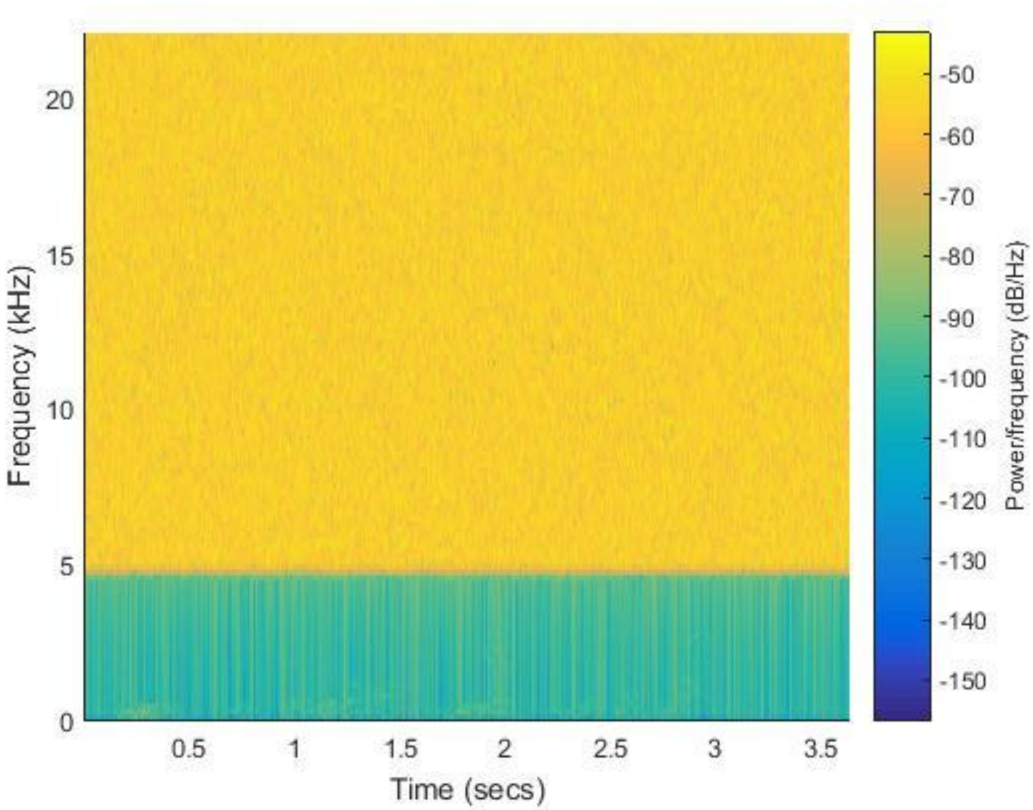
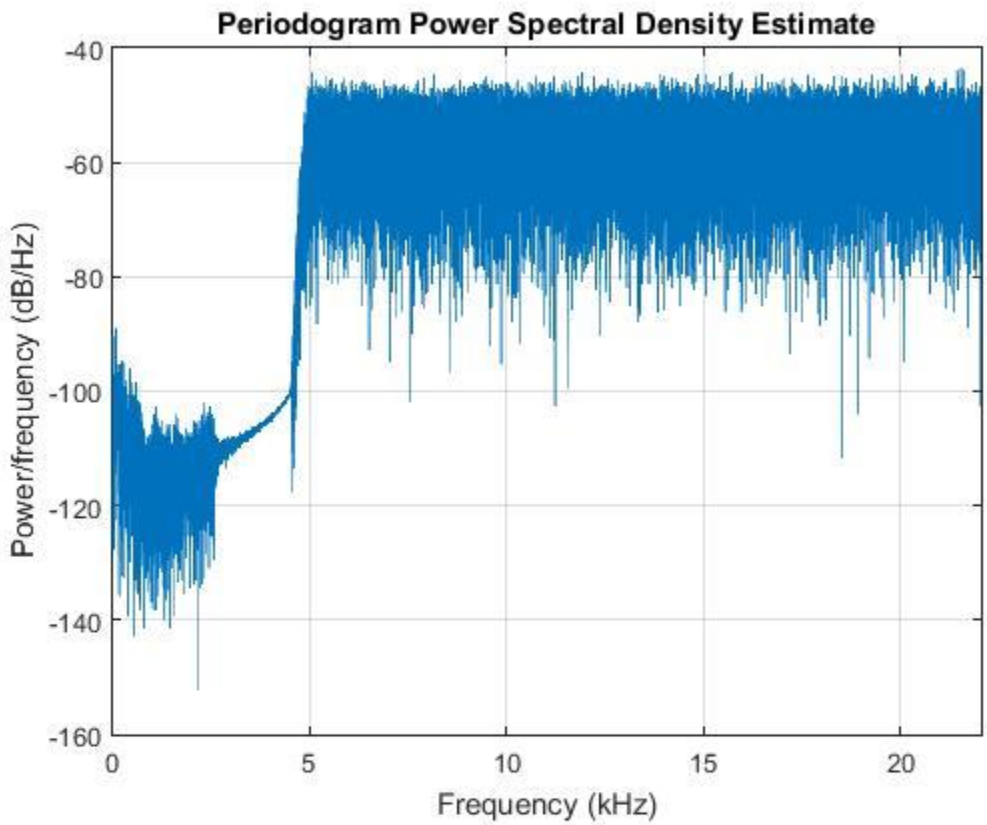
```
%Finite Impulse Response System - Thi T. Le
%This is one of interesting task in DSP that I want to share. The task is
%using Spectrum analysis to define type and properties of your filter.

% Original file noisy2.wav

%Step1: Using spectrogram to mark which part is noise and which part is
%signal

[sound_sample,sampling_rate]=audioread('noisy2.wav');
spectrogram (sound_sample,256,128,256,sampling_rate,'yaxis');
 >>
```

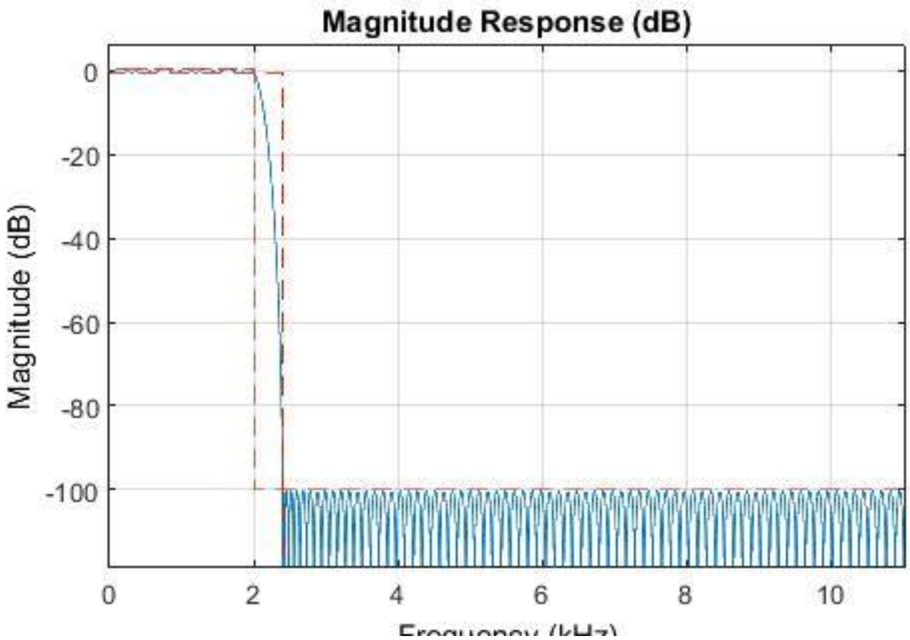



```
% From this we can see the signal in the range of frequency from 0 to 5 kHz
% and level in dB around from -170 to -130 dB. The rest is Noise
```

```
[signal,fs] = audioread('noisy2.wav');
plot(psd(spectrum.periodogram,signal,'Fs',fs,'NFFT',length(signal)));
 >>
```




```
%With the power Spectral Density we can see it more clearly are and decide
%what type of filter and coefficient of it

%It clearly that Lowpass is the optional, cutoff frequency is 2kHz
%Stopband is 100dB (80dB still fine but I decide 100dB for sure)
%Transition bandwidth is 400Hz


Fs = 22050;
 d= designfilt('lowpassfir', ...
  'PassbandFrequency',2000,'StopbandFrequency',2400, ...
  'PassbandRipple',1,'StopbandAttenuation',100, ...
  'DesignMethod','equiripple','SampleRate',Fs);
fvtool(d);
 >>

%This is the amplitude response we have from our filter.
```




```
%We filter the signal and try to hear what we have. At first it's nothing
%but if you amplify the signal and also the rate sample you will able hear
%something from it.

df= filter(d,signal);
audiowrite('afterwork.wav',df*50,Fs*2.5)
[y,Fs]=audioread('afterwork.wav');
sound(y,Fs);


%File name is afterwork.wav
```

```
%Optional, you can check the power spectral density again to see what you
%have after filter the noise. I used this step to choice the coefficient of
%my lowpass filter

plot(psd(spectrum.periodogram,df,'Fs',Fs,'NFFT',length(df)));
>>
```