Learning Exercise 6: Power Spectrum Estimation

by

Jordan Hayes

EGR 434-01 Bioelectric Potentials

Date Assigned: October 14, 2021

Date Submitted: October 28, 2021

Instructor: Dr. Rhodes

**Objectives**

The objectives of this laboratory are to practice working in MATLAB environment, practice using frequency domain analysis tools in MATLAB, and to obtain and interpret power spectra of physiological data, as well as cross power spectrum density to analyze the shared frequency content of multiple signals. The frequency content of a signal provides more useful information than the time domain. Many biological signals demonstrate interesting and useful properties from a frequency domain perspective. Classical methods are based on the Fourier Transform shown below. The analysis of any signal in its time or frequency domain, as well as the PSD, can be used to draw inferences when perceived through the lens of bioelectric potentials.

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi f t} dt \qquad (1)$$

From Parseval's theorem, the auto power spectral density (PSD) of the signal can be estimated as Eq. 2.

$$P_{xx}(f) = X(f)X^*(f) = |X(f)|^2 \qquad (2)$$

The cross-power spectrum is useful in looking at the relationship between two time series, the process is shown in Eq. 3.

$$P_{xy}(f) = X(f)Y^*(f) = X^*(f)Y^*(f) \qquad (3)$$

**<u>Exercise 1:</u>**

For this part, signals were created, and the MATLAB function *fft()* was used to estimate the signal in the frequency domain. The frequency domain wave was used to estimate the auto power spectrum.

**Code and Custom Functions**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Jordan Hayes
%Lab 6:  Power SPectrum Estimation
%EGR434: Bioelectric Potentials
%Dr. Rhodes
%10/28/2021
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all
clear all
%Exercise 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%1. Create a 35 Hz sine wave, x(t), with amplitude 2.5V, sampled at
```

```
%fs = 200 Hz.
%Plotted vs. time - 3 seconds of this wave.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f = 35;
omega = 2 * pi * f;

A = 2.5;

f_s = 200;
T_s = 1/f_s;

t = 0:T_s:3;

x_t = A * sin(omega .* t);

figure(1)
subplot(2,1,1);
plot(t,x_t);
title('2.5V 35Hz Sine Wave Sampled at 200Hz');
xlabel('t (s)');
ylabel('x(t) (V)');
grid on
hold on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%2. Converted the signal, x(t), into the frequency domain, X(f)
%using the FFT (Fast Fourier Transform). Used
%the built-in Matlab command fft with an n=512 points. Plotted the PSD, |X(f)|2
% vs. frequency in Hz.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = 512;
x_freq = (0:n-1) * f_s / n;

X_f = fft(x_t, n);

X_f = abs(X_f).^2;

subplot(2,1,2);
plot(x_freq(1:n/2), X_f(1:n/2));
title('PSD of 2.5V 35Hz Sine Wave Sampled at 200Hz');
xlabel('Frequency (Hz)');
ylabel('Power Density (V^2 / Hz)^2');
grid on
hold off


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%3. repeat 1 and 2 for the following:
% 2V 20Hz sine wave
% 3V 15Hz sine wave
% fs = 150Hz
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f1 = 20;
omega1 = 2 * pi * f1;
A1 = 2;
```

```
f2 = 15;
omega2 = 2* pi * f2;
A2 = 3;

f_s = 150;
T_s = 1/f_s;

t = 0:T_s:3;

x_t = A1 * sin(omega1 .* t) + A2 * sin(omega2 .* t);

figure(2)
subplot(2,1,1);
plot(t,x_t);
title('2V 20Hz + 3V 15Hz Sine Wave Sampled at 150Hz');
xlabel('t (s)');
ylabel('x(t) (V)');
grid on
hold on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = 512;
x_freq = (0:n-1) * f_s / n;

X_f = fft(x_t, n);

X_f = abs(X_f).^2;

subplot(2,1,2);
plot(x_freq(1:n/2), X_f(1:n/2));            % / n to normalize?
title('PSD of 2V 20Hz + 3V 15Hz Sine Wave Sampled at 150Hz');
xlabel('Frequency (Hz)');
ylabel('Power Density (V^2 / Hz)^2');
grid on
hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%4.
% Repeat Part 1 and 2 for a sawtooth wave
% 1V amplitude
% 15Hz frequency and
% sampled at 200Hz
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f1 = 15;
omega1 = 2 * pi * f1;
A1 = 1;

f_s = 200;
T_s = 1/f_s;

t = 0:T_s:3;

x_t = sawtooth(omega1 .* t);
```

```
figure(3)
subplot(2,1,1);
plot(t,x_t);
title('1V 15Hz Sawtooth Wave Sampled at 200Hz');
xlabel('t (s)');
ylabel('x(t) (V)');
grid on
hold on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = 512;
x_freq = (0:n-1) * f_s / n;

X_f = fft(x_t, n);

X_f = abs(X_f).^2;

subplot(2,1,2);
plot(x_freq(1:n/2), X_f(1:n/2));             % / n to normalize?
title('PSD of 1V 15Hz Sawtooth Wave Sampled at 200Hz');
xlabel('Frequency (Hz)');
ylabel('Power Density (V^2 / Hz)^2');
grid on
hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%5. Repeat Part 1 and 2 for a square wave
% 1V amplitude
% 15Hz frequency and
% sampled at 200Hz
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f1 = 15;
omega1 = 2 * pi * f1;
A1 = 1;

f_s = 200;
T_s = 1/f_s;

t = 0:T_s:3;

x_t = square(omega1 .* t, 50);

figure(4)
subplot(2,1,1);
plot(t,x_t);
title('1V 15Hz Square Wave Sampled at 200Hz');
xlabel('t (s)');
ylabel('x(t) (V)');
grid on
hold on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = 512;
x_freq = (0:n-1) * f_s / n;

X_f = fft(x_t, n);
```

```
X_f = abs(X_f).^2;

subplot(2,1,2);
plot(x_freq(1:n/2), X_f(1:n/2));            % / n to normalize?
title('PSD of 1V 15Hz Square Wave Sampled at 200Hz');
xlabel('Frequency (Hz)');
ylabel('Power Density (V^2 / Hz)^2');
grid on
hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%6. Repeat Part 1 and 2 for:
% 2V 70Hz sine wave
% fs = 150Hz
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f1 = 70;
omega1 = 2 * pi * f1;
A1 = 2;

f_s = 150;
T_s = 1/f_s;

t = 0:T_s:3;

x_t = A1 * sin(omega1 .* t);

figure(5)
subplot(2,1,1);
plot(t,x_t);
title('2V 70Hz Sine Wave Sampled at 150Hz');
xlabel('t (s)');
ylabel('x(t) (V)');
grid on
hold on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = 512;
x_freq = (0:n-1) * f_s / n;

X_f = fft(x_t, n);

X_f = abs(X_f).^2;

subplot(2,1,2);
plot(x_freq(1:n/2), X_f(1:n/2));            % / n to normalize?
title('PSD of 2V 70Hz Sine Wave Sampled at 150Hz');
xlabel('Frequency (Hz)');
ylabel('Power Density (V^2 / Hz)^2');
grid on
hold off
```

**Output**

Each plot displays a 3 second duration of the created signals as well as its power spectral density; descriptions for the plots appear below figures.

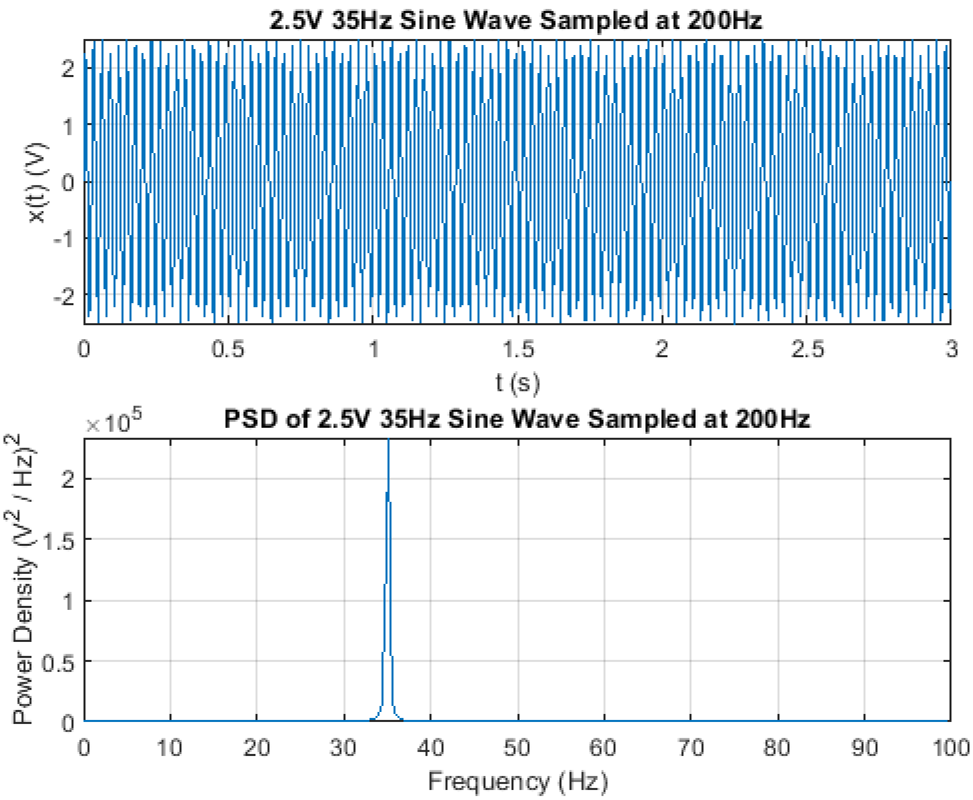


**Figure 1: Time Domain and PSD Plots for 2.5V 35Hz Signal $f_s = 200Hz$**

Figure 1 above displays a simple 35 Hz sinusoid with an amplitude of 2.5V. It can be seen that the auto power density spectrum contains the main frequency that the sinusoid was created with at 35 Hz. The peak of the PSD is at 35Hz with an amplitude to an order determined by the amplitude of the original signal, 2.5. In general, the PSD is used to show the power present in the signal as a function of frequency. A simple sinusoid like the one shown above has a frequency response that is an impulse function at its fundamental frequency.

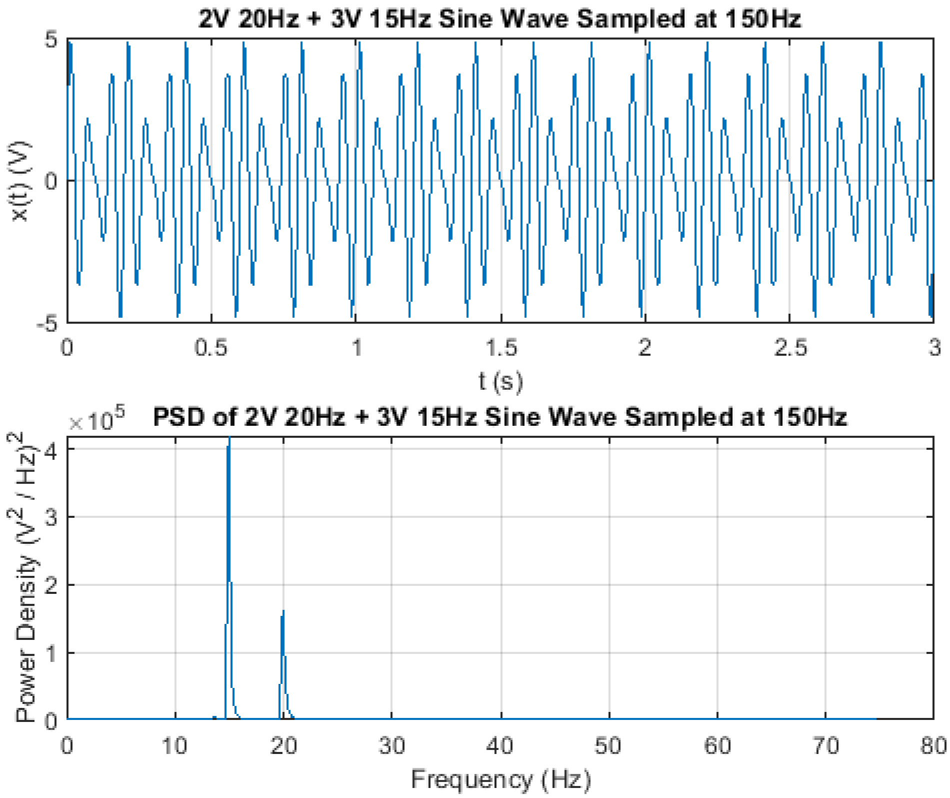


**Figure 2: Time Domain and PSD Plots for 2V 20Hz + 3V 15Hz Signal $f_s = 150Hz$**

Figure 2 above shows a signal that is the summation of two different sinusoids. One is a 2V 20Hz; the other, a 3V 15Hz. This is not obvious from the time domain plot, showing the usefulness of the frequency domain response. The frequencies that appear on the lower plot of the PSD are expected, then. The spikes in power are found at 15 and 20 Hz; however, the 15Hz sinusoid contributes more to the power in the signal because of the higher amplitude and units of the y axis. Differences in power corresponding to the frequency by which they are produced are not linear, as an increase in amplitude in an original contributing signal represents an exponential increase in power.

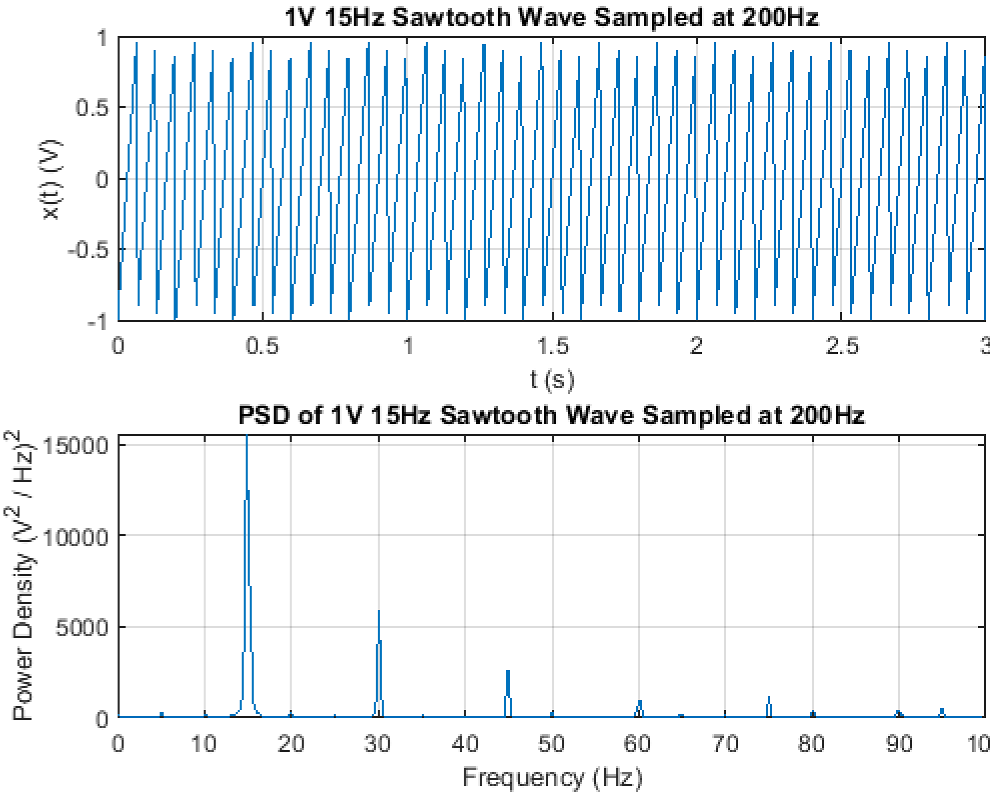


**Figure 3: Time Domain and PSD Plots for 1V 15Hz Triangle Wave with $f_s = 200Hz$**

The figure above is created using the same process as in Figures 1 and 2, but with a sawtooth-type triangle wave. The signal seems simple in the time domain, but the PSD plot shows that the signal is created through the contribution of multiple frequencies. The strongest is the fundamental frequency 15 Hz. The slope of the time domain signal is then formed because of the summation of other present frequencies. The PSD plot shows the frequencies clearly, with their power contributions shown as integer multiples of the fundamental frequency in descending order. The vertical portion of the wave, or as close as MATLAB can get, starts at a peak amplitude of 1, and drops to the negative amplitude. This is only possible if the frequencies that contribute to this signal are harmonics.

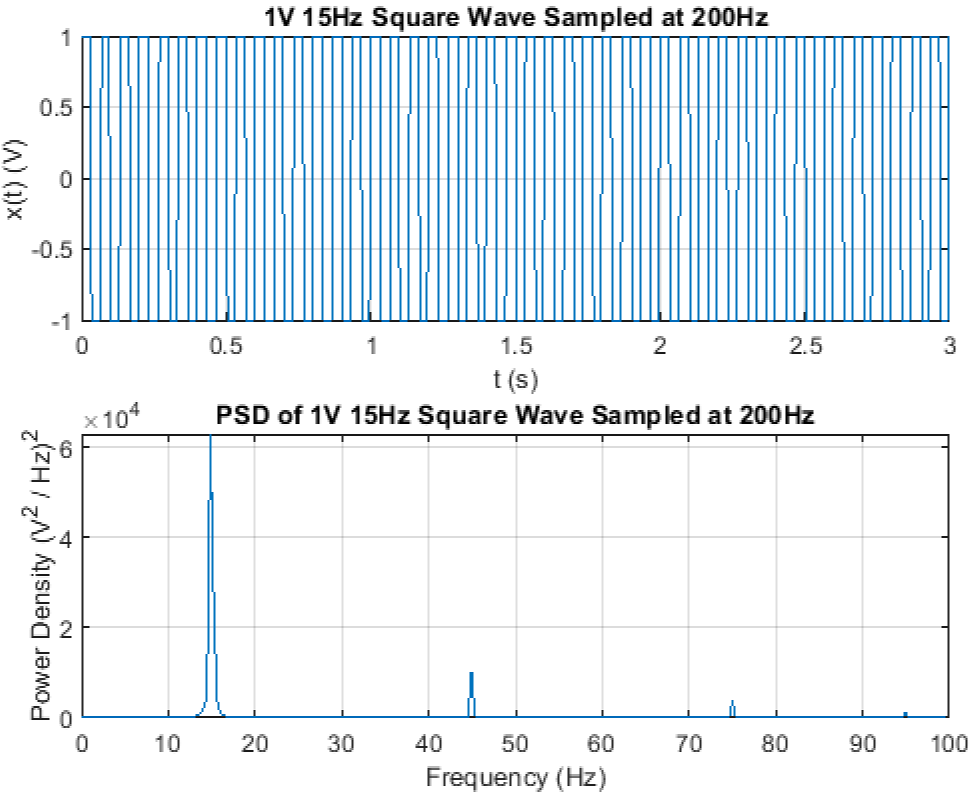


**Figure 4: Time Domain and PSD Plots for 1V 15Hz Square Wave with $f_s = 200Hz$**

The figure above shows a square wave with a fundamental frequency of 20 Hz and an amplitude of 1. The PSD of this type of wave is similar to that in Figure 3 by having multiple frequencies that make up the wave. The other frequencies shown in the PSD plot are positive odd integers, skipping every other multiple, causing the flattened peak(s) and virtually undefined slope for the edges.

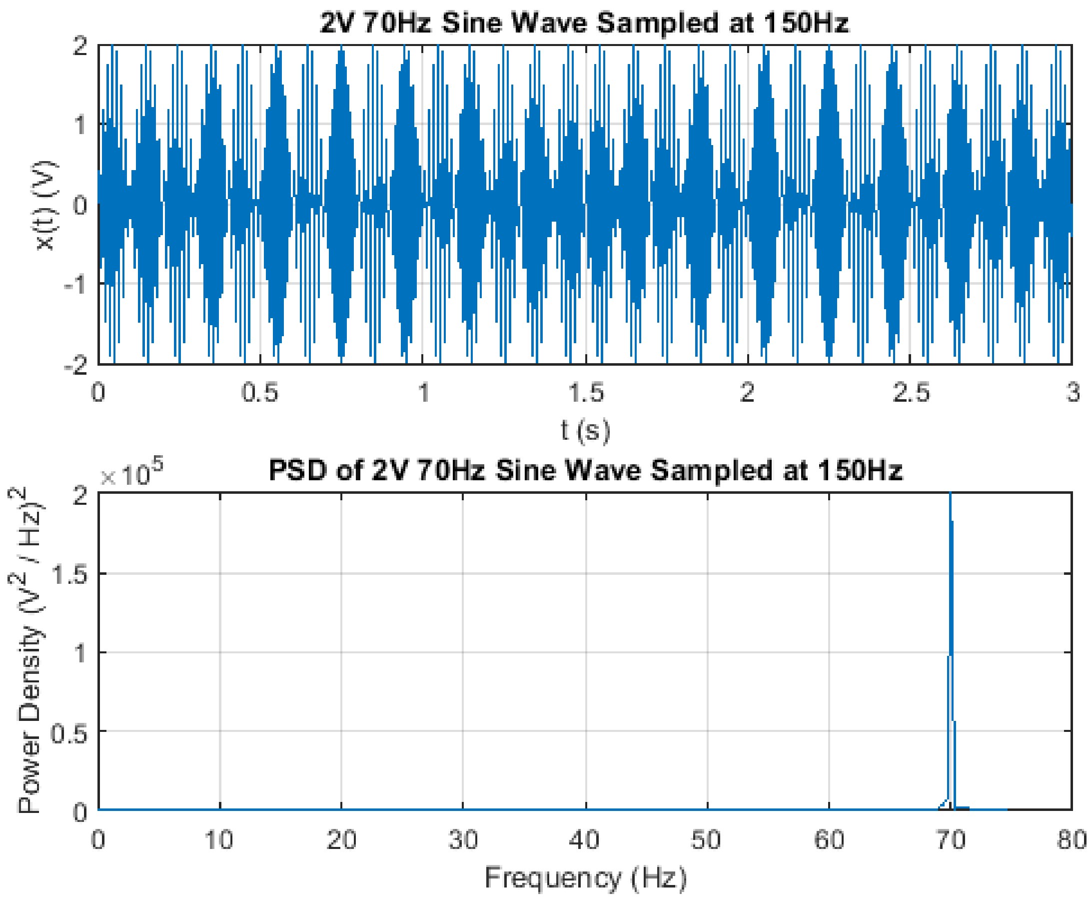


**Figure 5: Time Domain and PSD Plots for 2V 70Hz Sine Wave with $f_s = 150Hz$**

The plots in Figure 5 above are similar to Figure 1 in which there is only one frequency that makes up this signal. The fundamental frequency is 70Hz, shown by the large peak is the PSD at 70Hz. The contrast of the plot above to Figure 1 is the sampling frequency. Nyquist's Theorem states that the sampling frequency of a signal must be at least twice the signal's fundamental frequency. The highest frequency that can be detected by 150Hz sampling is 75Hz, which is greater than 70Hz; therefore, the signal is not aliased in the frequency domain.

## <u>Exercise 2:</u>

### Code and Custom Functions

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Jordan Hayes
%Lab 6:  Power SPectrum Estimation
%EGR434: Bioelectric Potentials
%Dr. Rhodes
%10/28/2021
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
close all

%Exercise 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%1. Create a 2.5V, 35 Hz sine wave + 3V 15 Hz sine wave,
%y(t), sampled at fs = 200 Hz. Plot this signal vs.
%time - generate at least 3 seconds of this wave.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f1 = 35;
omega1 = 2 * pi * f1;
A1 = 2.5;

f2 = 15;
omega2 = 2* pi * f2;
A2 = 3;

f_s = 200;
T_s = 1/f_s;

t = 0:T_s:3;

x_t1 = A1 * sin(omega1 .* t) + A2 * sin(omega2 .* t);

figure(1)
subplot(2,1,1);
plot(t,x_t1);
title('2.5V 35Hz + 3V 15Hz Sine Wave Sampled at 200Hz');
xlabel('t (s)');
ylabel('x(t) (V)');
grid on
hold on

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%2. Create a 1.5V, 35 Hz sine wave + 2V 50 Hz sine wave,
%y(t), sampled at fs = 200 Hz. Plot this signal vs.
%time - generate at least 3 seconds of this wave.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f1 = 35;
omega1 = 2 * pi * f1;
A1 = 1.5;
```

```
f2 = 50;
omega2 = 2* pi * f2;
A2 = 2;

f_s = 200;
T_s = 1/f_s;

t = 0:T_s:3;

x_t2 = A1 * sin(omega1 .* t) + A2 * sin(omega2 .* t);

subplot(2,1,2);
plot(t,x_t2);
title('1.5V 35Hz + 2V 50Hz Sine Wave Sampled at 200Hz');
xlabel('t (s)');
ylabel('x(t) (V)');
hold off

sgtitle('Exercise 2: Parts 1 and 2');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%3. Estimate the auto-power spectrum Pxx and Pyy and plot vs.
% frequency in Hz
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n = 512;
x_freq1 = (0:n-1) * f_s / n;
X_f1 = fft(x_t1, n);
X_f1_xx = abs(X_f1).^2;


x_freq2 = (0:n-1) * f_s / n;
X_f2 = fft(x_t2, n);
X_f2_xx = abs(X_f2).^2;

figure(2)
subplot(2,1,1);
plot(x_freq1(1:n/2), X_f1_xx(1:n/2));
title('PSD (P_x_x) of 2.5V 35Hz + 3V 15Hz Sine Wave Sampled at 200Hz');
xlabel('Frequency (Hz)');
ylabel('Power Density (V^2 / Hz)^2');
grid on

subplot(2,1,2);
plot(x_freq2(1:n/2), X_f2_xx(1:n/2));
title('PSD (P_y_y) of a 1.5V 35Hz + 2V 50Hz Sine Wave Sampled at 200Hz');
xlabel('Frequency (Hz)');
ylabel('Power Density (V^2 / Hz)^2');
grid on

sgtitle('Auto Power Spectrum of Parts 1 and 2');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%4. Estimate the cross-power spectrum and plot the absolute
% value of Pxy vs. frequency in Hz
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%use the frequency domain
X_f2_Conj = conj(X_f2);

P_xy = X_f2_Conj .* X_f1;

figure(3)
grid on
plot(x_freq1(1:n/2), P_xy(1:n/2));
title('Cross Power Density Spectrum (P_x_y)');
xlabel('Frequency (Hz)');
ylabel('Power Density (V^2 / Hz)^2');
```

**Output**

For this exercise, each plot displays a 3 second duration of the created signals and focuses on analyzing the result of the PSD of the signals individually, then the cross power spectrum of the signals combined.

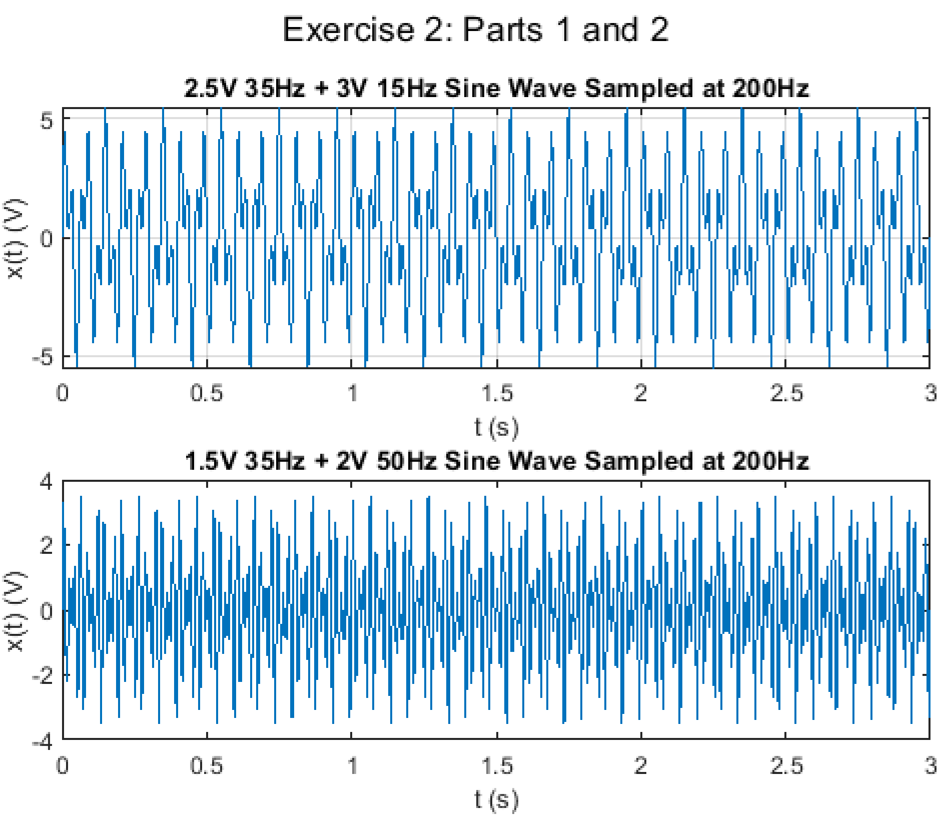


**Figure 6: Time Domain Plots for Combined Signals in Parts 1 and 2**

Much like Figure 2, the figure above shows two individual signals which are summations of two signals described by the plot titles. The 3V 15Hz and 2V 50Hz signals have higher amplitudes compared to their complimentary signals that they combine with. This should be shown in the PSD plots.

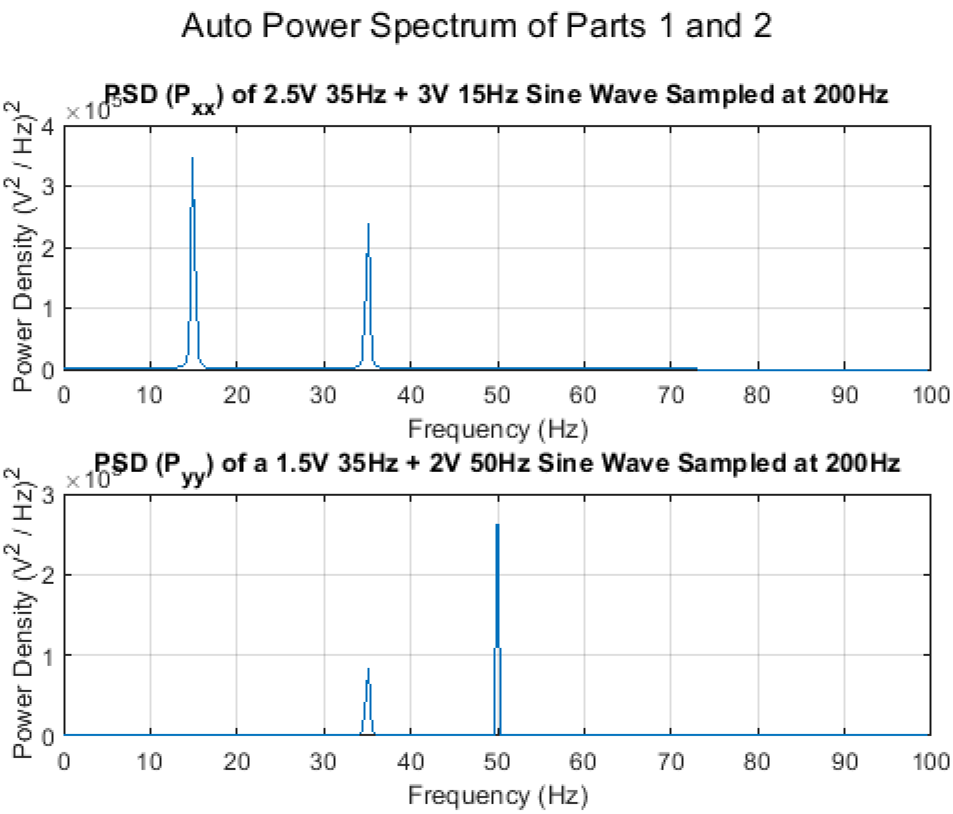


**Figure 7: Power Spectrum Density Plots for Combined Signals in Parts 1 and 2**

From Figure 7, it can be seen that the prediction made from Figure 6 is correct. Power output is much greater in the top plot for 15Hz, and 50Hz power is greater on the bottom. There is a greater difference in the bottom plot between the two powers because the PSD does not have a linear relationship with amplitude, but will have a much greater difference as voltage magnitude increases when combined with other signals; however, it is not always possible to classify the source of the power accurately using these methods. Both plots share 35Hz as a fundamental frequency.

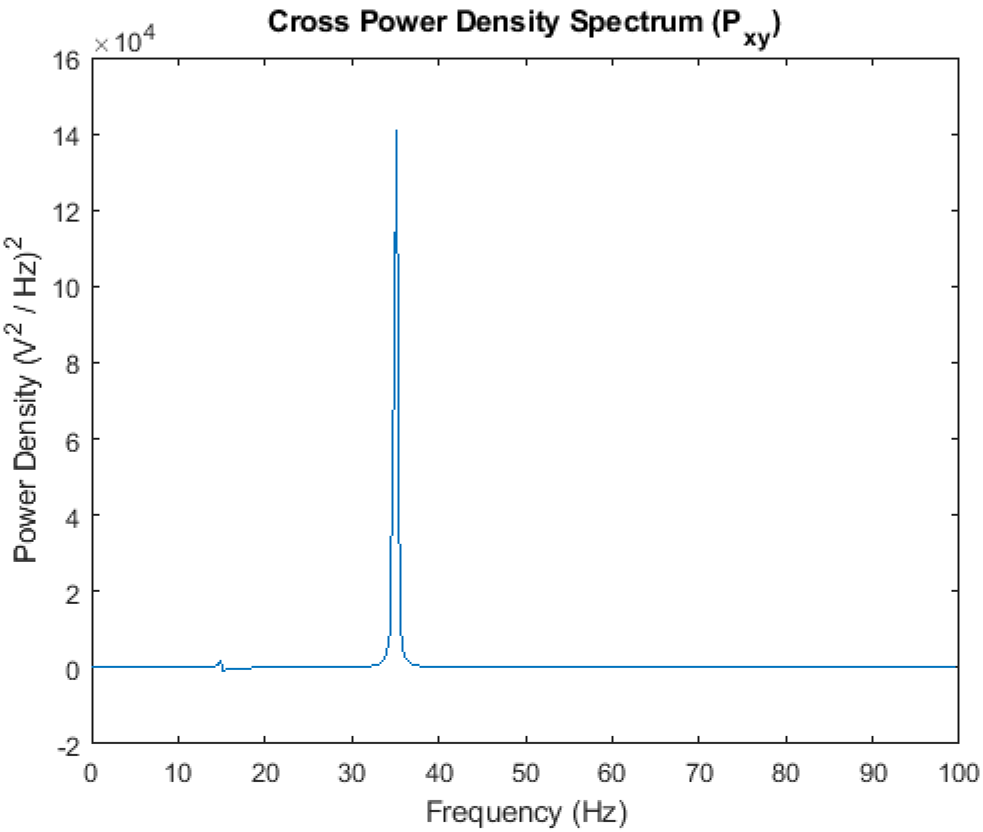


**Figure 8: Cross Power Spectrum Density Plot for Combined Signals in Parts 1 and 2**

The figure above displays the cross-power spectrum density of the two signals from Figure 6. This operation is intended to show the frequencies shared between the two signals. This information can be used to determine the influence of a signal in relation to another, in this case, 35Hz. The plot confirms that 35Hz is a shared fundamental frequency. The magnitude of the power is decreased, however.

## Exercise 3:

### Code and Custom Functions

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Jordan Hayes
%Lab 6:  Power SPectrum Estimation
%EGR434: Bioelectric Potentials
%Dr. Rhodes
%10/28/2021
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all
clear all

%Exercise 3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%1. Plot raw data vs 10 seconds of time
```

```
ECG_Resting_dataTable = readtable("EGR434_Project1_Resting_Jordan.txt");

ECG_PostExercise_dataTable = readtable("EGR434_Project1_PostExercise_Jordan.txt");

t = ECG_Resting_dataTable(:,1);
t = table2array(t);
t = t(1:2001,1);

Resting_VolPulse = ECG_Resting_dataTable(:,2);
Resting_VolPulse = table2array(Resting_VolPulse);

Resting_BloodFlow = ECG_Resting_dataTable(:,3);
Resting_BloodFlow = table2array(Resting_BloodFlow);

Resting_ECG = ECG_Resting_dataTable(:,4);
Resting_ECG = table2array(Resting_ECG);
Resting_ECG = Resting_ECG(1:2001,1);
Resting_ECG(isnan(Resting_ECG)) = 0;

figure(1)
subplot(2,1,1);
plot(t, Resting_ECG);
grid on;
xlabel('t (s)');
ylabel('ECG (uV)');
title('Resting ECG vs Time');


t = ECG_PostExercise_dataTable(:,1);
t = table2array(t);
t = t(1:2001,1);

PostExercise_VolPulse = ECG_PostExercise_dataTable(:,2);
PostExercise_VolPulse = table2array(PostExercise_VolPulse);

PostExercise_BloodFlow = ECG_PostExercise_dataTable(:,3);
PostExercise_BloodFlow = table2array(PostExercise_BloodFlow);

PostExercise_ECG = ECG_PostExercise_dataTable(:,4);
PostExercise_ECG = table2array(PostExercise_ECG);
PostExercise_ECG = PostExercise_ECG(1:2001,1);
PostExercise_ECG(isnan(PostExercise_ECG)) = 0;

subplot(2,1,2);
plot(t, PostExercise_ECG);
grid on;
xlabel('t (s)');
ylabel('ECG (uV)');
title('Post Exercise ECG vs Time');
sgtitle('Resting and Post Exercise ECG Data');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%2. Zero Mean Data
Resting_ECG = Resting_ECG - mean(Resting_ECG);
PostExercise_ECG = PostExercise_ECG - mean(PostExercise_ECG);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%3. Obtain PSD of the resting and exercise ECG
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

T_s = .005;
f_s = 1/T_s;

n = 1024;
freq = (0:n-1) * f_s / n;
%freq = freq(1,1:257);

RestingECG_Xf = fft(Resting_ECG, n);
RestingECG_Xf_xx = abs(RestingECG_Xf).^2;
%RestingECG_Xf_xx = RestingECG_Xf_xx(1:257, 1);

PostExerciseECG_Xf = fft(PostExercise_ECG, n);
PostExerciseECG_Xf_xx = abs(PostExerciseECG_Xf).^2;
%PostExerciseECG_Xf_xx = PostExerciseECG_Xf_xx(1:257,1);



figure(2)
subplot(2,1,1);
plot(freq(1,1:257), RestingECG_Xf_xx(1:257, 1));
grid on;
xlabel('Frequency (Hz)');
ylabel('Power Density (V^2 / Hz)^2');
title('Resting ECG Power Desnity Spectrum vs Frequency');

subplot(2,1,2);
plot(freq(1,1:257), PostExerciseECG_Xf_xx(1:257, 1));
title('Post Exercise ECG PSD vs Frequency');
xlabel('Frequency (Hz)');
ylabel('Power Density (V^2 / Hz)^2');
hold off

sgtitle('PSDs Of Resting and Post Exercise ECG Data');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%6.
n = 1024;
freq = (0:n-1) * f_s / n;

RestingECG_diff = diff(Resting_ECG);
RestingECG_Xf_diff = fft(RestingECG_diff, n);
RestingECG_Xf_diff_xx = abs(RestingECG_Xf_diff).^2;

PostExerciseECG_diff = diff(PostExercise_ECG);
```

```
PostExerciseECG_Xf_diff = fft(PostExerciseECG_diff, n);
PostExerciseECG_Xf_diff_xx = abs(PostExerciseECG_Xf_diff).^2;

figure(3)
subplot(2,1,1);
plot(freq, RestingECG_Xf_diff_xx);
xlim([0, f_s/2])
grid on;
xlabel('Frequency (Hz)');
ylabel('Power Density (V^2 / Hz)^2');
title('Resting ECG PSD Derivative vs Frequency');

subplot(2,1,2);
plot(freq, PostExerciseECG_Xf_diff_xx);
xlim([0, f_s/2])
title('Post Exercise ECG PSD Derivative vs Frequency');
xlabel('Frequency (Hz)');
ylabel('Power Density (V^2 / Hz)^2');
hold off

sgtitle('PSD Derivatives Of Resting and Post Exercise ECG Data');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%7. Find the median frequency of resting ECG and exercise ECG

medianFreq_Resting = medfreq(RestingECG_Xf_xx, freq);
medianFreq_PostExercise = medfreq(PostExerciseECG_Xf_xx, freq);

fprintf("The median resting ECG frequency is %0.3f,\n", medianFreq_Resting);
fprintf("The median post exercise ECG frequency is %0.3f.\n", medianFreq_PostExercise);
```

```
The median resting ECG frequency is 91.192,
The median post exercise ECG frequency is 27.230.
```

### Output

For this exercise, raw ECG data was taken in class and analyzed using tools practiced above.

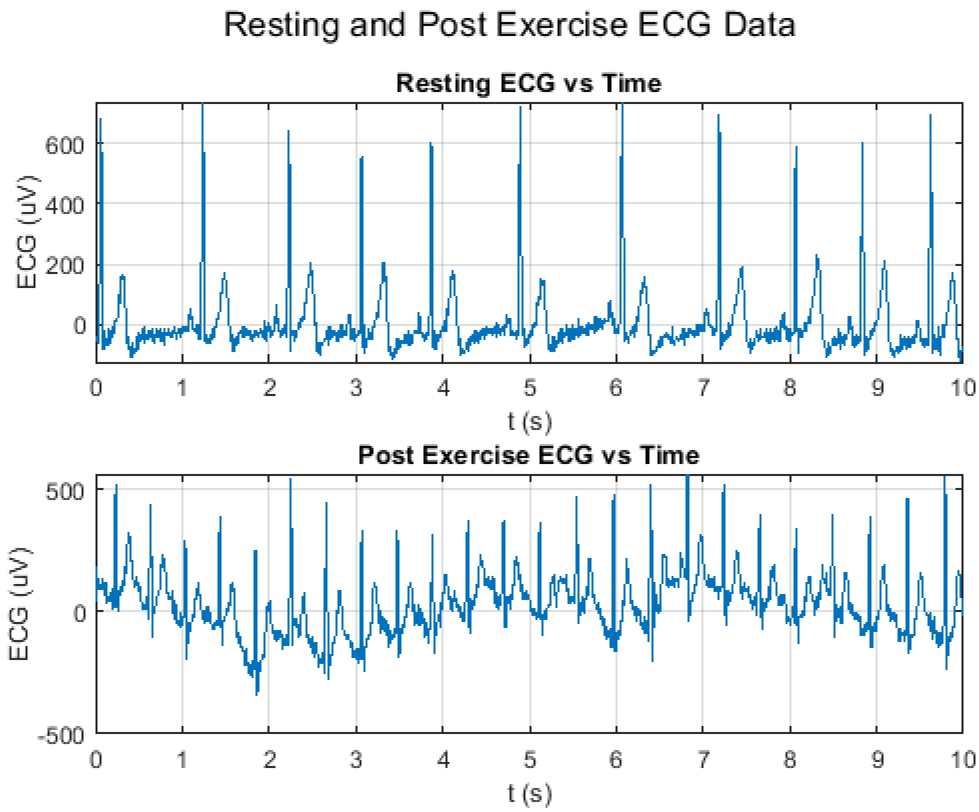


**Figure 9: Resting and Post Exercise ECG Data vs. Time**

The figure above displays the zero-mean raw data for resting and post exercise ECG. From the time domain plots, it can be seen that the resting and post exercise data differ in frequency and amplitude of Q peaks the most. The resting data has more stable, distinguishable QRS events with higher electrical output, while after exercise the electrical output becomes more erratic and resting voltages seem to oscillate, the QRS events are about twice as frequent, and the magnitude drops from 575-650 uV resting Q peak to 300-500 uV post exercise. QRS events happen about once every second in resting data, and twice every second in post exercise.

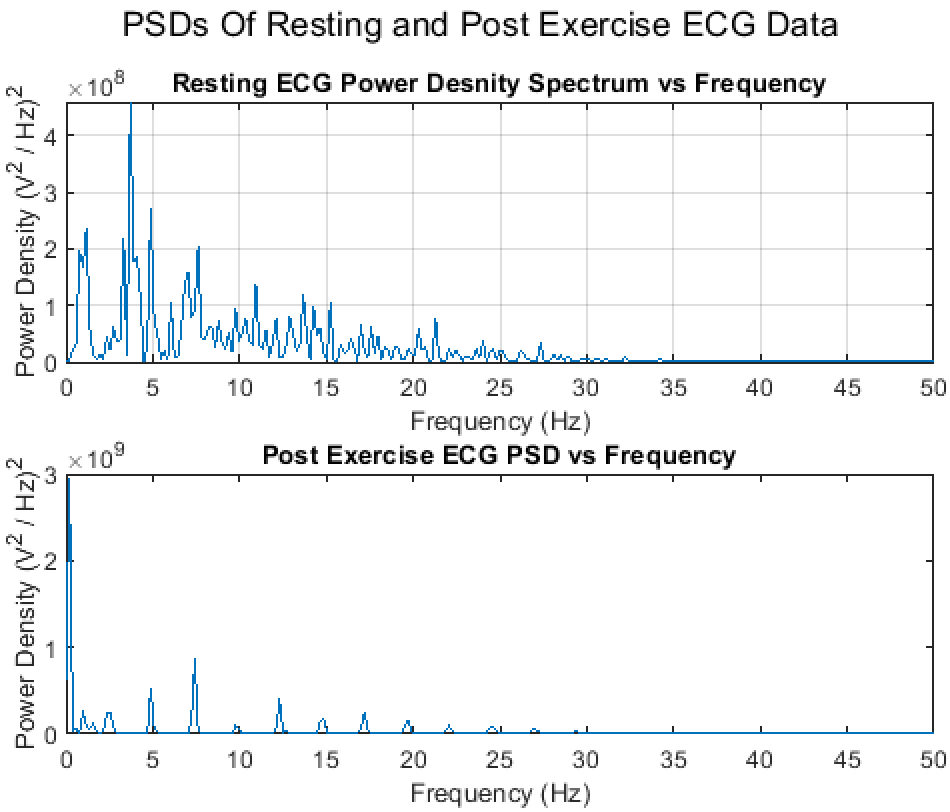


**Figure 10: Resting and Post Exercise ECG Power Spectrum Densities**

The figure above displays the PSD for both trials of data, and shows that the signals are made up of many different frequencies. Both have the majority of their power contributed by frequencies below 10Hz. In the post exercise data, these frequencies are more distinct and specific frequencies up to 30Hz, while the resting data seems to have contributions from all frequencies up to 30Hz and even beyond. The largest peak at very low frequencies in post exercise data is due to the slow oscillation of resting potential, which theoretically should be at 0V, and are included in the PSD as summations of very low frequencies. Both PSDs have the similar characteristic of square and triangle wave PSDs in that the frequencies that contribute to the signals are greatest in power at lower frequencies and decrease as frequency increases, but the post exercise PSD shows more similarity due to the more distinct frequencies that exist.

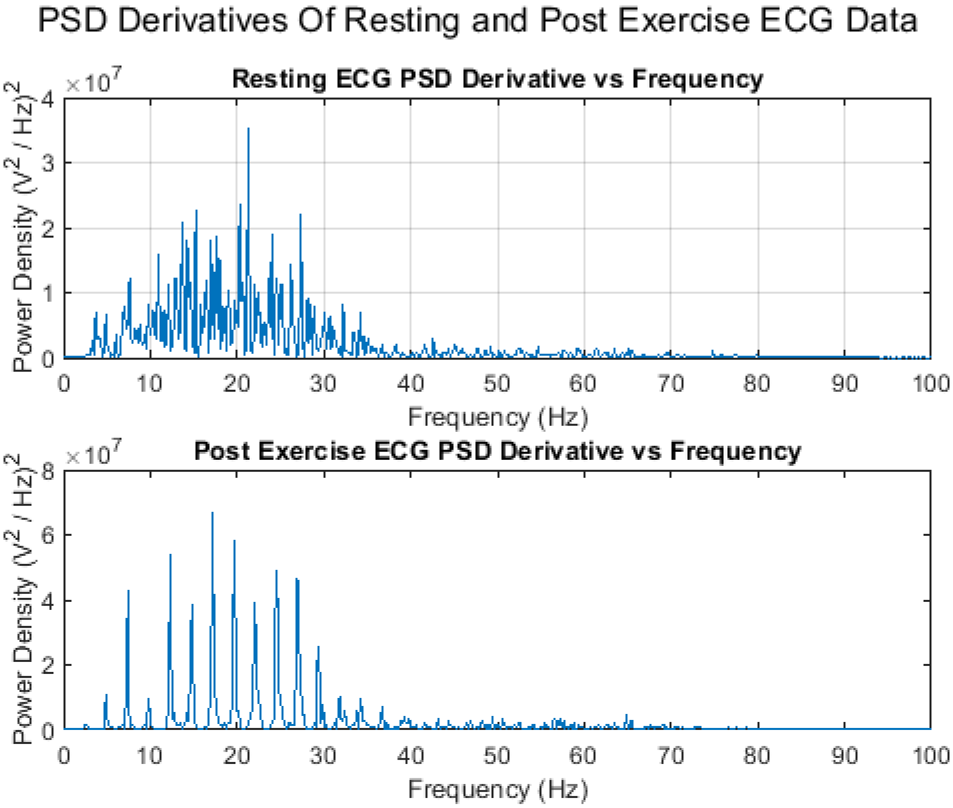


**Figure 11: Resting and Post Exercise ECG Derivative Power Spectrum Densities**

The subplot above shows the PSDs of the derivatives of the raw data.

***Question: Now take the first derivative of your resting and exercise ECG. Plot the PSD of the resulting signals and comment on what you see. What has taking the derivative done to your power spectrum?***

Taking the first derivative of the raw ECG data further exemplifies the high slopes of the ECG data. This operation makes the QRS complexes more distinct in comparison to small oscillations. Because of this, the derivative reveals more about the natural behavior in the heart. Taking the derivative clumped more active frequencies together in the resting ECG between 10 and 30Hz; for the post exercise data, the same band of frequencies was shown but remained more distinct; this is because the large spike in power at frequencies < 5Hz were removed and the plot was able to scale down the y axis to show the frequencies in better detail.

```
The median resting ECG frequency is 91.192,
The median post exercise ECG frequency is 27.230.
```

**Figure 12: Resting and Post Exercise**

***Question: Find the median frequency of resting ECG and exercise ECG. Are they different? Speculate on why or why not.***

Yes, the median frequencies are different in the resting and post exercise ECGs because of the change in function of the heart from maintaining stability in the muscles and body to pumping blood and feeding the body as efficiently as possible during and after exercise. The median frequency also helps to generalize the frequencies of the heart output.  The resting ECG power spectrum has a median frequency close to *fs/2*, indicating a more balanced sprecturm. In contrast, the post exercise data seems to be more densely made up of lower frequency signals, indicating an increase in heartrate