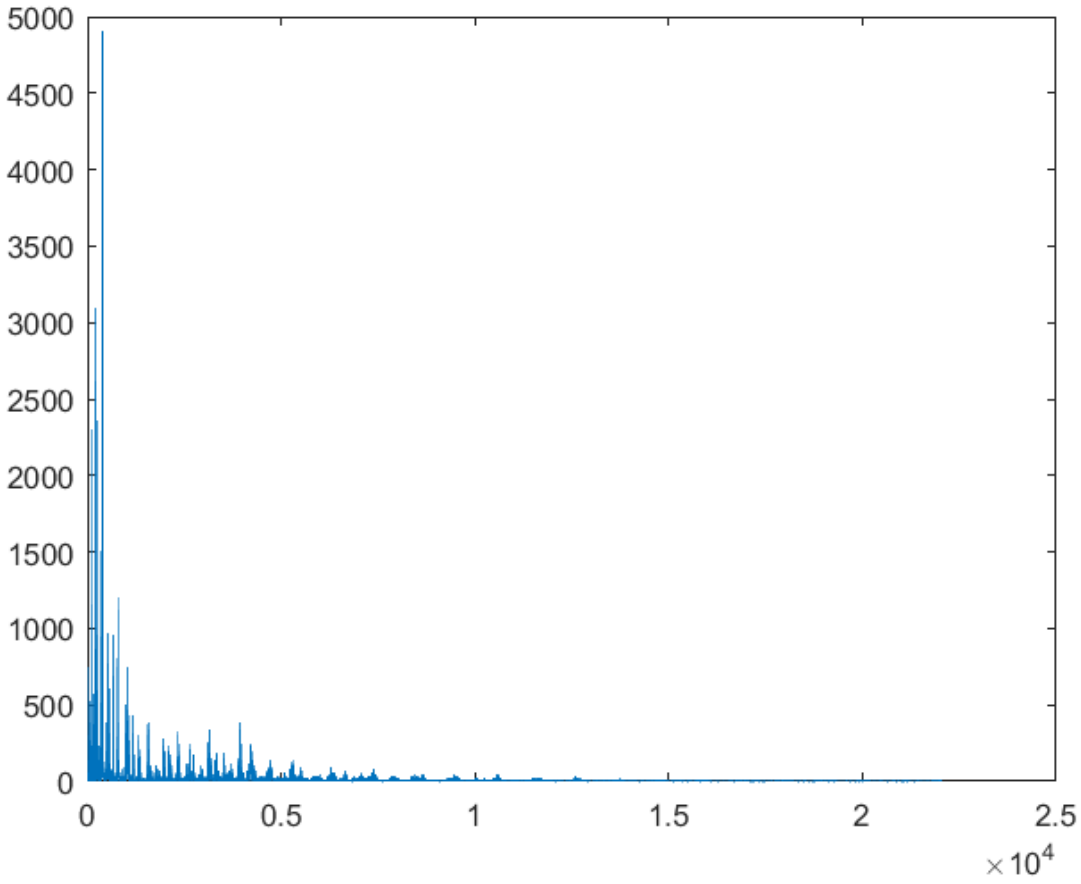
Frequency domain for composite signal



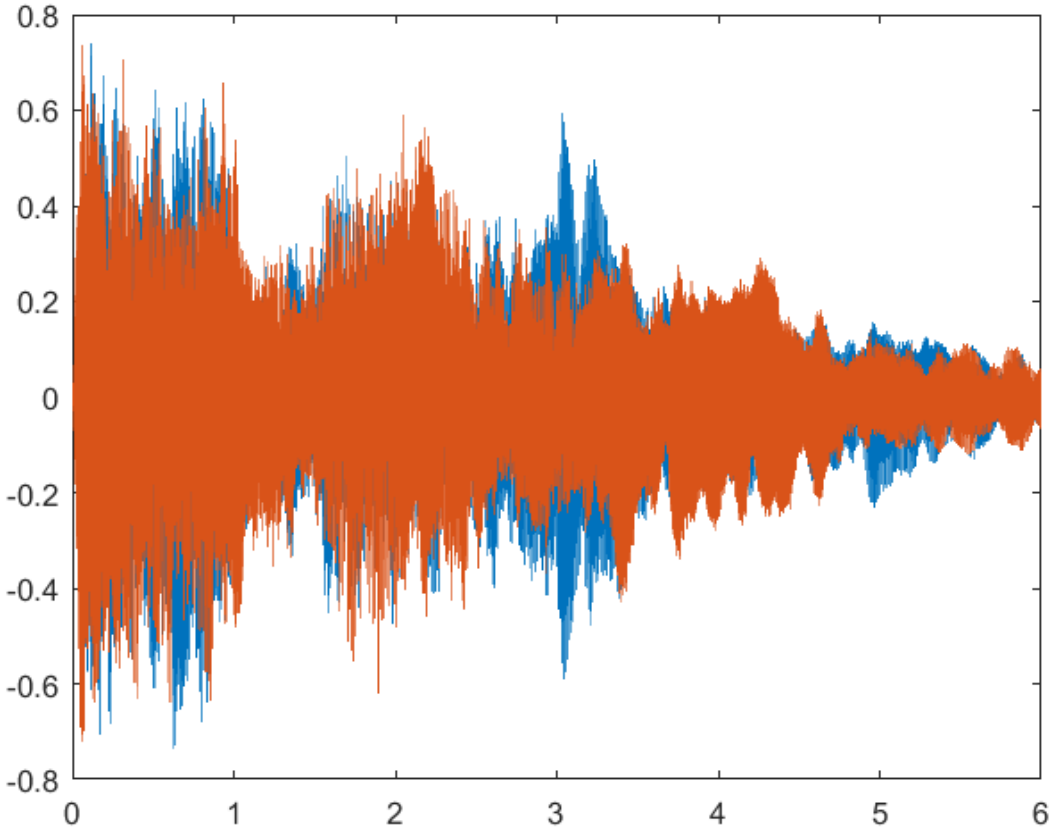
Half rate FIR example

```
main_function('strings4.wav',0.5,'FIR Window',1,1,1,2,2,2,3,3,3);
```

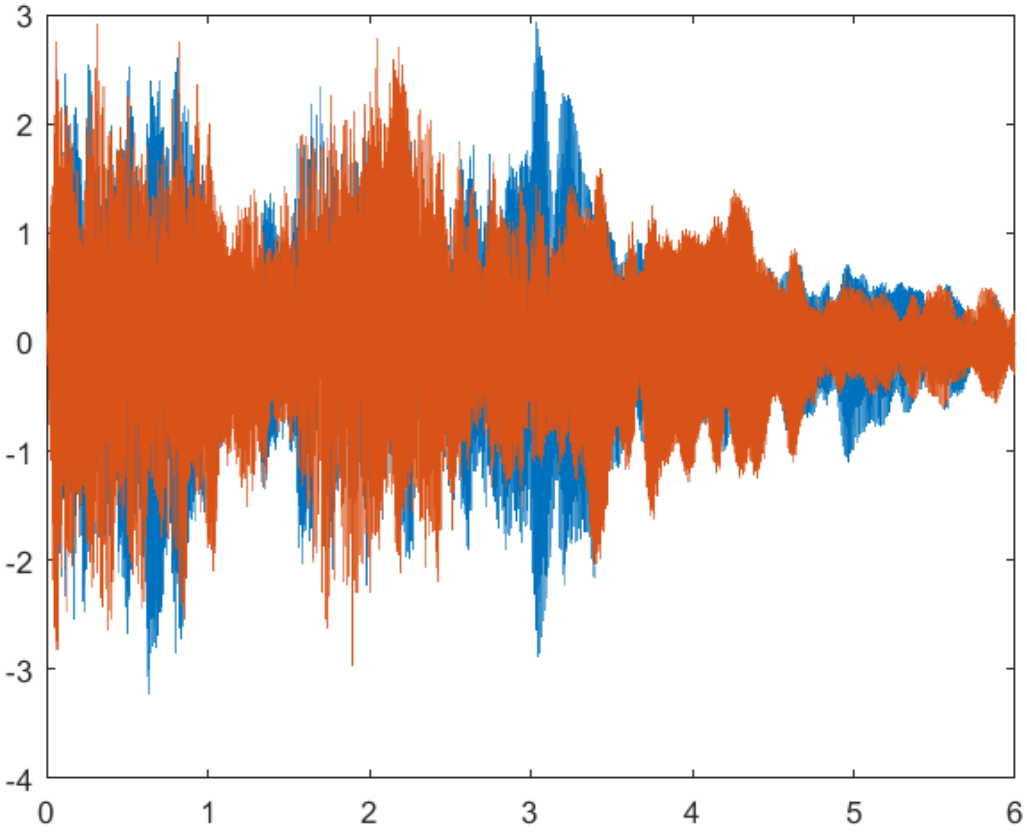
Frequency domain for Original signal



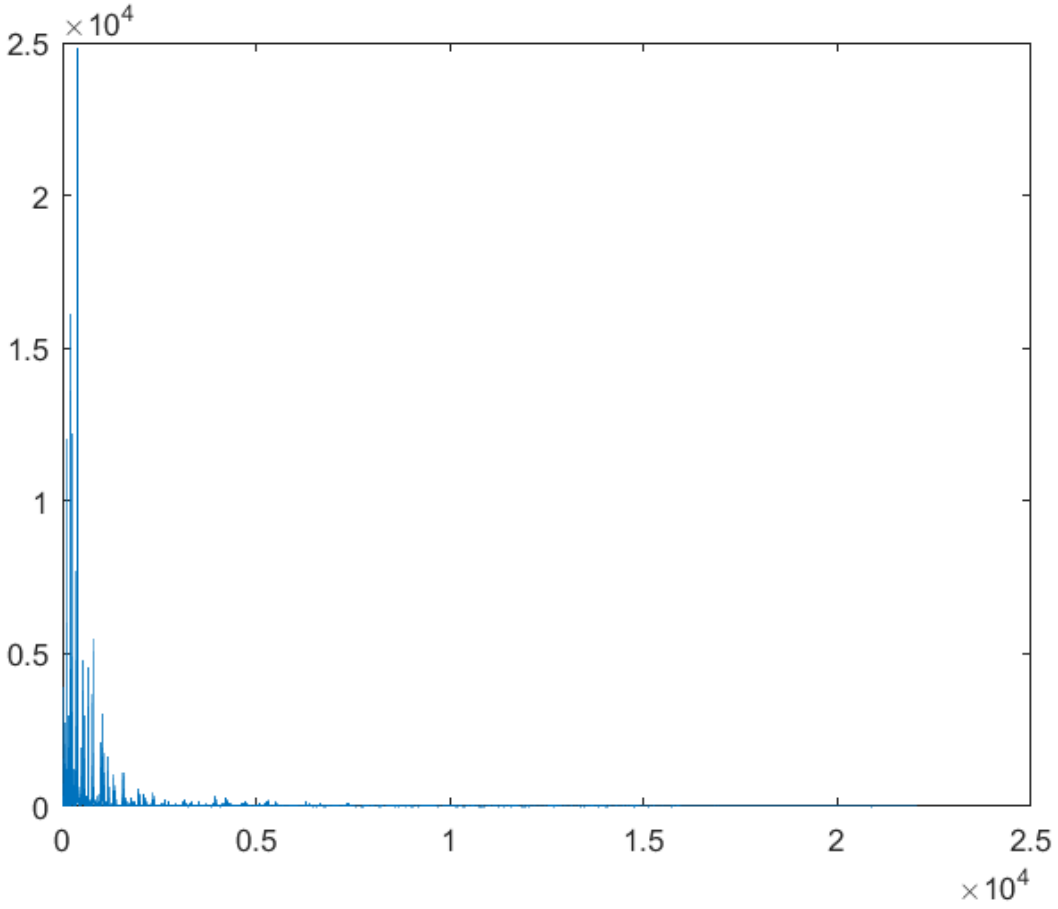
Time domain for Original signal



Time domain for Composite signal



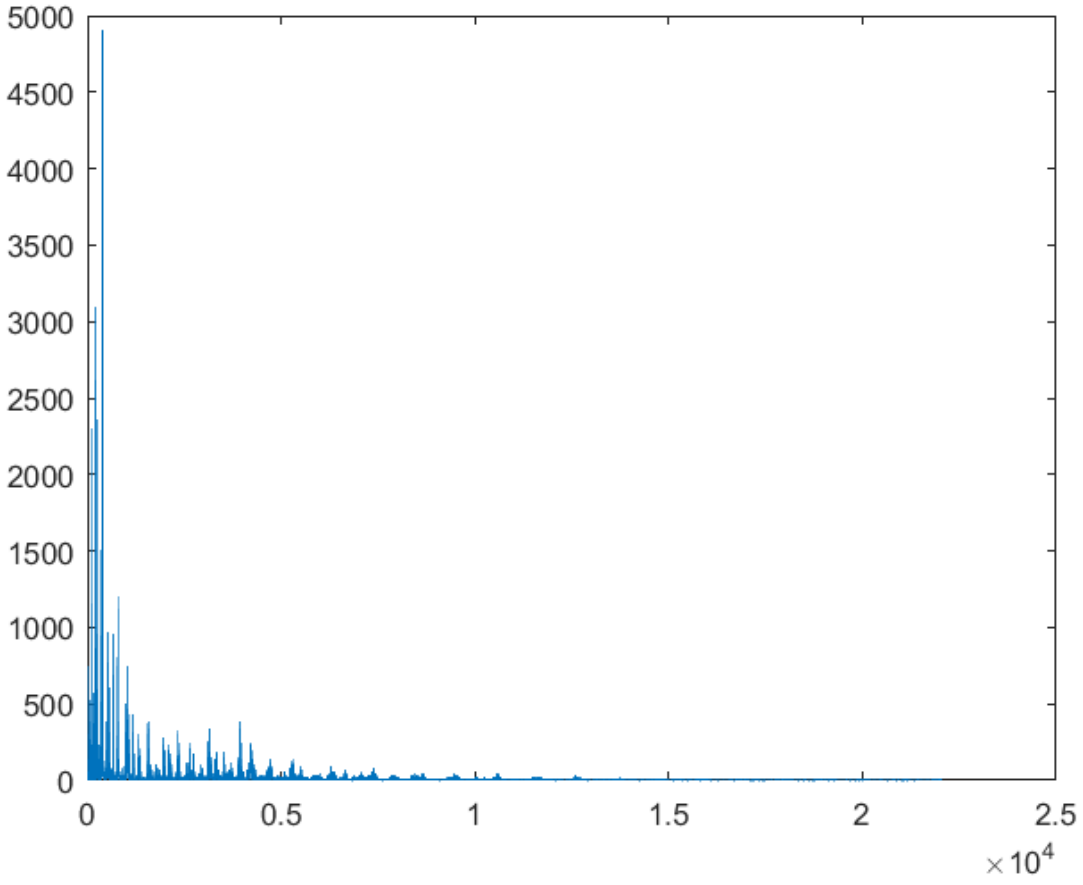
Frequency domain for Composite signal



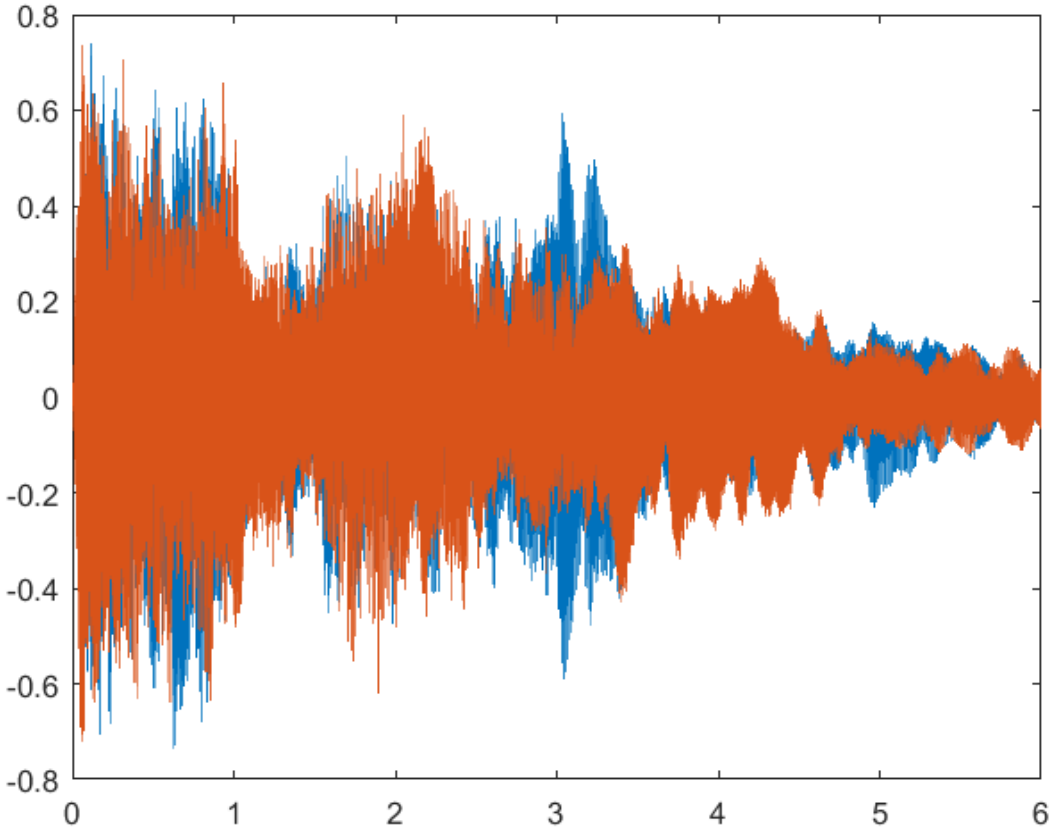
Base Sample Run

```
main_function('strings4.wav',1,'FIR Window',12,0,0,0,0,0,0,0,0,0);
```

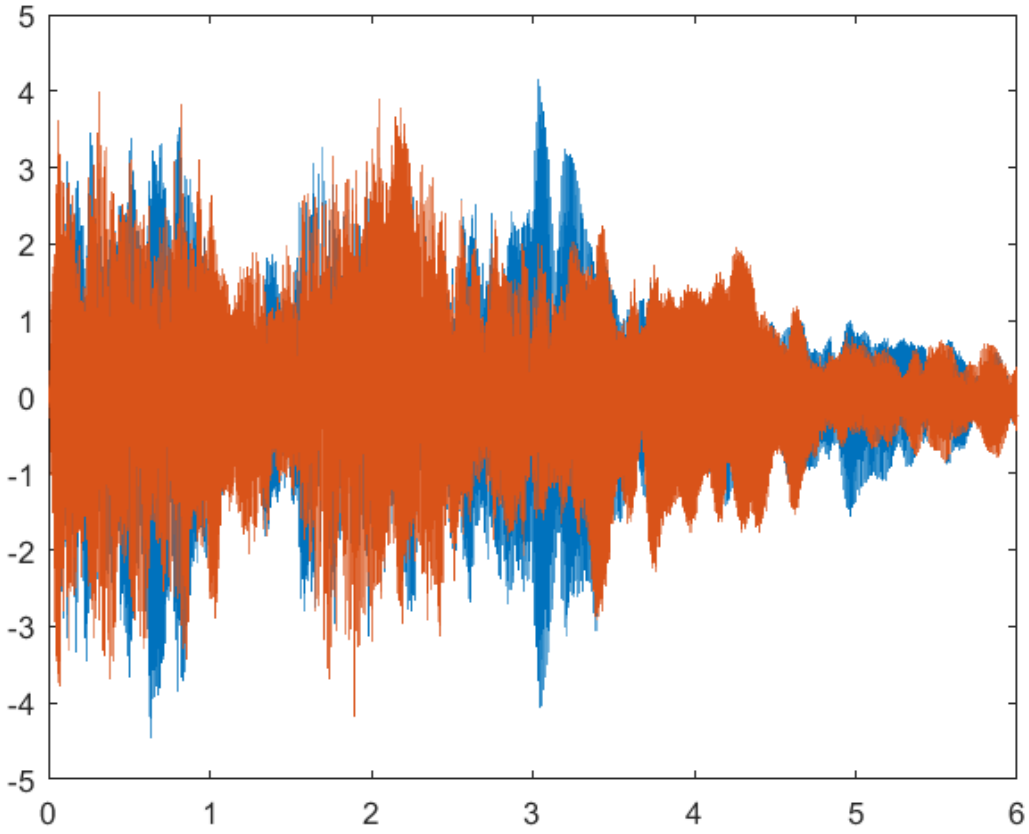
Frequency domain for Original signal



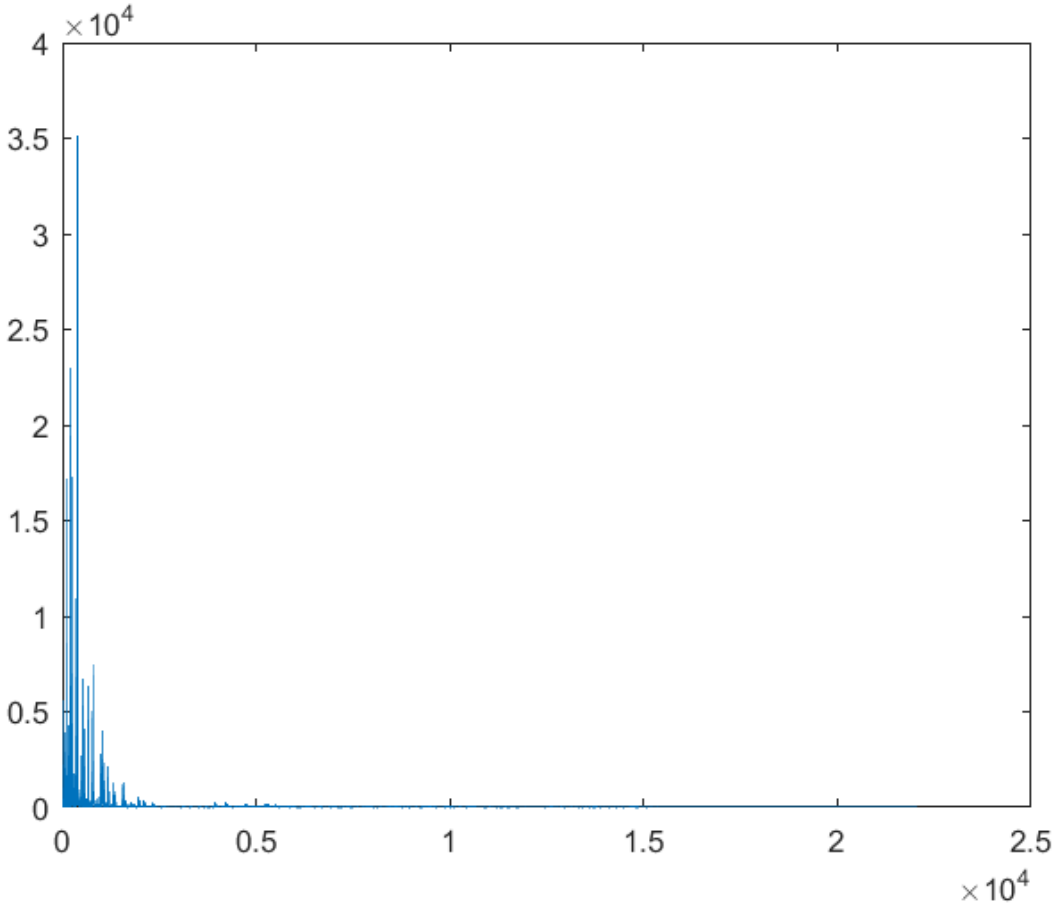
Time domain for Original signal



Time domain for Composite signal

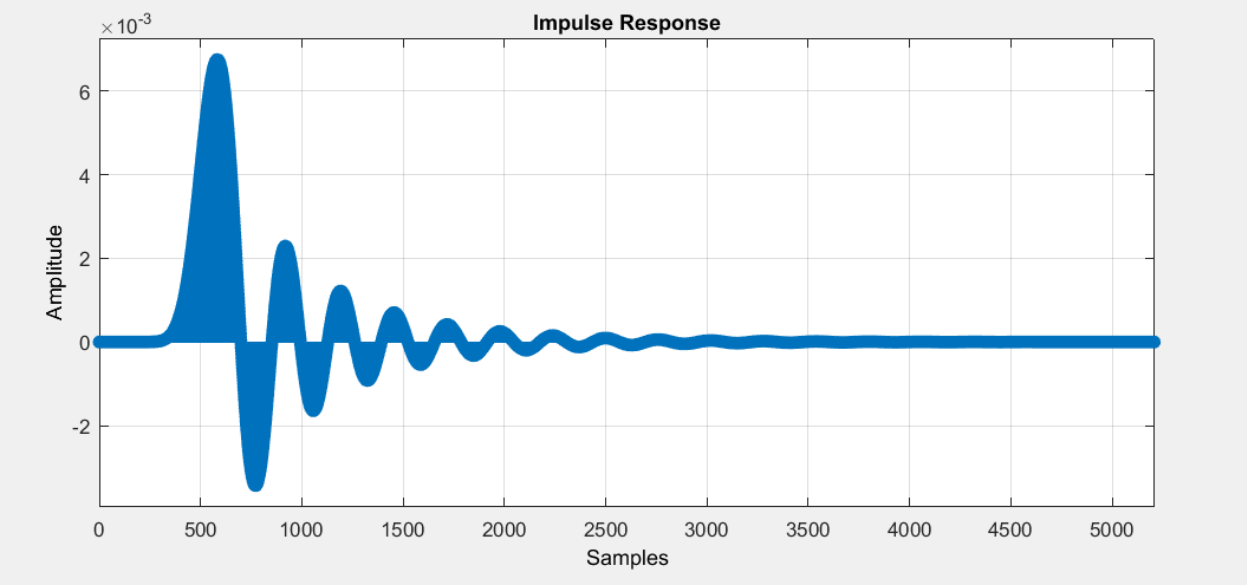


Frequency domain for Composite signal

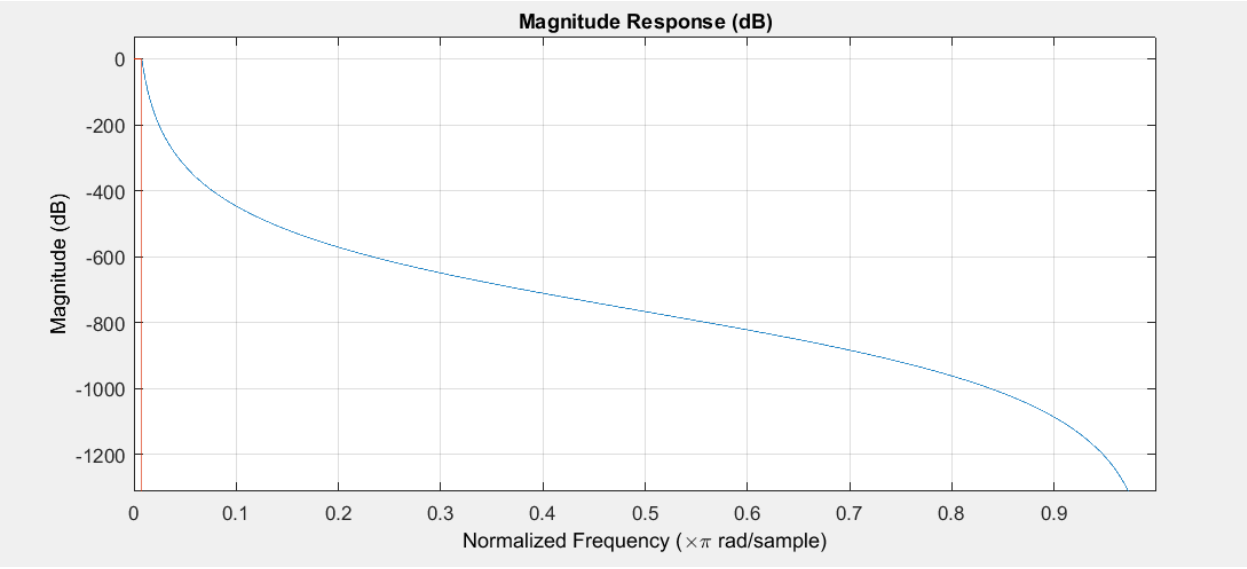


Filter Specification for band 1 IIR

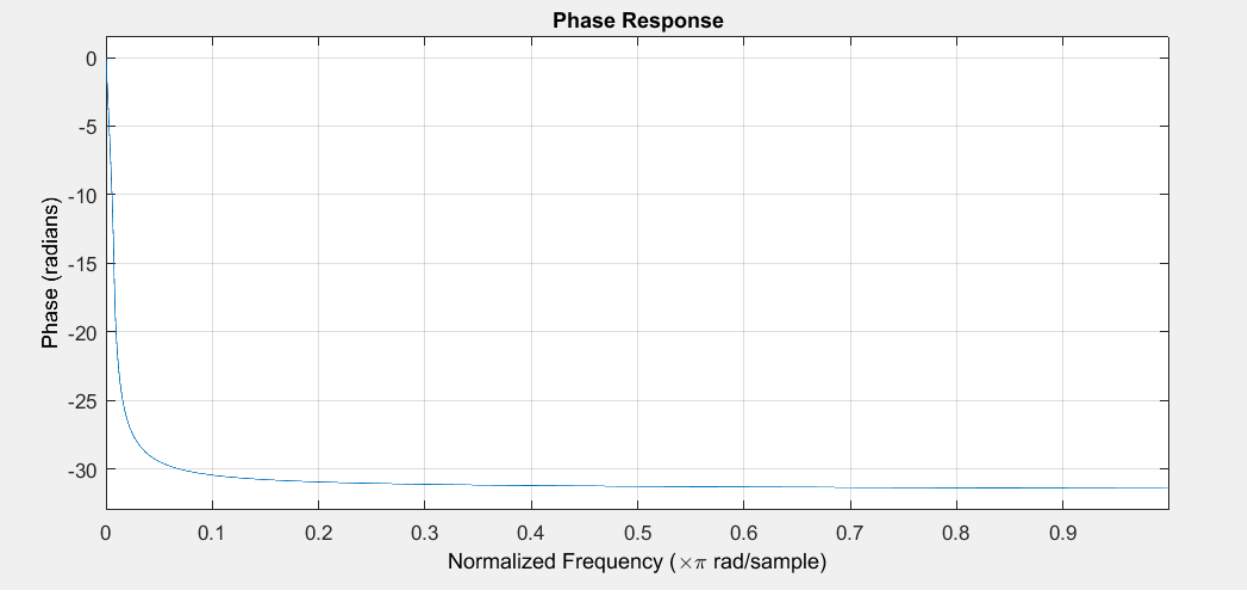
Impulse response



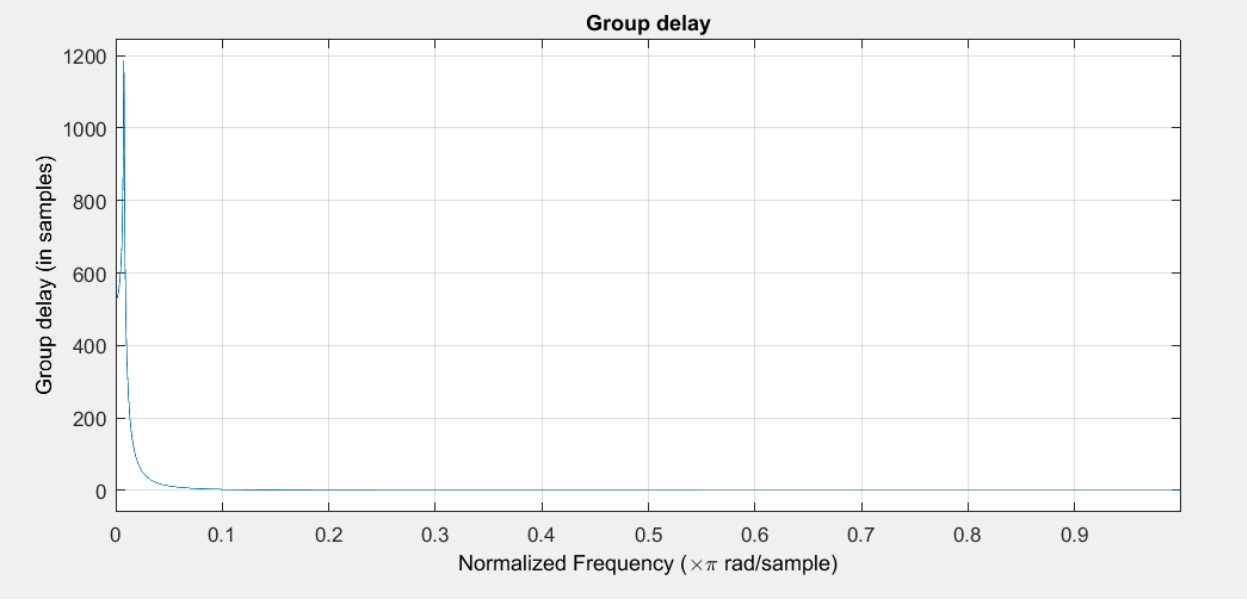
Frequency Response (Gain)



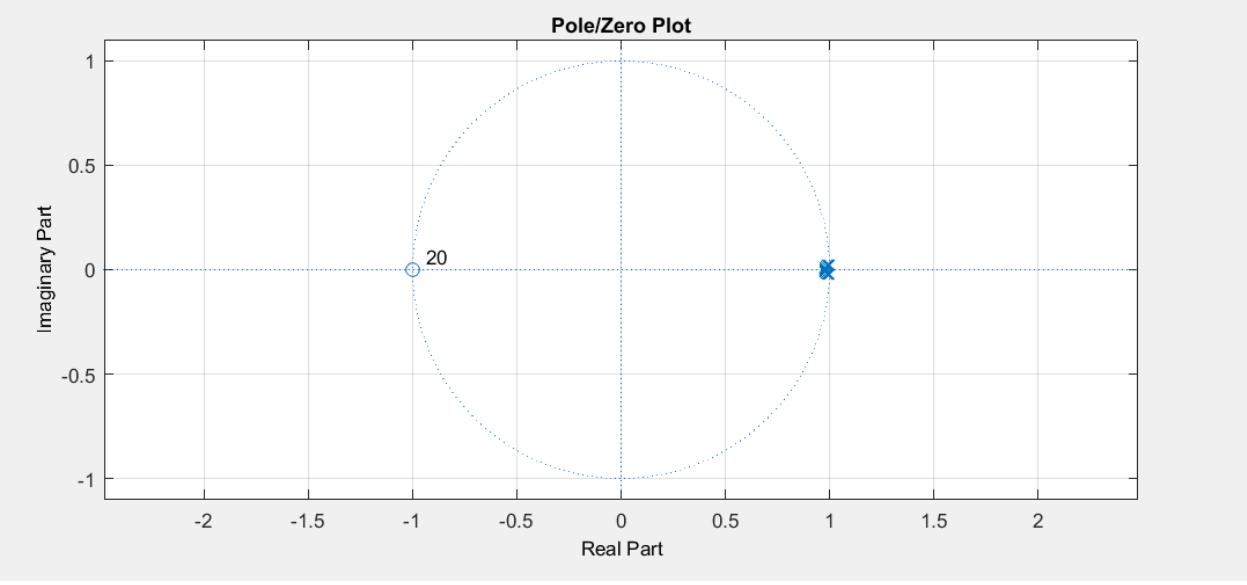
Frequency Response (phase)



Group delay

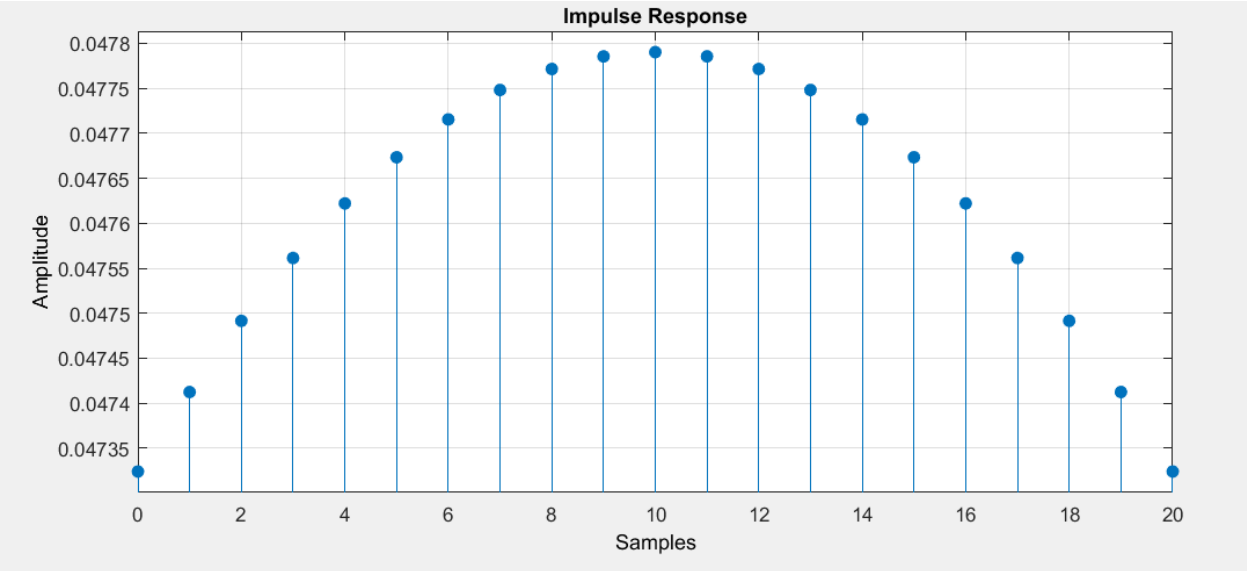


poles and zeros

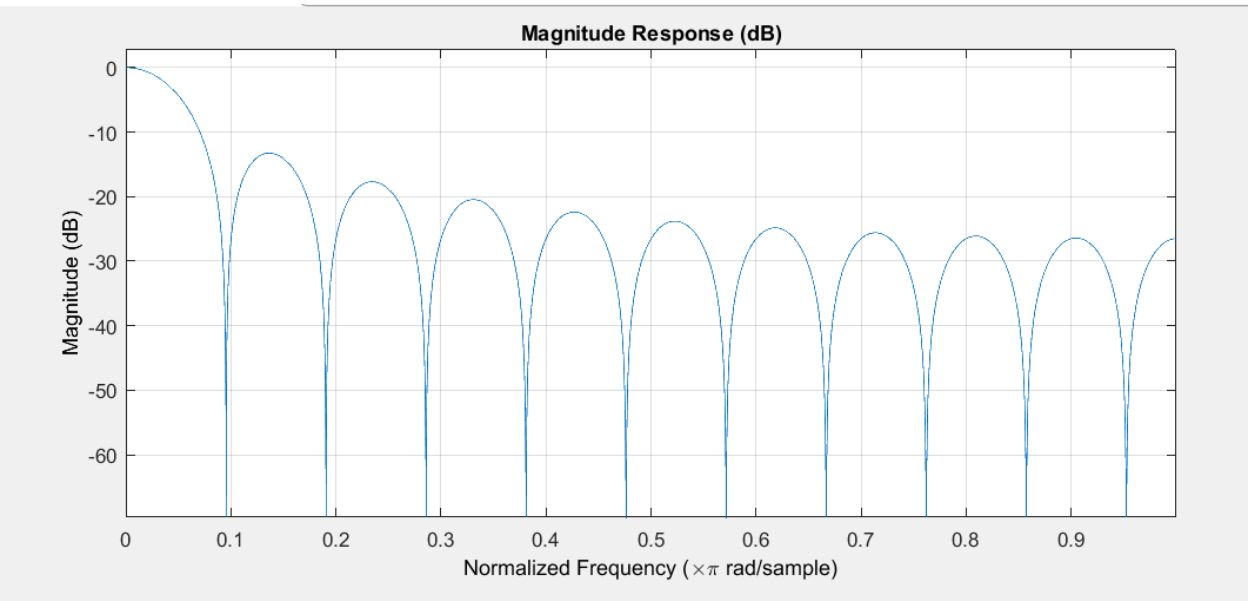


Filter Specification for band 1 FIR

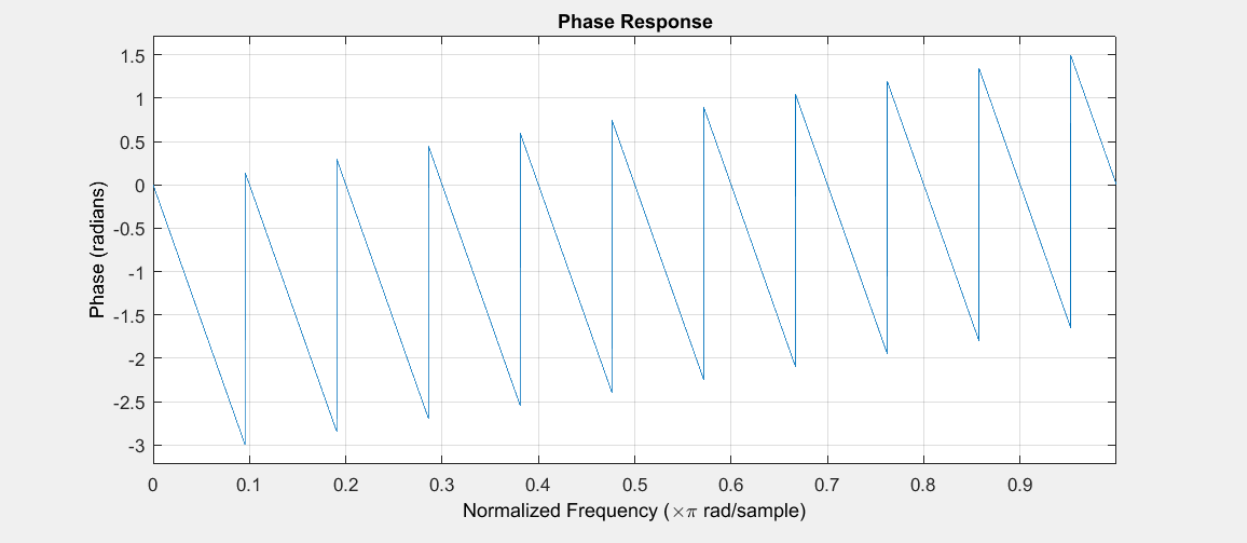
Impulse response



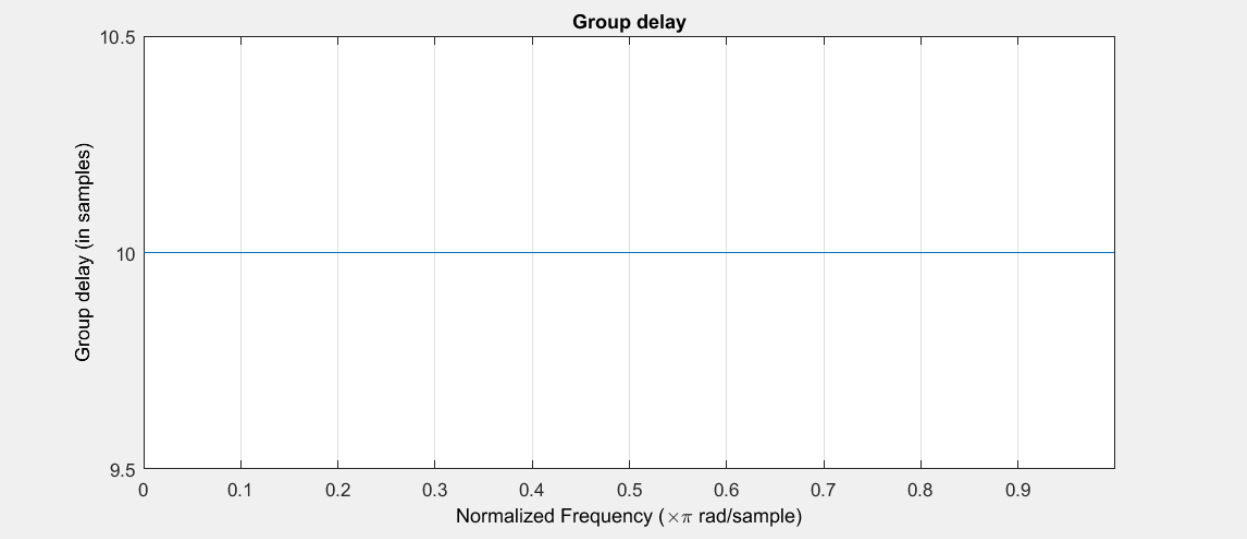
Frequency response (gain)



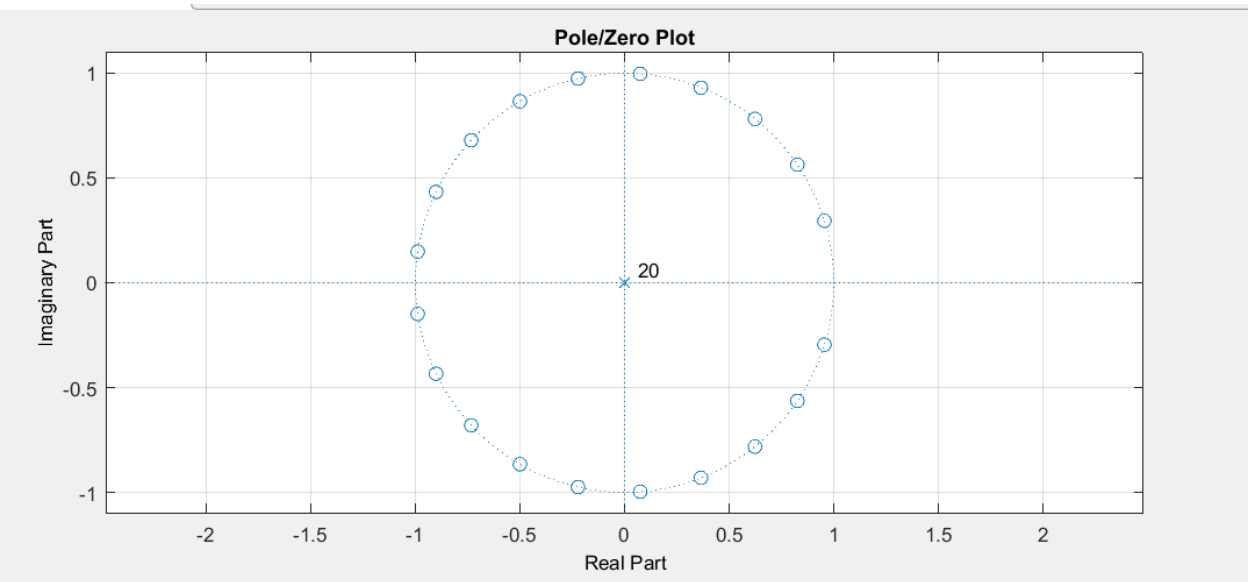
Frequency response (phase)



Group delay



Plots and zeros



Error :-

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} |Hd(\omega) - H(\omega)|\, d\omega, \text{ average absolute error}$$

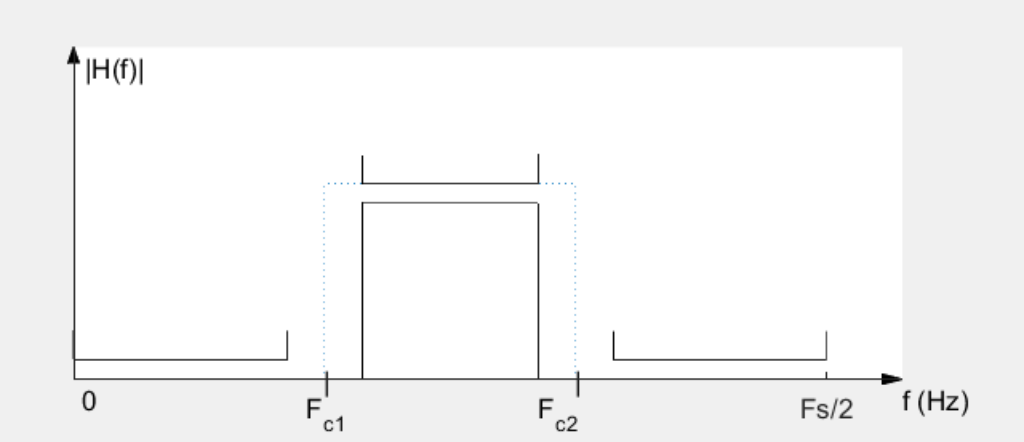
$\max_\omega |Hd(\omega) - H(\omega)|$, maximum error ($p = \infty$)

$\max_\omega |W(Hd(\omega) - H(\omega))|$, maximum weighted error ($p = \infty$)

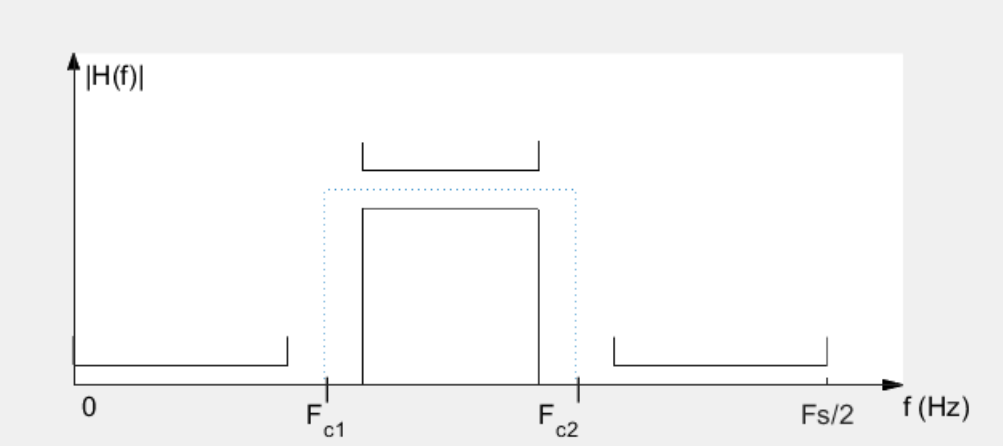
the maximum weighted error between the desired response and the actual frequency

response of an FIR filter. We want to find the FIR filter that minimizes this error.

Filter Specifications for IIR

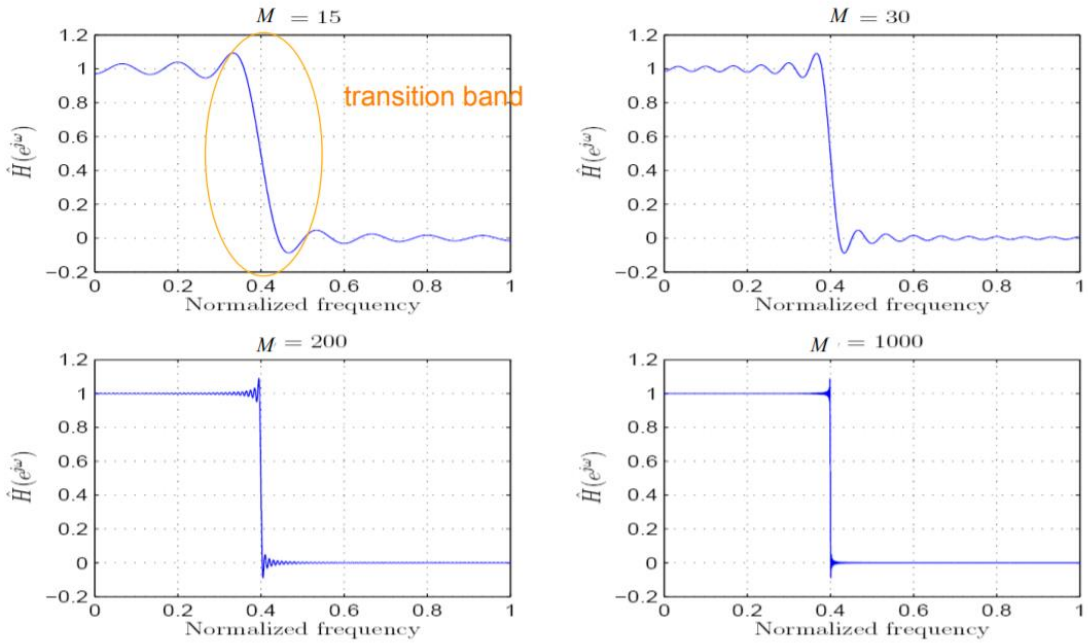


Filter Specifications for FIR

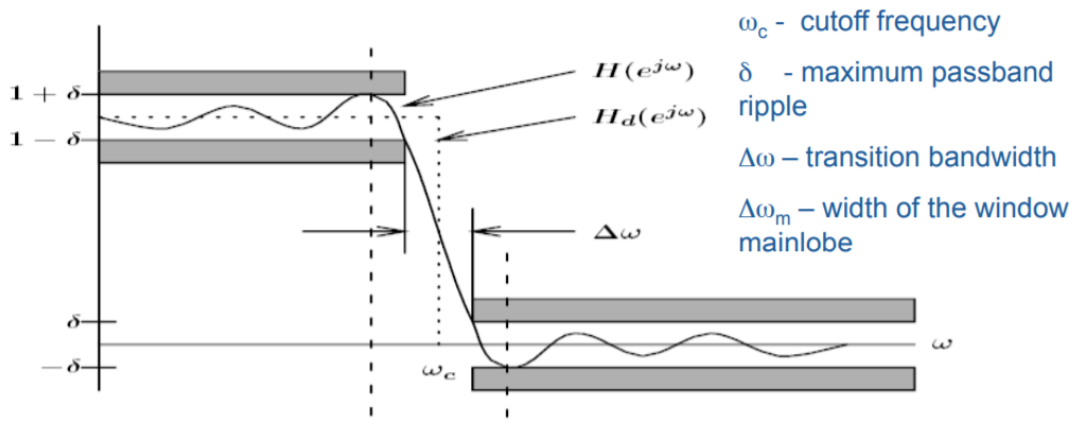


Fc1,Fc2 : range of frequency of band pass

Order : as order increase, performance of filter increase we choose a suitable order of 10



As number of samples increase the transition band gets narrower which is the delay so performance of filter is better as error decrease



Low pass Window filter

equal transition bandwidth on both sides of the ideal cutoff frequency

peak approximation error in the passband is equal to that in the stopband

peak approximation error is determined by the window shape independent of the filter order

References :

https://drive.google.com/file/d/11KhjiktC3-GpGtRSdeGu3jFT8wkFr7BX/view?usp=drivesdk

Published with MATLAB® R2017a