

Discussion

This homework was pretty straight forward once I understood how to find the Nyquist frequency and how to compute the power spectrum of the FFT of the signal. I originally was unable to determine what type of signal each data set was because I neglected to subtract the mean from the signal before computing the FFT. After correcting this, I can see that N1 is pink noise. This is because it has a more gradual return to white noise from the peak corresponding to $1/f$, and is much different than the sudden drop-off seen in N2. Based on the plot seen in Figure 4, I would say that the corner frequency for N1 is around 0.005 times the Nyquist frequency. I calculated the Nyquist frequency to be 500 Hz, so that means the corner frequency is 2.5 Hz. N2 appears to be a white noise signal with a sharp peak due to an added sine wave. The frequency of this peak denotes the frequency of the sine wave, which is 0.004 times the Nyquist frequency. This means that the sine wave oscillates at 2 Hz. Finally, N3 appears to be a simple white noise case, with no sharp peaks to speak of. I normalized the y-axis by dividing by N points and multiplying everything by 2 except the first and last points. I believe this is in accordance with Parseval's theorem.

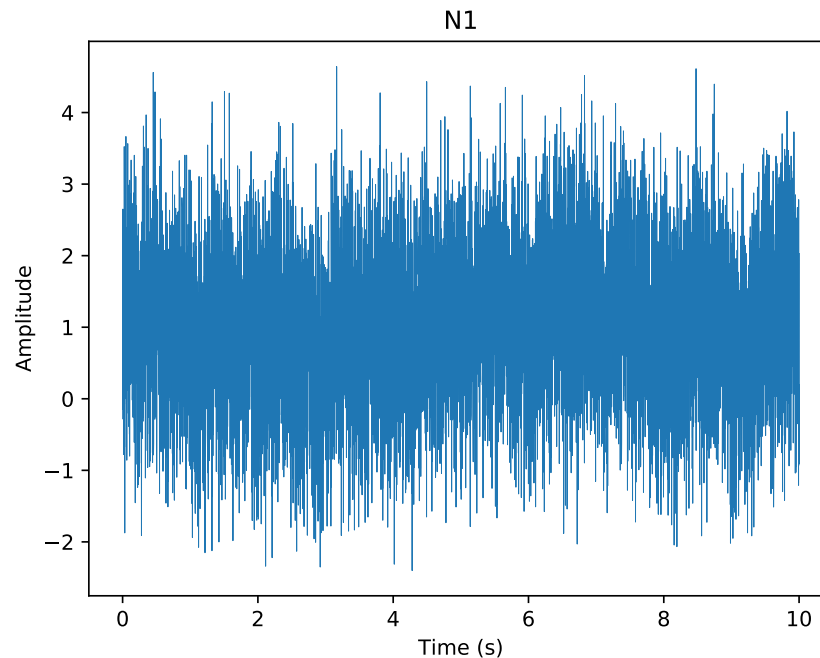


Figure 1

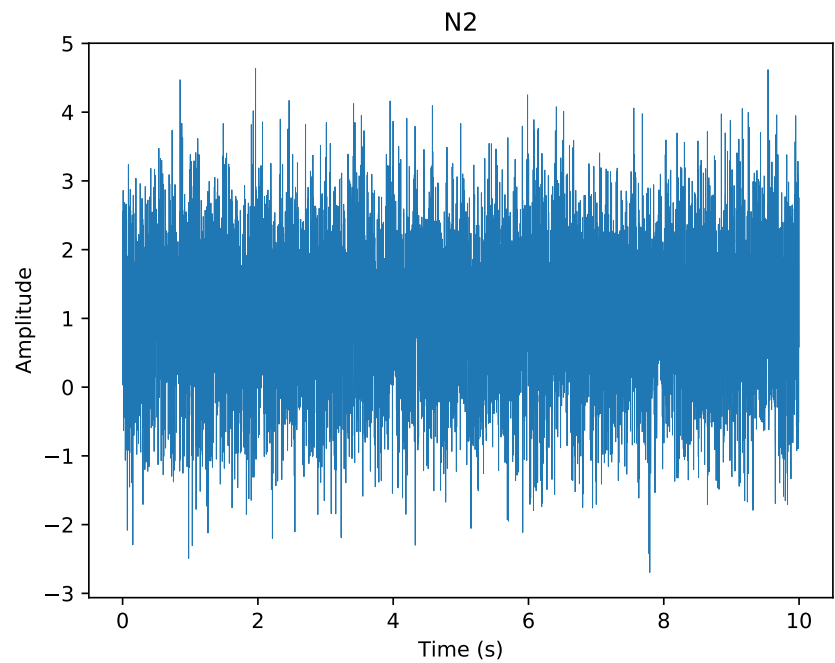


Figure 2

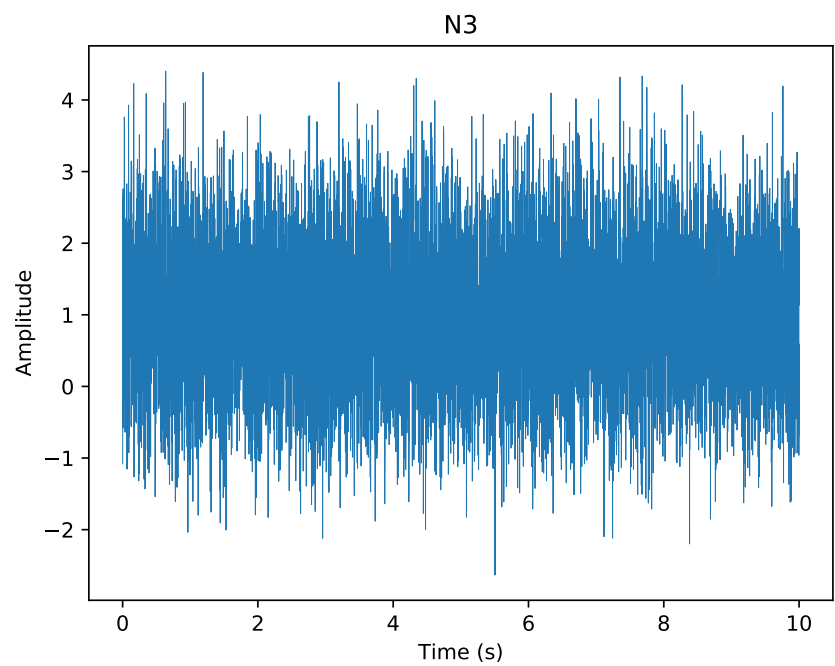


Figure 3

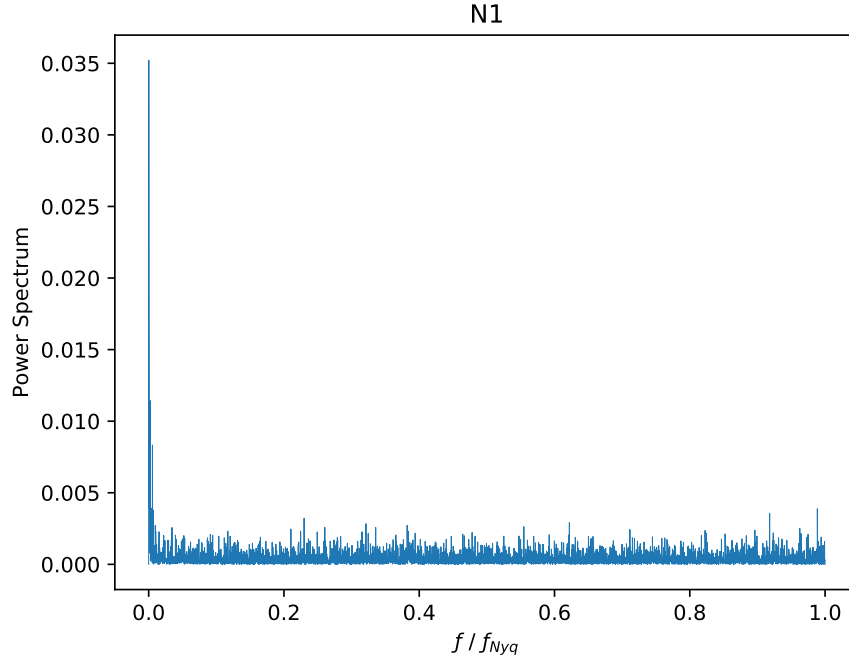


Figure 4: Power series of the FFT for N1. This signal is pink noise, as can be seen by the curved drop from the peak to white noise. The knee frequency is around 0.005 times the Nyquist frequency.

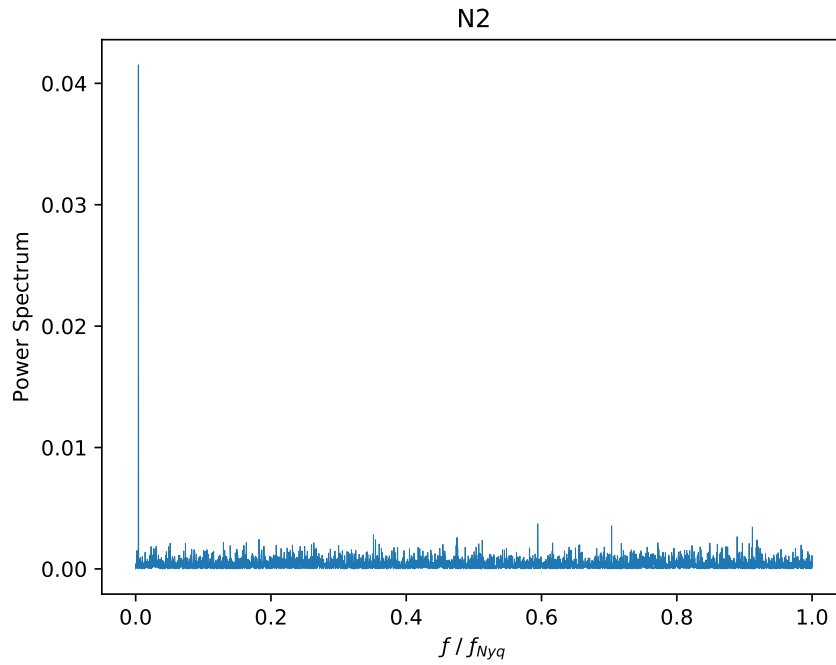


Figure 5: Power series of the FFT for N2. The large spike in the signal and then immediate return to white noise makes apparent that there is a sinusoidal signal added into regular white noise. Based on the peak, the sine wave has a frequency of 0.004 times the Nyquist frequency.

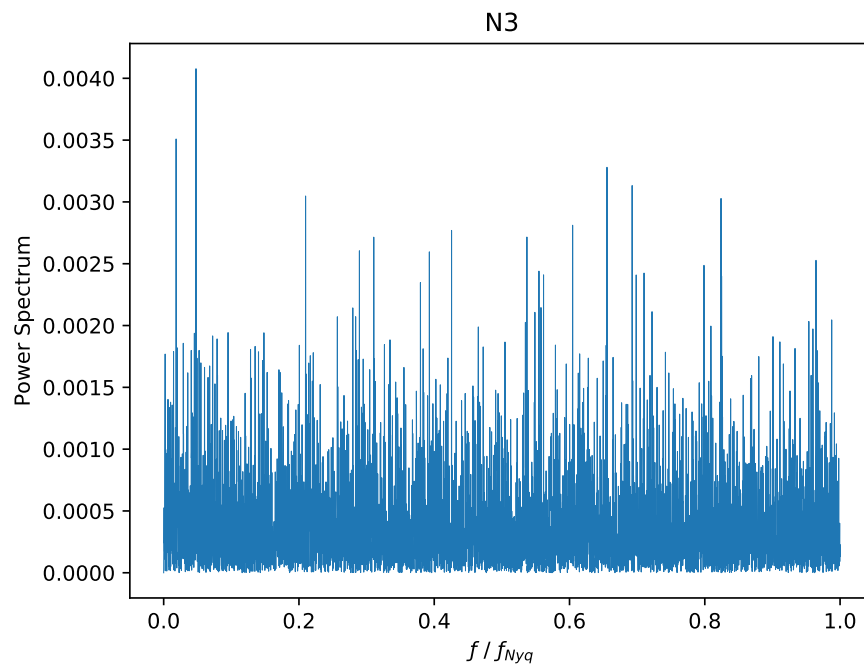


Figure 6: Power series of the FFT for N3. Nothing too special going on here, just white noise.

```

# -*- coding: utf-8 -*-

import numpy as np
import matplotlib.pyplot as plt

plt.close('all')

d1 = np.genfromtxt('noise1_trial.csv',delimiter=',',skip_header=1)
d2 = np.genfromtxt('noise2_trial.csv',delimiter=',',skip_header=1)
d3 = np.genfromtxt('noise3_trial.csv',delimiter=',',skip_header=1)

#Sampling rate and Nyquist frequency
Fs = abs(1/(d1[1,0]-d1[0,0]))
nyq = Fs/2
freq = np.arange(0,nyq+Fs/len(d1[:,0]),Fs/len(d1[:,0]))
freq = [a/nyq for a in freq]

#Initial Amplitude Plots
plt.figure()
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('N1')
plt.plot(d1[:,0],d1[:,1],linewidth=0.1)
plt.savefig('N1_amp.pdf')
plt.figure()
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('N2')
plt.plot(d2[:,0],d2[:,1],linewidth=0.1)
plt.savefig('N2_amp.pdf')
plt.figure()
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('N3')
plt.plot(d3[:,0],d3[:,1],linewidth=0.1)
plt.savefig('N3_amp.pdf')

f1 = np.fft.rfft(d1[:,1] - np.average(d1[:,1]))
f2 = np.fft.rfft(d2[:,1] - np.average(d2[:,1]))
f3 = np.fft.rfft(d3[:,1] - np.average(d3[:,1]))

f1 = [a*2/len(d1[:,0]) for a in f1]
f1[0] = f1[0]/2
f1[-1] = f1[-1]/2

f2 = [a*2/len(d2[:,0]) for a in f2]
f2[0] = f2[0]/2
f2[-1] = f2[-1]/2

f3 = [a*2/len(d3[:,0]) for a in f3]
f3[0] = f3[0]/2
f3[-1] = f3[-1]/2

##Nyquist cutoff
#f1 = f1[int(len(f1)/2):len(f1)]
#f2 = f2[int(len(f2)/2):len(f2)]

```

```
#f3 = f3[int(len(f3)/2):len(f3)]
```

```
p1 = [x*np.conj(x) for x in f1]  
p2 = [x*np.conj(x) for x in f2]  
p3 = [x*np.conj(x) for x in f3]
```

```
plt.figure()  
plt.plot(freq,p1,linewidth = 0.5)  
plt.xlabel('$f$ / $f_{Nyq}$')  
plt.ylabel('Power Spectrum')  
plt.title('N1')  
plt.savefig('N1_fft.pdf')
```

```
plt.figure()  
plt.plot(freq,p2,linewidth = 0.5)  
plt.xlabel('$f$ / $f_{Nyq}$')  
plt.ylabel('Power Spectrum')  
plt.title('N2')  
plt.savefig('N2_fft.pdf')
```

```
plt.figure()  
plt.plot(freq,p3,linewidth = 0.5)  
plt.xlabel('$f$ / $f_{Nyq}$')  
plt.ylabel('Power Spectrum')  
plt.title('N3')  
plt.savefig('N3_fft.pdf')
```