```
%Authors: Wendy Mendoza & Shahrear Khan Faisal
%Due Date: October 18, 2021
%Mid-term Project PHYS 5387
addpath 'https://drive.matlab.com/files/'
%(i):Loading Data for LIGO S5 strain
load S5_878486500_878486600_DARM.mat
% data1 = load('dat1.mat');
data1 = dat1;
fs = 16384; %Sampling frequency

t = length(dat1)/fs; %Duration in seconds
len = t*fs; %total length of the data
timeVec = 0:1/fs:(len-1)/fs; %time vector


%(ii):Estimation of the power spectrum of dat1 using pwelch
methods.
%Power spectrum density measure the signal power versus
frequency(width).
%PDS shows the strong and weak frequencies
variations(energy).
%They are known to become auto PSD Pxx(f)=Pxx(exp[ij2pif])
and cross PSD Pxy(f)=Pxy(exp[j2pif])
%Auto-PSD describes the frequency of the power of x[n] and
real and
%nonnegative. Cross-PSD describes the frequency component
in x[n] are
%associated with large or small amplitude same frequency
y[n].
%For the PSD we use the terms of Welch method to approach a
spectral density estimation.
% Using PWELCH methods to calculate the spectral density.
[pxx, f] = pwelch(data1,[],[],[],fs);

% (iii):Plotting the time series and the power of spectrum.
figure(1)
subplot(2,1,1);
plot(timeVec,dat1);
xlabel('Time(sec)');
ylabel('Amplitude');
title('Time Series');
subplot(2,1,2);
plot(f, 10*log10(pxx));
xlabel('Frequency(Hz)');
```

```
ylabel('PSD (dB/Hz)');
title('Power Spectral Density(PWELCH)')
```

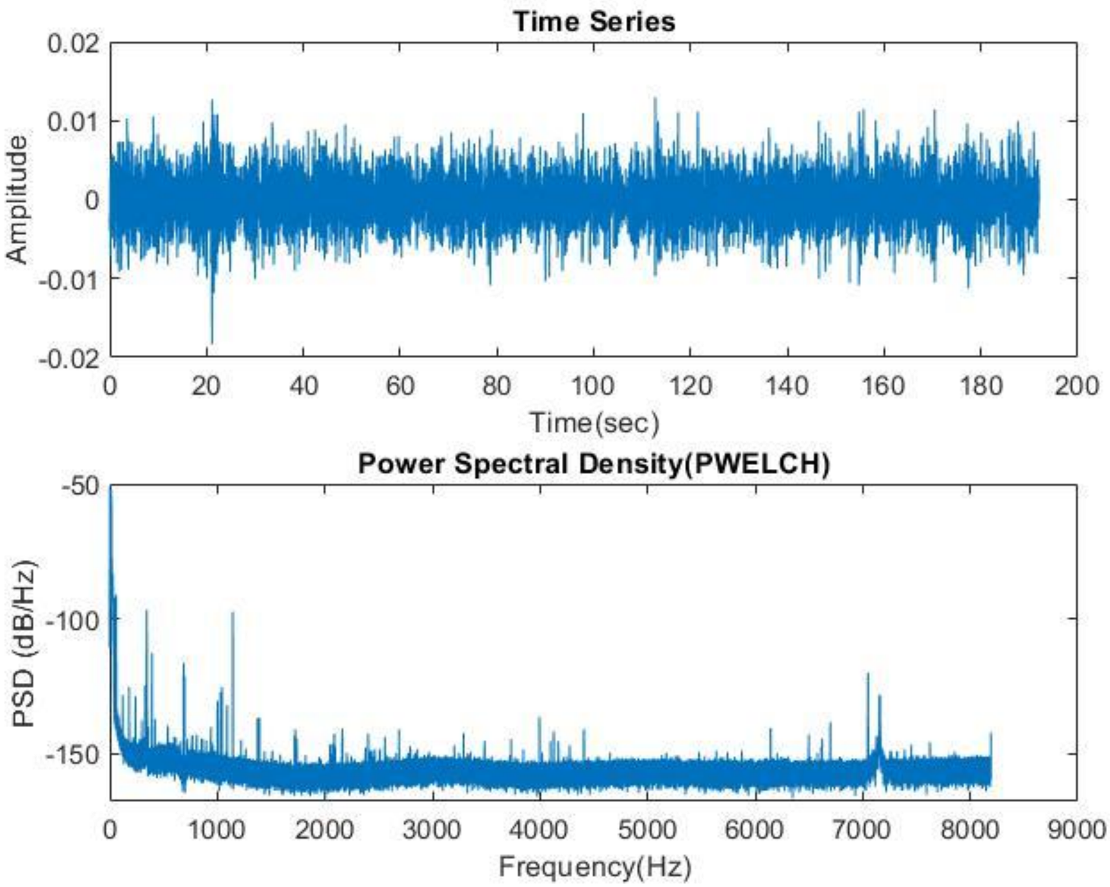

Figure 1: Time Series and Power Spectral Density of the Original Data

```
%(iv):Applying the lowpass fiter to data1
Wn = 2048/fs/2;  %Cutoff frequency of 0.25
[b,a] = butter(6, Wn, 'low'); %Bandpass of 6th order
Butterworth filter
% The transfer function
% H(z)=B(z)/A(z)=b0+b1z^-1_b2z^-2+..+bnz^-
N/B(z)/A(z)=a0+a1z^-1_a2z^-2+..+bMz^-M

%Filtering the data
dat_low_pass = filter(b,a,data1);
[pxx_2, f2] = pwelch(dat_low_pass,[],[],[],fs);
figure(2)
```

```
subplot(2,1,1);
plot(timeVec,dat_low_pass);
xlabel('Time(seconds)');
ylabel('Amplitude');
title('Filtered Time Series');
subplot(2,1,2);
plot(f2, 10*log10(pxx_2))
xlabel('Frequency (Hz)');
ylabel('PSD (dB/Hz)');
title('Power Spectral Density(PWELCH) low pass filter');
%The low pass filter retains frequencies below a given cut-
off which means
%it eliminate the higher frequencies to allow the lower
frequencies to pass
%through.
```

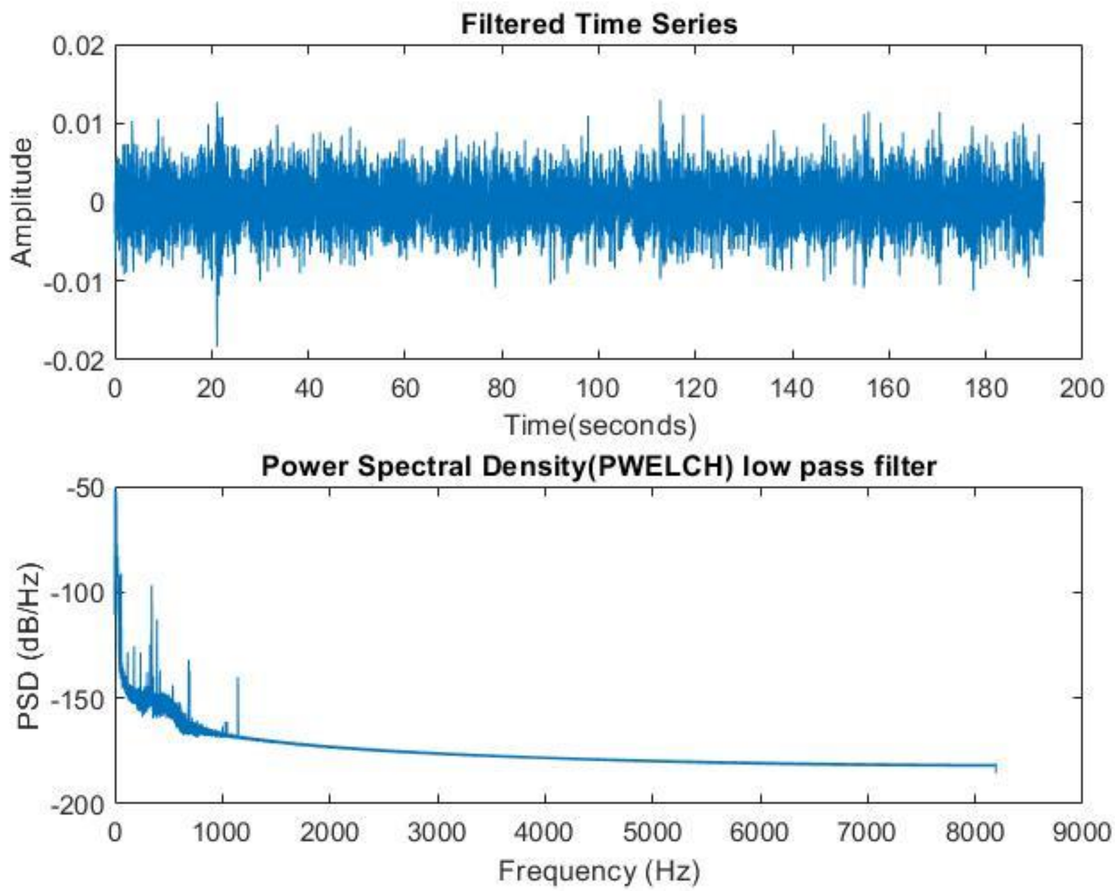

Figure 2: Time Series and Power Spectral Density of the Original Data

```
%(v):Resample the data to a lower sampling frequency
p = 1; %Resampling factors
q = 4; %Resampling factors
dat_low_pass2=resample(dat_low_pass,1,4); %change sampling
rate
%New sample frequency will be 4096Hz
fs_new = fs * p/q;
t2 = length(dat_low_pass2)/fs_new;  %duration in seconds
len2 = t2*fs_new; %lenght of the data
timeVec2 =  0:1/fs_new:(len2-1)/fs_new; %time vector


%(vi)pwelch the new sampling rate
[pxx_3, f3] = pwelch(dat_low_pass2,[],[],[],fs_new);
figure(3);
plot(f3, 10*log10(pxx_3));
xlabel('Frequency(Hz)');
ylabel('PSD (dB/Hz)');
title('Power Spectral Density with Resample Data');
% Comparing the power spectrum of the data before and after
low pass
% filering, we can see that before the data had frequency
components above
% 8 kHz, but after applying the filter the frequency
components above 2048
% Hz have been discarded, which is expected, as out cut of
frequency of the
% butterworth filter was 2048 Hz.
```

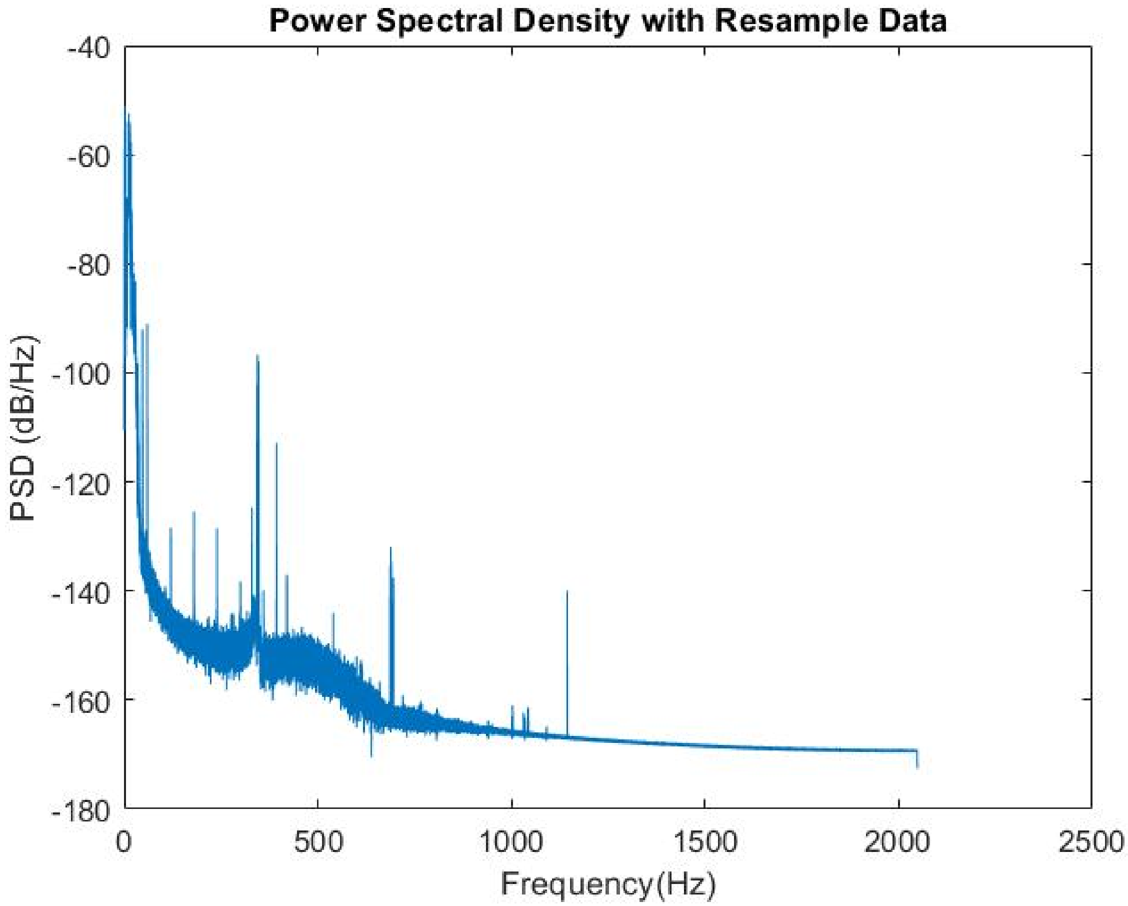

Figure 3: Power Spectral Density of the resampled data

```
%(vii):Whiten the data
pxx_median_est = rngmed2(pxx,256);
%Median is the middle number of a Gaussian distribution in
an order set of data values
%symmetrically around mean=median=mode.

%Inverse the median
freq=0:0.125/(fs*4):1;
%This function design filters with frequency inverse and
the magnitude characterisitics of
%the white noise which is the containing vectors desires in
the frequency.
N = 500;
bfilt=fir2(500,freq',1./sqrt(pxx_median_est)); %N is number
of order, frequency, & magnitude
```

```
%fir2 is a digital filter order specified as an integer
scalar.
%This return the 500 orders FIR filter with frequency
magnitude charateristics.
%The filter coefficients are obtained by linearly
interpolating the required
%frequency response onto a dense grid and then using the
inverse Fourier transform and a Hamming window.
Dat_whitened=fftfilt(bfilt,dat_low_pass2);
[pxx_4, f4] = pwelch(dat_whitened,[],[],[],fs_new);

figure(4);
subplot(2,1,1)
plot(timeVec2,dat_whitened);
xlabel('Time(sec)');
ylabel('Amplitude');
title('Time Series of Whitened Data');
subplot(2,1,2);
plot(f4, 10*log10(pxx_4));
xlabel('Frequency(Hz)');
ylabel('PSD (dB/Hz)');
title('Power of Spectral Density of Whiten Data')
```

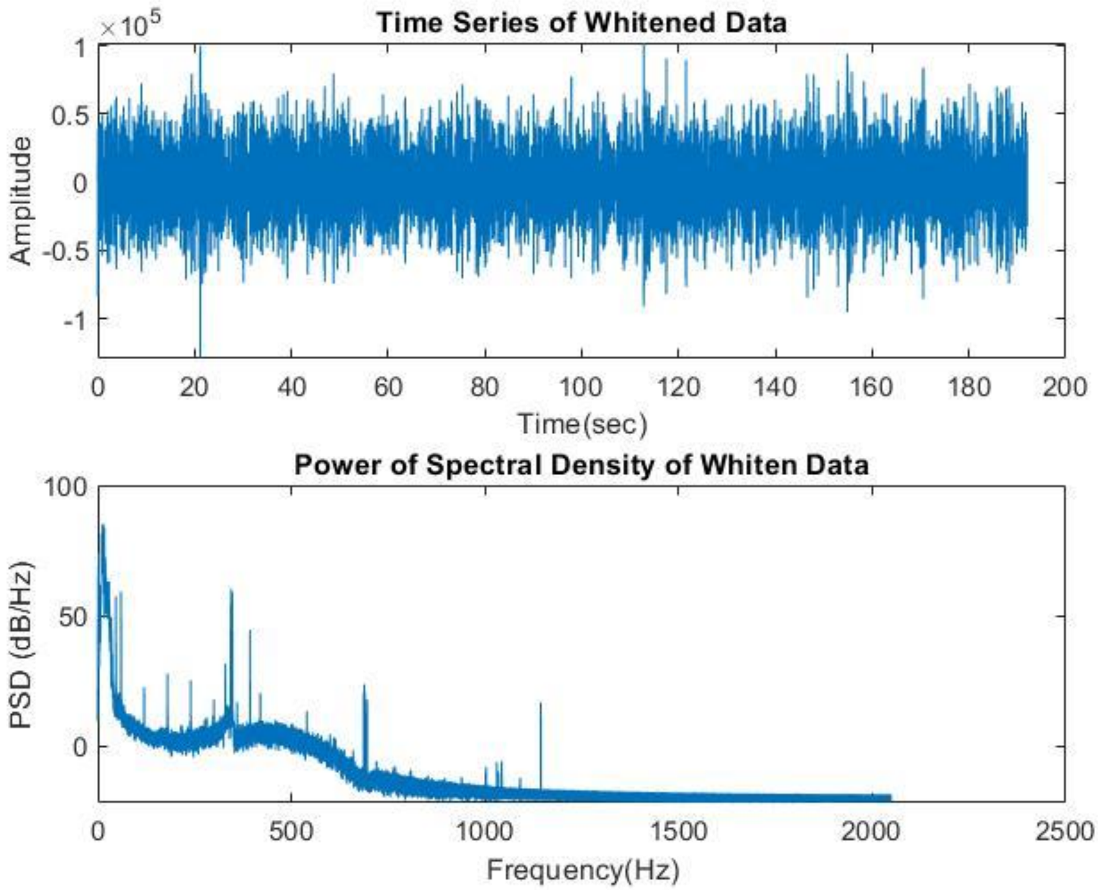

Figure 4: Time Series and Power Spectral Density of whitened data

```
%(viii) Applying the Bandpass filter to the whitened data
[b,a]=butter(6,[100 1024]/2048);
dat_whitened2=filtfilt(b,a,dat_whitened);
[pxx_5, f5] = pwelch(dat_whitened2,[],[],[],fs_new);

figure(5);
subplot(2,1,1)
plot(timeVec2,dat_whitened2);
xlabel('Time(sec)');
ylabel('Amplitude');
title('Time Series of Bandpass Whiten Data');

subplot(2,1,2);
plot(f5, 10*log10(pxx_5));
xlabel('Frequency(Hz)');
```

```
ylabel('PSD (dB/Hz)');
title('Power of Spectral Density Bandpass Whiten Data');
%The bandpassfilter retains frequencies between a given
lower cut-off and a
%higher cut-off. % Why we bandpassed the whitened data -
Because LIGO is most sensitive
%between 100 Hz and 1000 Hz. Outside this frequency range
signals other
%than our desired signals (noise) prevail, which makes it
nearly
%impossible to detect gravitational wave in those frequency
ranges.
```

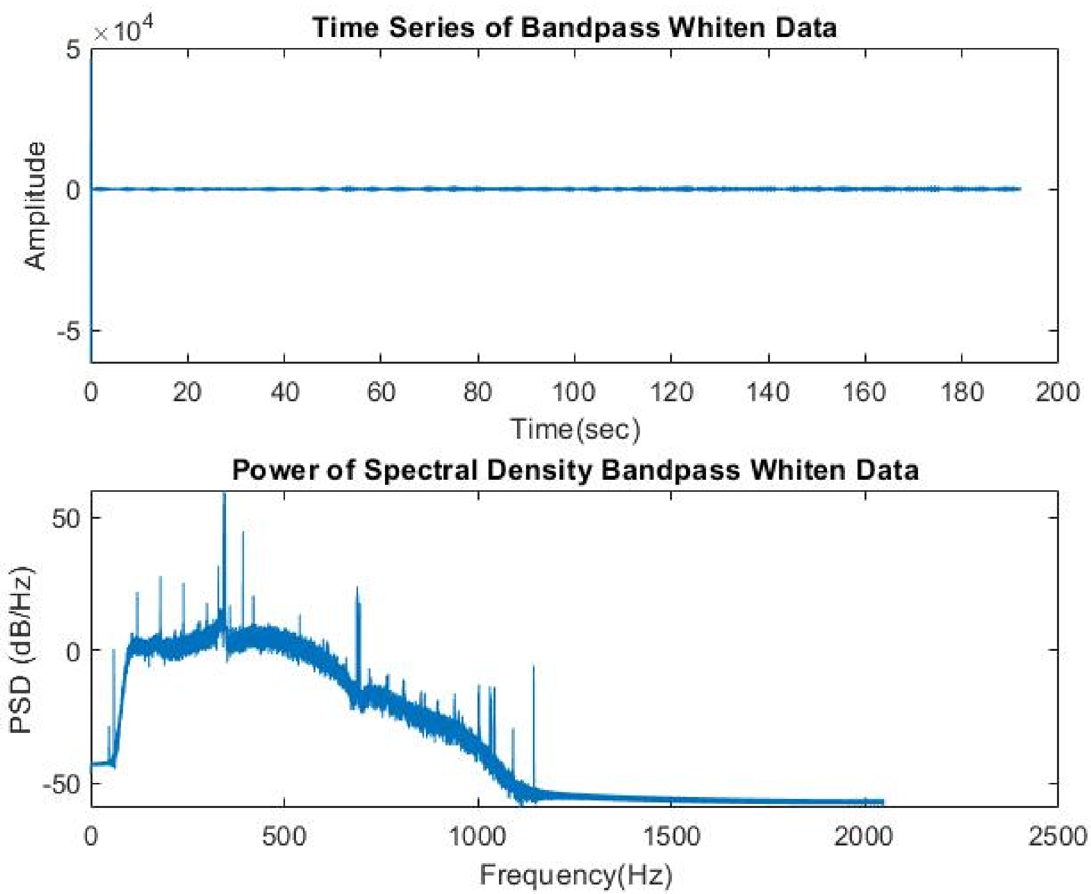

Figure 5: Time Series and Power Spectral Density of bandpassed and whitened data

```
%(ix):Resample the whitened data to 2018 HZ
dat_low_pass2=resample(dat_low_pass,1,4);
nfft = 2018 * 4;
[pxx_6, f6] = pwelch(dat_low_pass2,[],[],nfft,fs_new);
figure(6);
plot(f6, 10*log10(pxx_6));
xlabel('Frequency(Hz)');
ylabel('PSD (dB/Hz)');
title('Power of Spectral Density Resample Data to 2018
Hz');
```

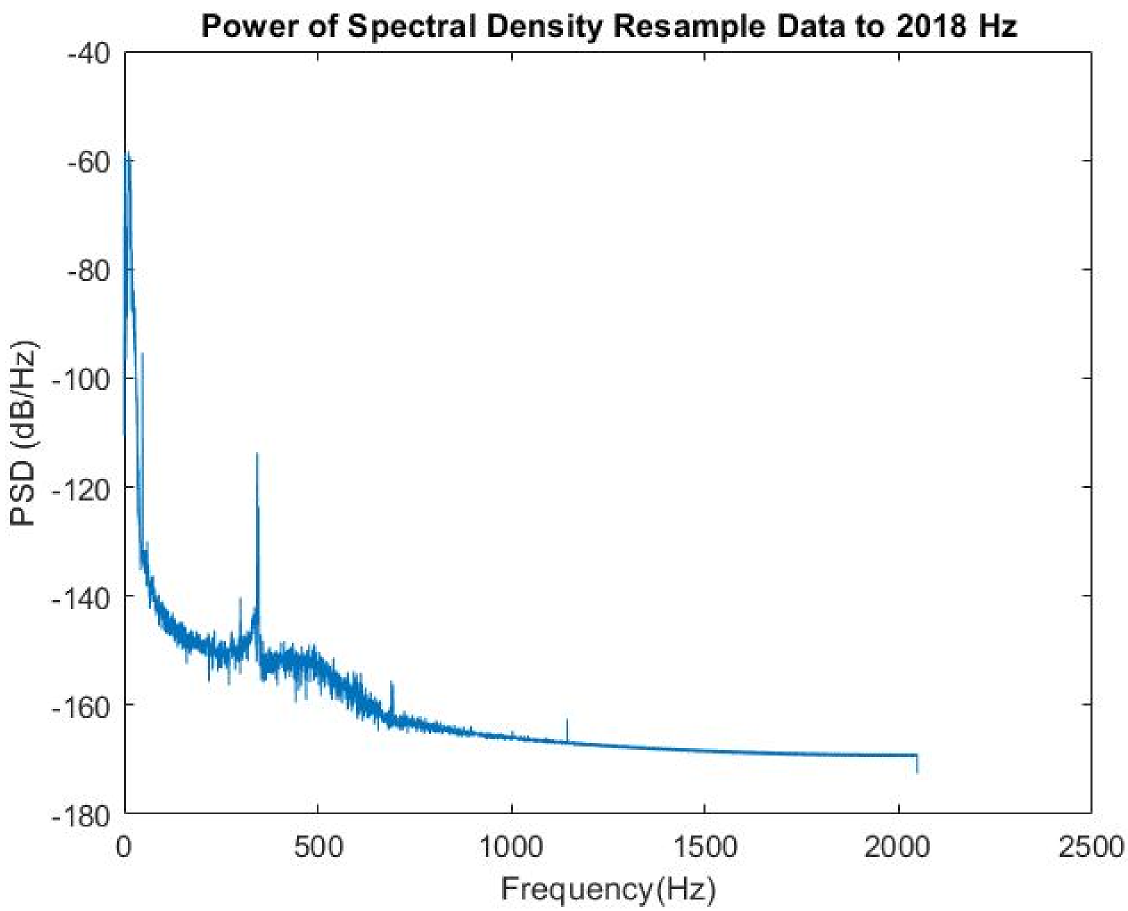

Figure 6: Power Spectral Density of resampled data to 2018 Hz

```
%(x): In the Power of Sepctral Density Resample Data of
2018 Hz we can see
%several narrowband noises. These narrowband noises could
be produce by
%electrical and mechanical resonances. On our loop below we
created the
%list of the highest peak frequencies that detect our
narrowband frequency
%ranges and the half of the maximum amplitude. Running our
loop it detected
%11 lines that could be the narrowband noises. We were able
to identify
%some source between the ranges of 0Hz-1200Hz due to
measured noise, expected noise
%& thermal noise.

% Finding peaks in the power spectral density plot.
[pks, locs, w, p] = findpeaks(10*log10(pxx_6));
pks = pks';
locs = locs';
w = w';
p = p';
p_sorted = sort(p, 'descend');
indx = zeros(1,11);
% Get the indices of the strongest narrowband lines.
for i = 1:11
    for j = 1:length(pks)
        if p_sorted(i) == p(j)
            indx(i) = locs(j);
        end
    end
end
freq_list = f6(indx);
w1 = w(w_indx);
%Full Width Half Maximum.
w1 = w1'/2;
%2X2 matrix with narrowband frequencies and full width half
maximum.
narrowband = [freq_list w1];
```