MAE 3113
MEASUREMENTS AND INSTRUMENTATIONS
Spring 2018
Lab 6
04/27/2018
Fourier Transforms

Group 25
Team Members

Jonathan C. Cook
Ben Davis
Patrick Steichen
Gerardo Toth
Colton Tubbs
Garrett O. Weber

<u>Experiment Performed by:</u>

Jonathan C. Cook
Ben Davis
Patrick Steichen
Gerardo Toth
Colton Tubbs
Garrett O. Weber

<u>Report Prepared by:</u>

Jonathan C. Cook
Ben Davis
Patrick Steichen
Gerardo Toth
Colton Tubbs
Garrett O. Weber

Summary of Contributions

Jonathan C. Cook: Completed the section covering Short-Time Fourier Transforms (STFT).

Ben Davis: Completed the section covering Aircraft Vibration.

Patrick Steichen: Completed the section covering Simple Waveforms / Pure Tones

Gerardo Toth: Completed the sections Beat and Mix.

Colton Tubbs: Completed the section covering Victor's Tone.

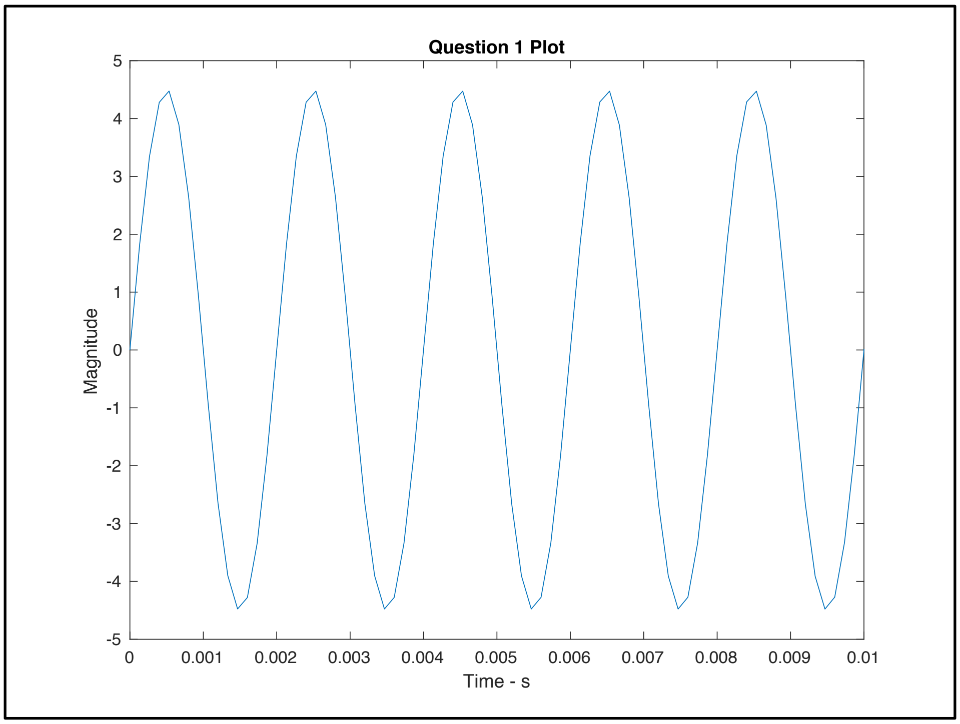
Garrett O. Weber: Completed the section covering Car Engine's Data.

Simple Waveforms / Pure Tones

1. **Using MATLAB or Excel, create a simple harmonic waveform of the following signal for 5,000 data points, with a time step of at least 5,000 Hz. You should include a copy of the code used.**

MATLAB Example Code

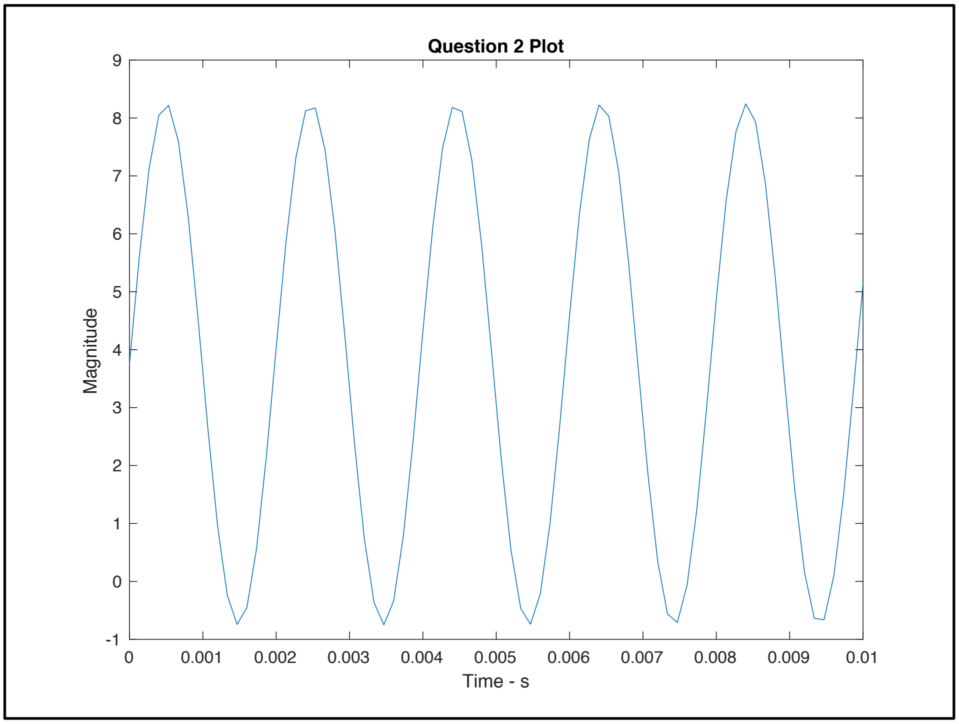
```
% Time Domain
N=5000;
fs=7500;
T=N/fs;
t = linspace(0,T,5000);
y1 = 4.5.*sin(2*pi*500*t) + 0;
y2 = 4.5.*sin(2*pi*505*t) + 3.75;

% Part 1 Time Domain
plot(t,y1)
xlim([0 0.01])
title('Question 1 Plot')
xlabel('Time - s')
ylabel('Magnitude')
pause
```



2. Create a second waveform, shown below, using the same sample conditions as the signal above.

MATLAB Example Code

```
% Time Domain
N=5000;
fs=7500;
T=N/fs;
t = linspace(0,T,5000);
y1 = 4.5.*sin(2*pi*500*t) + 0;
y2 = 4.5.*sin(2*pi*505*t) + 3.75;

% Part 1 Time Domain
plot(t,y1)
xlim([0 0.01])
title('Question 1 Plot')
xlabel('Time - s')
ylabel('Magnitude')
pause

% Part 2 Time Domain
plot(t,y2)
xlim([0 0.01])
title('Question 2 Plot')
xlabel('Time - s')
ylabel('Magnitude')
pause
```



3. Using MATLab, convert the signals in parts 1 and 2 into the frequency domain using the FFT. You should include a copy of the code used.

MATLAB Example Code

```
% Part 1 FFT
Y1 = fft(y1);
n1 = length(y1)/2; n1 = ceil(n1);
amp_spec1 = abs(Y1)/n1;
freq1 = linspace(0,fs/2,n1);

% Part 2 FFT
Y2 = fft(y2);
n2 = length(y2)/2; n2 = ceil(n2);
amp_spec2 = abs(Y2)/n2;
% Adjust first point for DC offset
amp_spec2(1) = amp_spec2(1)/2;
freq2 = linspace(0,fs/2,n2);
```

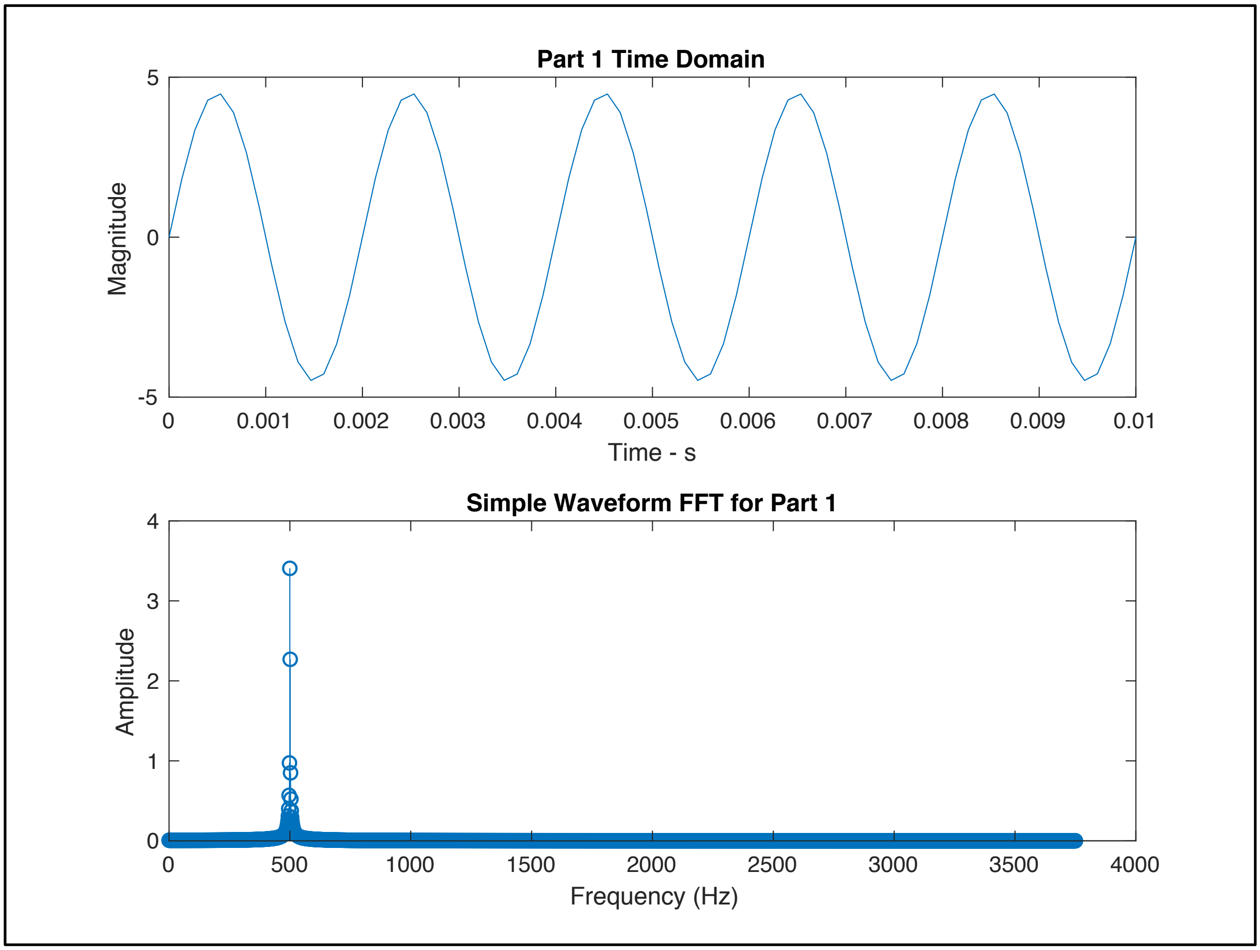
NOTE: Please see the two FFT plots in Question 4 below.

4. Create a subplot that contains the time domain in one plot and the frequency domain in the second plot, for each signal. (one set for each equation)

a) In the time domain, the graph should be time on the x-axis and magnitude on the y-axis. You should trim your data down to show only five to eight typical periods.

MATLAB Example Code

```
% Part 4a
subplot(2,1,1);
plot(t,y1)
xlim([0 0.01])
xlabel('Time - s')
ylabel('Magnitude')
title('Part 1 Time Domain')

subplot(2,1,2);
stem(freq1,amp_spec1(1:(n1)));
xlabel('Frequency (Hz)'); ylabel('Amplitude');
title('Simple Waveform FFT for Part 1');
pause
```

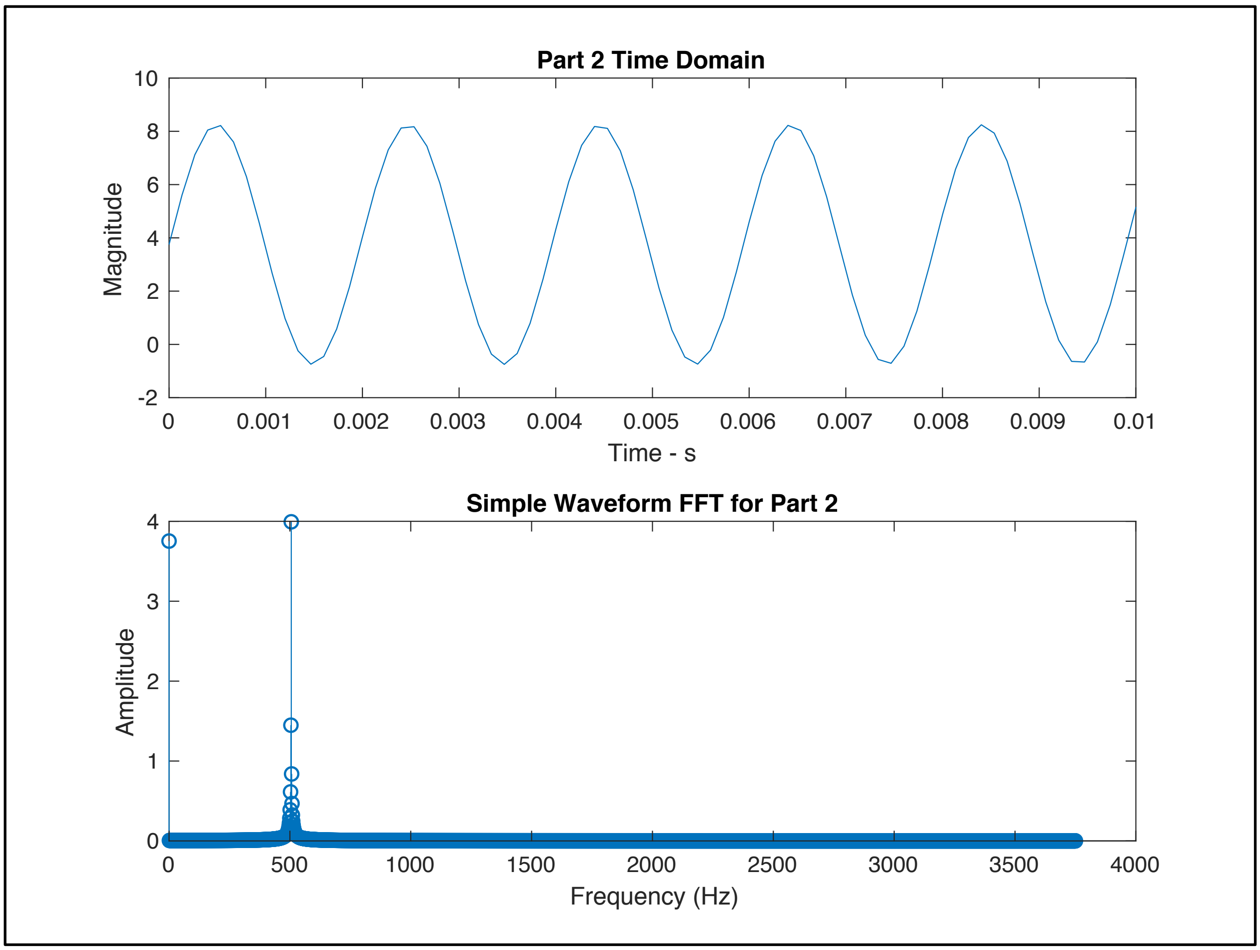


Part 1 Time Domain
Magnitude
5
0
-5
0
0.001
0.002
0.003
0.004
0.005
0.006
0.007
0.008
0.009
0.01
Time - s
Simple Waveform FFT for Part 1
Amplitude
4
3
2
1
0
0
500
1000
1500
2000
2500
3000
3500
4000
Frequency (Hz)

b) The frequency domain plot should be frequency on the x-axis and linear magnitude on the y-axis. Plot the entire frequency range from zero to the folding frequency.

MATLAB Example Code

```
% Part 4b
subplot(2,1,1);
plot(t,y2)
xlim([0 0.01])
xlabel('Time - s')
ylabel('Magnitude')
title('Part 2 Time Domain')

subplot(2,1,2);
stem(freq2,amp_spec2(1:(n2)));
xlabel('Frequency (Hz)'); ylabel('Amplitude');
title('Simple Waveform FFT for Part 2');
```



5. **Using the time domain data only, estimate the fundamental frequency for each signal. You will need to briefly explain the method(s) used to determine the frequency.** ***You are not to use the equation or frequency domain, only data obtained from discrete time domain data.***

The fundamental frequency can be found based on the period of the waveform. The fundamental frequency is simply the reciprocal of the fundamental period, or the smallest period of time over which the function can be completely described. From the time domain graph in part 1, one can observe that the Period = 0.002 s. Taking the reciprocal yields: $\mathbf{f_0}$ **= 500 Hz**.

For part 2, the observed period is slightly less than 0.002 s, or around 0.00198. Taking the reciprocal yields: $\mathbf{f_0}$ **= 505.05 Hz**.

6. **Using the frequency domain data only, estimate the fundamental frequency for each signal. You will need to briefly explain the method(s) used to determine the frequency.**

The fundamental frequency is represented by the first harmonic. In these values can be read off of the FFT Diagrams as the frequency with the largest amplitude. For part 1, the fundamental frequency is $\mathbf{f_0}$ **= 500 Hz.** For part 2, the fundamental frequency is $\mathbf{f_0}$ **= 505 Hz**.

7. **Using the time domain data only, estimate the null offset for each signal. You will need to briefly explain the method(s) used to determine the signals null offset.**

In the time domain, the offset is the average or mean of the data. Because the data is a sine wave, the offset is also the initial value of the magnitude vector. For part 1, the offset is **0.** For part 2, the offset is **3.75.**

8. **Using the frequency domain data only, estimate the null offset for each signal. You will need to briefly explain the method(s) used to determine the signals null offset.**

In the frequency domain, the offset would appear as an amplitude at 0 Hz frequency. In part 1, there is no significant amplitude at 0 Hz, meaning that the offset is **0.** In part 2, the amplitude value from the graph at 0 Hz is **3.75.** On the original FFT diagram, the offset amplitude was twice the actual offset, but this was corrected for in the MATLAB coding.

9. **Using the results from above, compare and contrast the information you can obtain between the time and frequency domains.**

Depending on what is required, it can be easier to obtain information by choosing whether to use the time or the frequency domain. In the case of fundamental frequency, the frequency domain was much easier to use. In the frequency domain, one had to simply look at the graph to find the frequency with the largest amplitude. However, finding the fundamental frequency in the time domain required estimating the period from the graph and calculating the frequency based off of this number. The frequency allowed for a simpler method and a more accurate answer. However, in the case of offset, the time domain would be slightly easier. Finding the offset in the frequency domain required correction within the MATLAB coding, which could easily be overlooked.

Beat

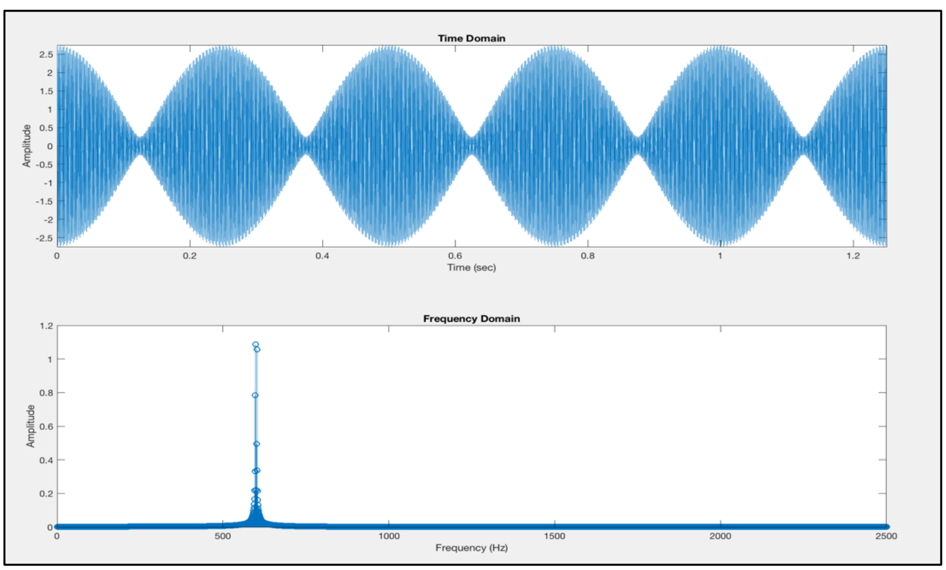
1. **Using MATLAB or Excel, create the following complex waveform over five seconds with a time step of at least 5 kHz. You should include a copy of the code used.**

$$y(t) = 1.5\sin(2\pi(598)t) + 1.25\sin(2\pi(602)t)$$

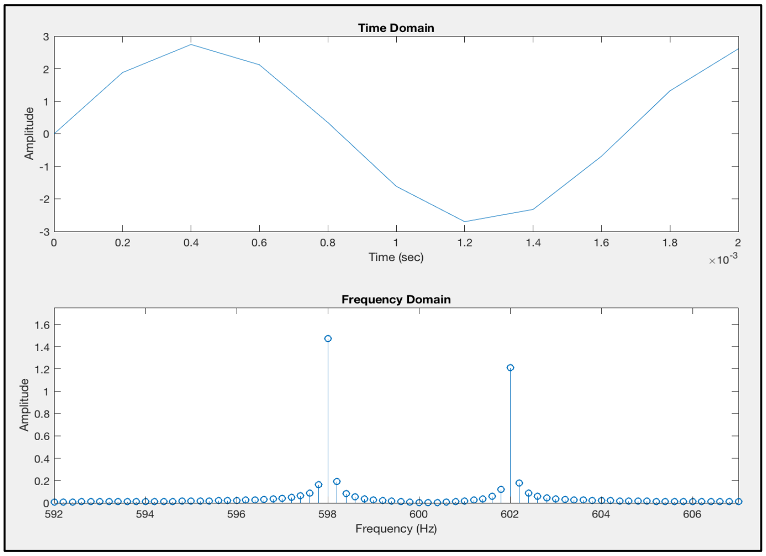
MATLAB Example Code

```
% Givens:
Ts = 5; %Total Sample time - sec
fs = 5000; %Sample Rate - Hz
A1 = 1.5;   A2 = 1.25; %Amplitudes of the signal
f1 = 598;   f2 = 602; %Frequencies of the signals

%Solving:
DeltaTime = 1/fs; %sec
FreqResoultion = 1/Ts; % Frequency Resolution
FoldingFreq = fs/2; %Folding Frequency - Hz
fbeat = f1-f2; % Beat Frequency - Hz
Tbeat = 1/fbeat; % Beat Period
%===========================TIME DOMAIN========================
subplot(2,1,1)
t = [0:DeltaTime:Ts]; % Time range with time step of 0.0002sec
y = A1*sin(2*pi*f1*t) + A2*sin(2*pi*f2*t); % Signal Equation
plot(t, y);
xlabel('Time (sec)');
ylabel('Amplitude');
axis([0, -5*Tbeat, -3, 3]);
title('Time Domain');
%===========================FREQUENCY DOMAIN===================
subplot(2,1,2)
Y = fft(y); %compute Fourier Transform
n = length(y)/2; % Number of data points
n = ceil(n);    % Round to the nearest interger
freq = linspace(0,FoldingFreq, n);  % frequency range
amp_spec = abs(Y)/n;    % Absolute value and normalize
stem(freq, amp_spec(1:n));
xlabel('Frequency (Hz)');
ylabel('Amplitude');
title('Frequency Domain');
```

2. **Create a subplot that contains the time domain in one plot and the frequency domain in a third plot, for the beat.**

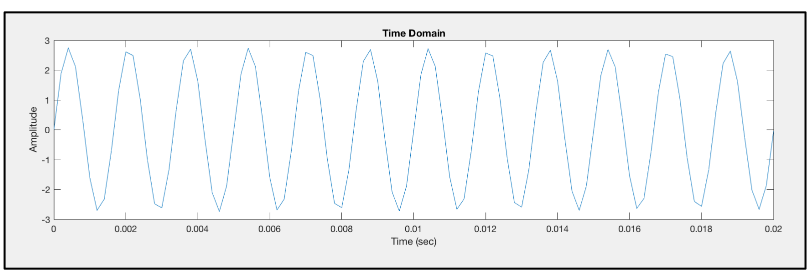


3. **Using the data obtained, determine the major frequency components of the signal. You will need to briefly explain the method(s) used to determine the signals null offset.**



The Null Offset for this complex waveform is 0. We can see this when we reduce the range of our time axis in the time domain. At 0 seconds we have an amplitude of 0, also you can see the mean of the time domain data is 0, or the Null Offset. To identify the major frequencies, I reduced the range of the frequency axis to capture where I believed the major frequencies to occur. Here we can clearly see the major frequencies to occur at <u>598 Hz</u> and <u>602 Hz</u>.

4. **Using the information shown below (also shown in Lecture Set 6 page 4-10), determine the two frequencies the signal is composed of using on data from the discrete time domain.**



$$y = A\sin(2\pi f_1 t) + A\sin(2\pi f_2 t) \qquad = 2\cos\left(2\pi \frac{f_1 - f_2}{2} t\right)\sin(2\pi \frac{f_1 + f_2}{2} t)$$

Within the time domain of range 0 to 0.02 seconds, and counted that 12 cycles occurred within this range. So to solve for the pitch frequency we divide the cyles by the time $\frac{12}{0.02} = 600Hz$. This can also be proved by the pitch frequency equation of $\frac{f1+f2}{2}$. Then from the equation above we can multiply the pitch frequency by two to get 1200 Hz. To get the beat frequency we do $fbeat = f1 - f2 = 4Hz$. So using the fbeat equation and using $2 * f1 = 1200 - 4$ we get our first frequency to be 598 Hz and the second frequency to be 602 Hz. And from the frequency domain graph, from question 3, we see that at 598 Hz and 602 Hz our amplitude is 1.5 and 1.25 respectively, thus proving $y(t) = 1.5\sin(2\Pi(598)t) + 1.25\sin(2\Pi(602)t)$ to be true.

Mixed

1. **Using MATLAB or Excel, create the following complex waveform over two seconds with a time step of at least 20 kHz. You should include a copy of the code used.**

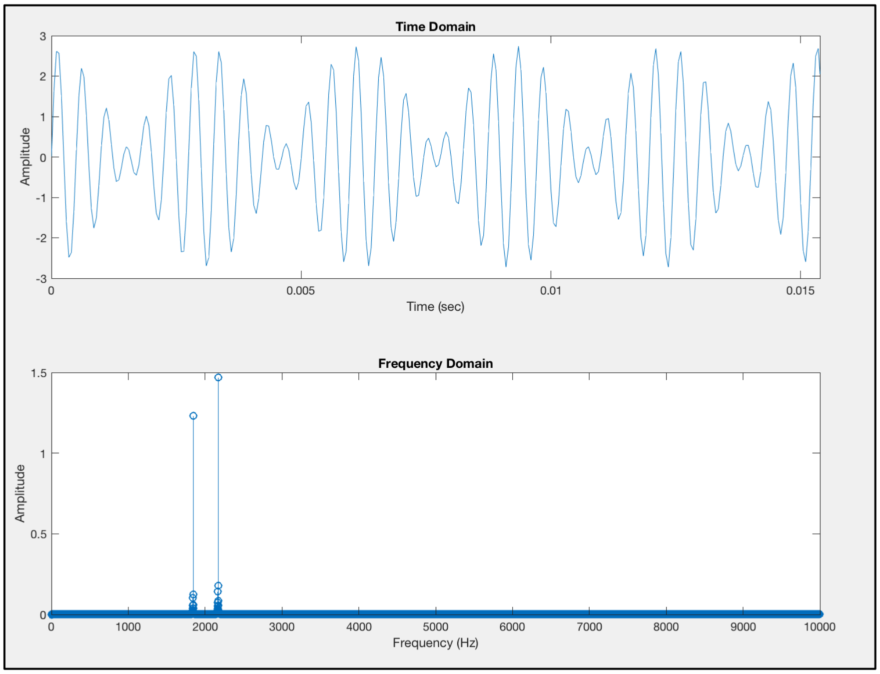
$$y(t) = 1.5\sin(2\pi(2168)t) + 1.25\sin(2\pi(1843)t)$$

2. **Using MatLab, convert the signal above into the frequency domain using the FFT.**

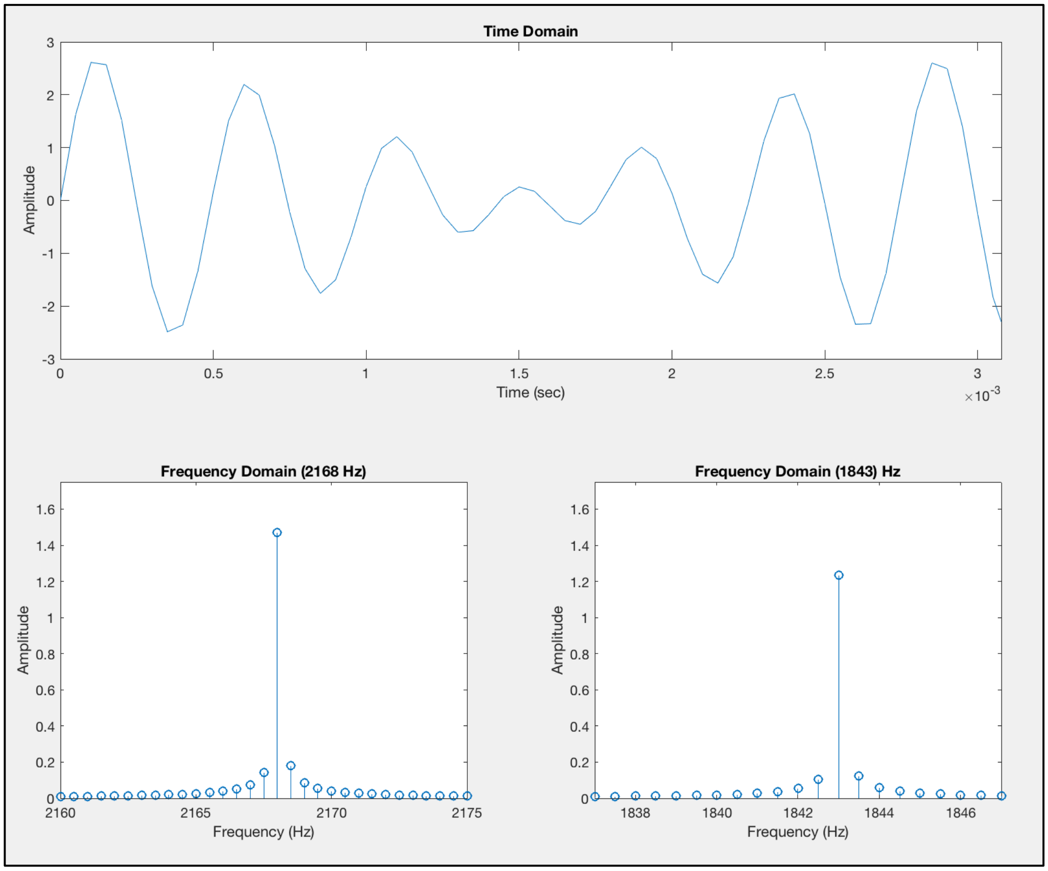
MATLAB Example Code

```
% Givens:
Ts = 2; %Total Sample time - sec
fs = 20000; %Sample Rate - Hz
A1 = 1.5;   A2 = 1.25; %Amplitudes of the signal
f1 = 2168;   f2 = 1843; %Frequencies of the signals

%Solving:
DeltaTime = 1/fs; %sec
FreqResoultion = 1/Ts; % Frequency Resolution
FoldingFreq = fs/2; %Folding Frequency - Hz
fbeat = f1-f2; % Beat Frequency - Hz
Tbeat = 1/fbeat; % Beat Period
%===========================TIME DOMAIN=====================
subplot(2,1,1)
t = [0:DeltaTime:Ts]; % Time range
y = A1*sin(2*pi*f1*t) + A2*sin(2*pi*f2*t); % Signal Equation
plot(t, y);
xlabel('Time (sec)');
ylabel('Amplitude');
axis([0, 5*Tbeat, -3, 3]);
title('Time Domain');
%============================FREQUENCY DOMAIN================
subplot(2,1,2)
Y = fft(y); %compute Fourier Transform
n = length(y)/2; % Number of data points
n = ceil(n);    % Round to the nearest interger
freq = linspace(0,FoldingFreq, n);  % frequency range
amp_spec = abs(Y)/n;    % Absolute value and normalize
stem(freq, amp_spec(1:n));
xlabel('Frequency (Hz)');
ylabel('Amplitude');
title('Frequency Domain');
```

3. Create a subplot that contains the time domain in one plot and the frequency domain in a third plot, for the beat.



4. Using the data obtained, determine the major frequency components of the signal. You will need to briefly explain the method(s) used to determine the signals null offset.



The Null Offset for this complex waveform is 0. We can see this when we reduce the range of our time axis in the time domain on the top graph. At 0 seconds we have an amplitude of 0, also you can see the mean of the time domain data is 0, or the Null Offset. To identify the major frequencies, I reduced the range of the frequency axis to capture where I believed the major frequencies to occur. Here we can clearly see the major frequencies to occur at <u>2168 Hz</u> on the right plot and <u>1843 Hz</u> on the left plot.

5. **Using MATLAB or Excel, create the following complex waveform over two seconds with a time step of at least 20 kHz. You should include a copy of the code used.**

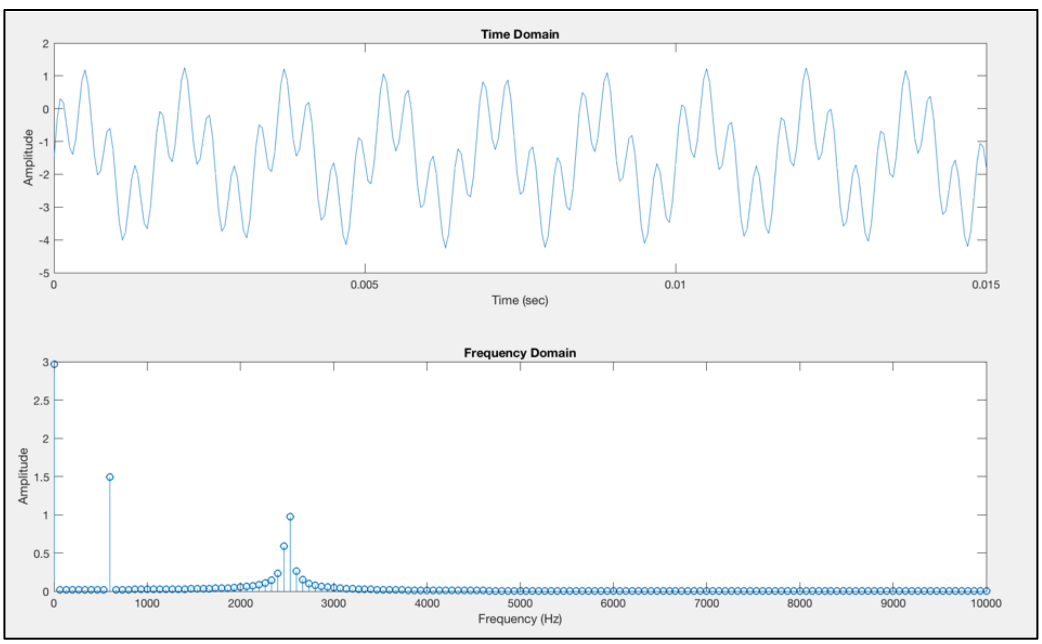
$$y(t) = 1.5\sin(2\pi(598)t) + 1.25\sin(2\pi(2500)t) - 1.5$$

6. **Using MatLab, convert the signal above into the frequency domain using the FFT.**

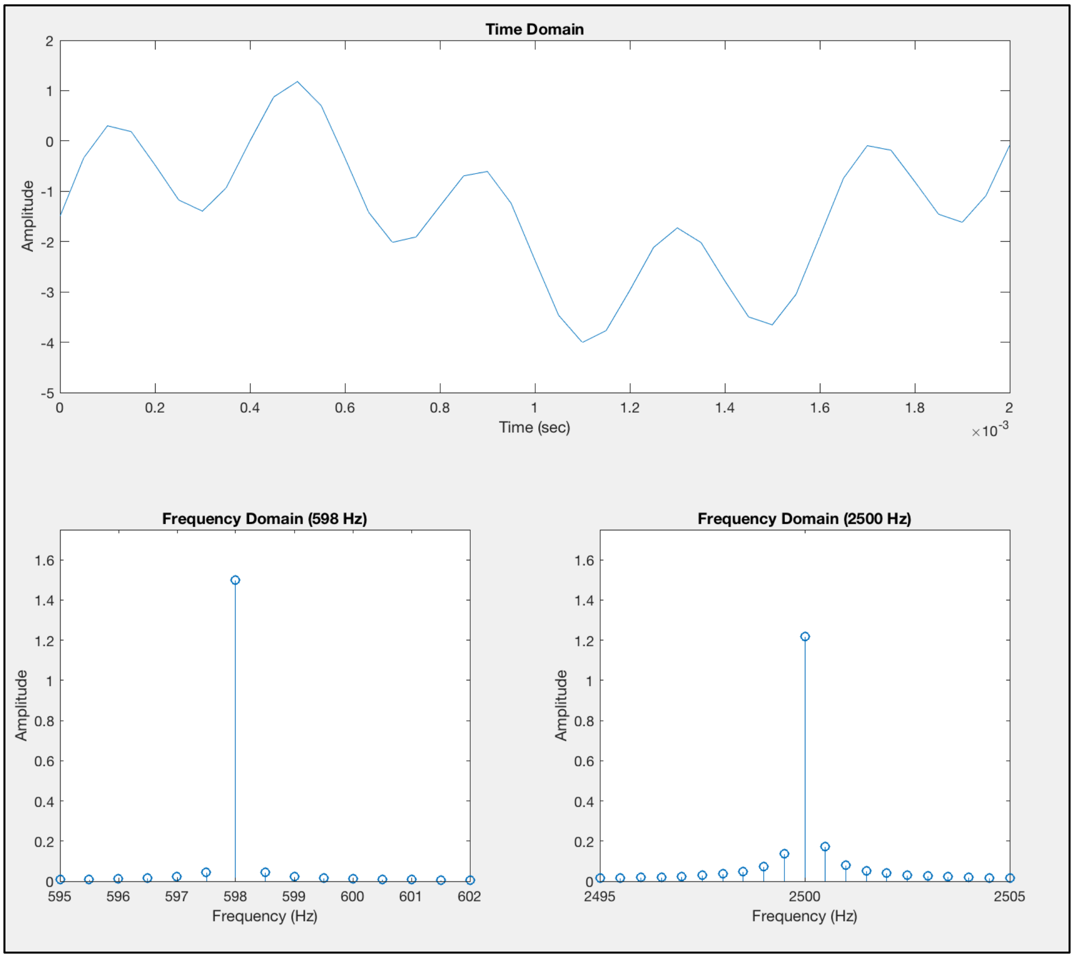
MATLAB Example Code

```
% Givens:
Ts = 2; %Total Sample time - sec
fs = 20000; %Sample Rate - Hz
A1 = 1.5;   A2 = 1.25; %Amplitudes of the signal
f1 = 598;   f2 = 2500; %Frequencies of the signals

%Solving:
DeltaTime = 1/fs; %sec
FreqResoultion = 1/Ts; % Frequency Resolution
FoldingFreq = fs/2; %Folding Frequency - Hz
%===========================TIME DOMAIN=============================
subplot(2,1,1)
t = [0:DeltaTime:Ts]; % Time range
y = A1*sin(2*pi*f1*t) + A2*sin(2*pi*f2*t) - 1.5; % Signal Equation
plot(t, y);
xlabel('Time (sec)');
ylabel('Amplitude');
axis([0, 0.015, -5, 2]);
title('Time Domain');
%===========================FREQUENCY DOMAIN========================
subplot(2,1,2)
Y = fft(y); %compute Fourier Transform
n = length(y)/2; % Number of data points
n = ceil(n);    % Round to the nearest interger
freq = linspace(0,FoldingFreq, n);  % frequency range
amp_spec = abs(Y)/n;    % Absolute value and normalize
stem(freq, amp_spec(1:n));
xlabel('Frequency (Hz)');
ylabel('Amplitude');
title('Frequency Domain');
```

7. **Create a subplot that contains the time domain in one plot and the frequency domain in a third plot, for the beat.**



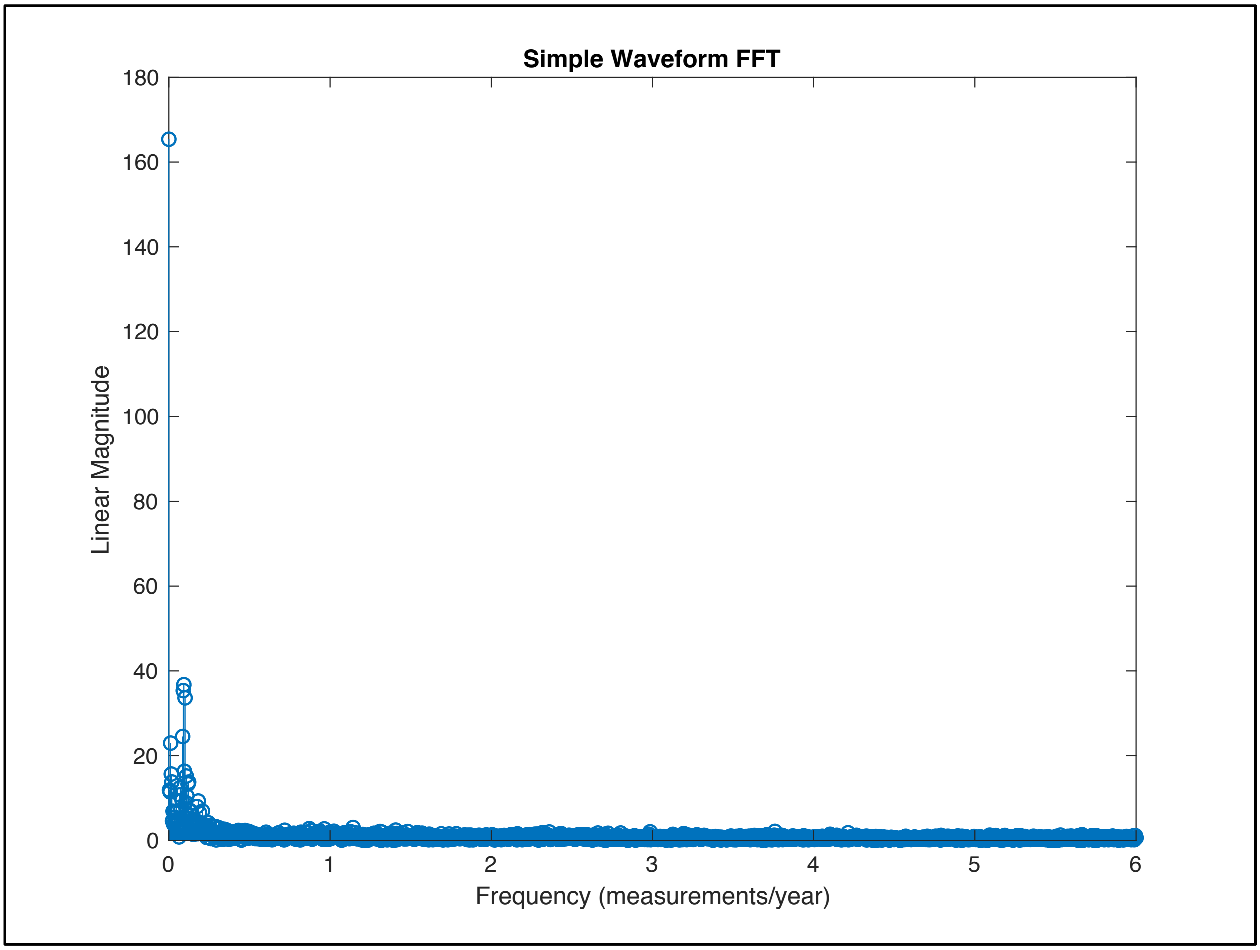
8. Using the data obtained, determine the major frequency components of the signal. You will need to briefly explain the method(s) used to determine the signals null offset.



The Null Offset for this complex waveform is about -1.5. We can see this when we reduce the range of our time axis in the time domain on the top graph. At 0 seconds we have an amplitude of -1.5, or the Null Offset. To identify the major frequencies, I reduced the range of the frequency axis to capture where I believed the major frequencies to occur. Here we can clearly see the major frequencies to occur at 598 Hz on the right plot and 2500 Hz on the left plot.

Sun Spot Activity

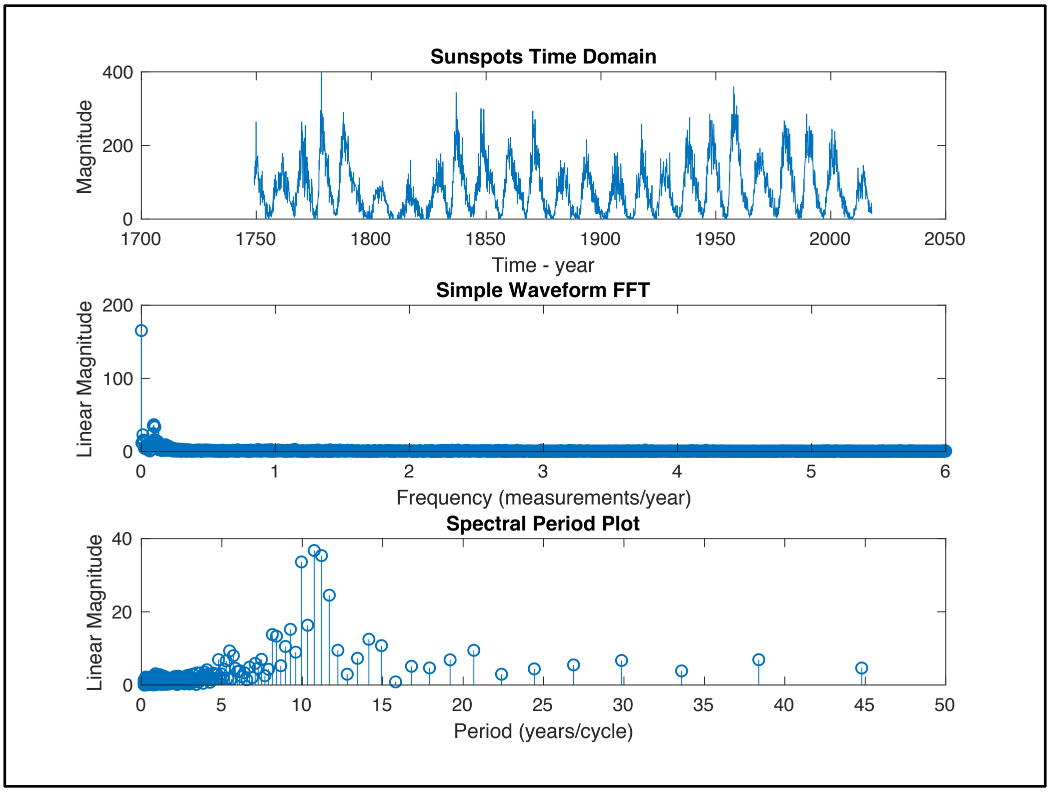
1. **Using MATLab, convert the signal above into the frequency domain using the FFT. You should include a copy of the code used.** ***In this case your frequency is 12 measurements per year, from 1745 to 2017.***



MATLAB Example Code

```
%Sun Spot - Problem 1
fs=12;

load sunspot.csv %this will load the file from the current project folder
year = sunspot(:,1); %parse out the decimal year from the data
sactivity = sunspot(:,2); %parse out the sunspot activity from the data

Y = fft(sactivity);
n = length(sactivity)/2; n = ceil(n);
amp_spec = abs(Y)/n;
freq = linspace(0,fs/2,n);
period = 1./freq;
stem(freq,amp_spec(1:(n)));
xlabel('Frequency (measurements/year)');
ylabel('Linear Magnitude');
title('Simple Waveform FFT');
pause
```

2. **Create a subplot that contains the time domain in one plot, frequency domain in a second plot, and spectral period in the third, for the signal.**
    a) **In the time domain the graph should be time (year) on the x-axis and magnitude on the y- axis.**
    b) **The frequency domain plot should be frequency (12 measurements/year) on the x-axis and linear magnitude on the y-axis. Plot the entire frequency range from zero to the folding frequency.**
    c) **The spectral period plot should be period (years/cycle) on the x-axis and linear magnitude on the y-axis. Plot from zero to 50 years/cycle.**



MATLAB Example Code

```
%Sun Spot - Problem 1
fs=12;

load sunspot.csv %this will load the file from the current project folder
year = sunspot(:,1); %parse out the decimal year from the data
sactivity = sunspot(:,2); %parse out the sunspot activity from the data

Y = fft(sactivity);
n = length(sactivity)/2; n = ceil(n);
amp_spec = abs(Y)/n;
freq = linspace(0,fs/2,n);
period = 1./freq;
stem(freq,amp_spec(1:(n)));
xlabel('Frequency (measurements/year)');
ylabel('Linear Magnitude');
title('Simple Waveform FFT');
pause

%Sun Spot - Problem 2
subplot(3,1,1)
plot(year,sactivity)
title('Sunspots Time Domain')
xlabel('Time - year')
ylabel('Magnitude')

subplot(3,1,2)
stem(freq,amp_spec(1:(n)));
xlabel('Frequency (measurements/year)');
ylabel('Linear Magnitude');
title('Simple Waveform FFT');
```

```
subplot(3,1,3)
stem(period,amp_spec(1:(n)))
xlim([0 50])
xlabel('Period (years/cycle)')
ylabel('Linear Magnitude')
title('Spectral Period Plot')
```

3. **Using the frequency spectrum obtained from above, identify and discuss any cycles present in the sunspot activity and determine the period(s). You can make use of both the frequency and time domain.**

The time domain plot and the Spectral Period plot are the most telling when it comes to identifying any cycles within the data. The time data shows a peak occurring approximately every ten years (there are 5 peaks over a 5-year period). This is fairly consistent across the range of the data. The Spectral Period plot confirms this estimate. A grouping of high-magnitude points right around the 10 years/cycle shows that a full period occurs about every 10 years.

4. **Discuss what information can be seen in the frequency spectrum that cannot be obtained from the plot of the time domain and what information can be gained from the time domain that cannot be readily obtained from the frequency spectrum?**

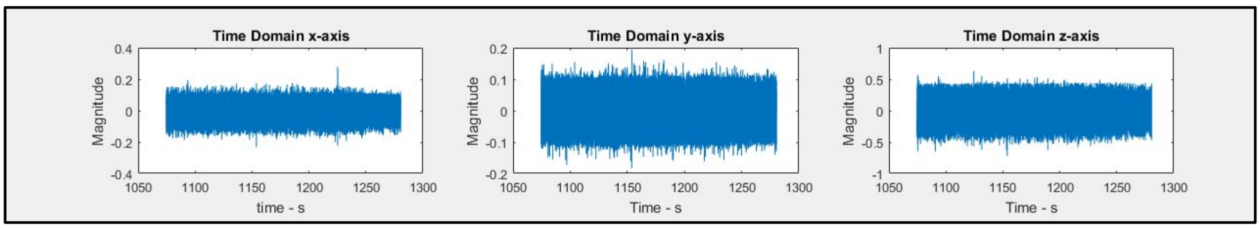
The time domain plot does a much better job at showing the actual magnitude of the measurements, allowing us to correlate with actual data values. This plot can also give us information on how the signal changes with time, which could show us any increasing or decreasing trends or allow us to find rate of change at specific areas. However, frequency domain allows us to separate parts of the data into separate categories based on the frequency. Information such as fundamental frequency and harmonic frequencies can also easily be found in the frequency domain. With some manipulation the frequency data can even show the signal's period (as seen above).

Aircraft Vibration

1. **Using the data create a subplot for that has three rows and three columns for the entire data set.**

   MATLAB Example Code

```
%%%%%%%%%%%%Given%%%%%%%%%%%
load Airplane_Vibration.mat; %Loads the data file
time = ADC(1,:)'; %parse out the time and put in the correct format
acc_x = ADC(2,:)'; %parse out the x axis data
acc_y = ADC(3,:)'; %parse out the y axis data
acc_z = ADC(4,:)'; %parse out the z axis data
fs_aircraft = 1/(time(2,1)-time(1,1)); %Determine sample rate

%%%%%%%%%%Solving%%%%%%%%%
Deltatime = 1/fs_aircraft; %Calculate delta_t
FreqResolution = 1/time; %Calculate Frequency Resolution
FoldingFreq = fs_aircraft/2; %Folding Frequency is 1/2 sample rate

%%%%%%%%%Time Domain%%%%%%%%%
%graph Time Domain x-axis
subplot(3,3,1); plot(time, acc_x); xlabel('time - s'); ylabel('Magnitude'); title('Time Domain x-axis');
%graph Time Domain y-axis
subplot(3,3,2); plot(time, acc_y); xlabel('Time - s'); ylabel('Magnitude'); title('Time Domain y-axis');
%graph Time Domain z-axis
subplot(3,3,3); plot(time, acc_z); xlabel('Time - s'); ylabel('Magnitude'); title('Time Domain z-axis');
```



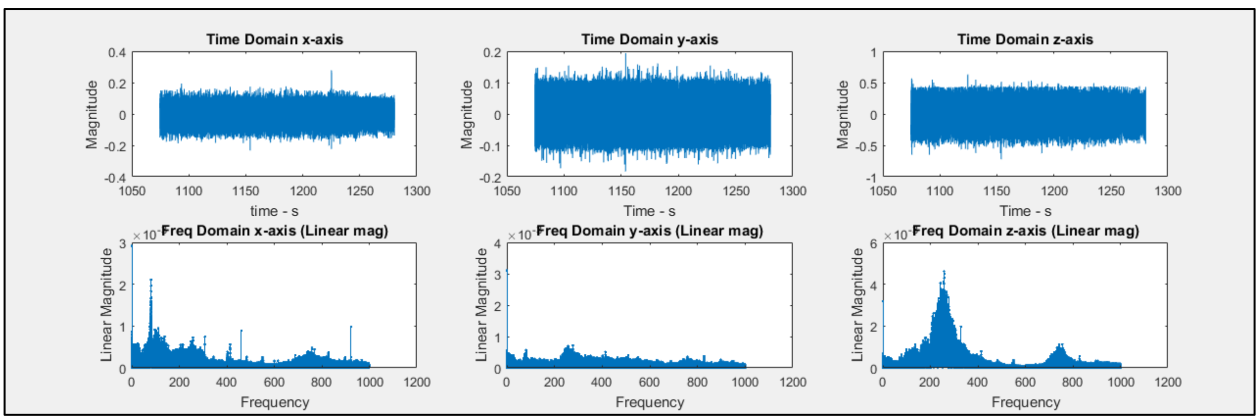
2. **Using MATLab, convert all three channels for entire signal into the frequency domain using the FFT. You should include a copy of the code used.**

MATLAB Example Code

```
%%%%%%%%%%FFT Frequency Domain%%%%%%%%
X1=fft(acc_x); %compute Fourier Transform for x-axis
Y1=fft(acc_y); %compute Fourier Transform for y-axis
Z1=fft(acc_z); %compute Fourier Transform for z-axis
n1 = length(acc_x)/2; n1=ceil(n1); %determine number of data points and cut in half
n2 = length(acc_y)/2; n2=ceil(n2); %determine number of data points and cut in half
n3 = length(acc_z)/2; n3=ceil(n3); %determine number of data points and cut in half
amp_spec1 = abs(X1)/n1; amp_spec2 = abs(Y1)/n2; amp_spec3 = abs(Z1)/n3; %absolute value and normalize
freq1 = linspace(0, FoldingFreq, n1); %set frequency for x-axis
freq2 = linspace(0, FoldingFreq, n2); %set frequency for y-axis
freq3 = linspace(0, FoldingFreq, n3); %set frequency for z-axis
subplot(3,3,4); stem(freq1, amp_spec1(1:n1)); xlabel('Frequency'); ylabel('Linear Magnitude'); title('Freq Domain x-axis (Linear mag)');
subplot(3,3,5); stem(freq2, amp_spec2(1:n2)); xlabel('Frequency'); ylabel('Linear Magnitude'); title('Freq Domain y-axis (Linear mag)');
subplot(3,3,6); stem(freq3, amp_spec3(1:n3)) ; xlabel('Frequency'); ylabel('Linear Magnitude'); title('Freq Domain z-axis (Linear mag)');
```

3. **Using the results plot the frequency domain for each axis. The plot should be frequency on the x-axis and linear magnitude on the y-axis. Plot the entire frequency range from zero to the folding frequency. (Recommend *stem*)**

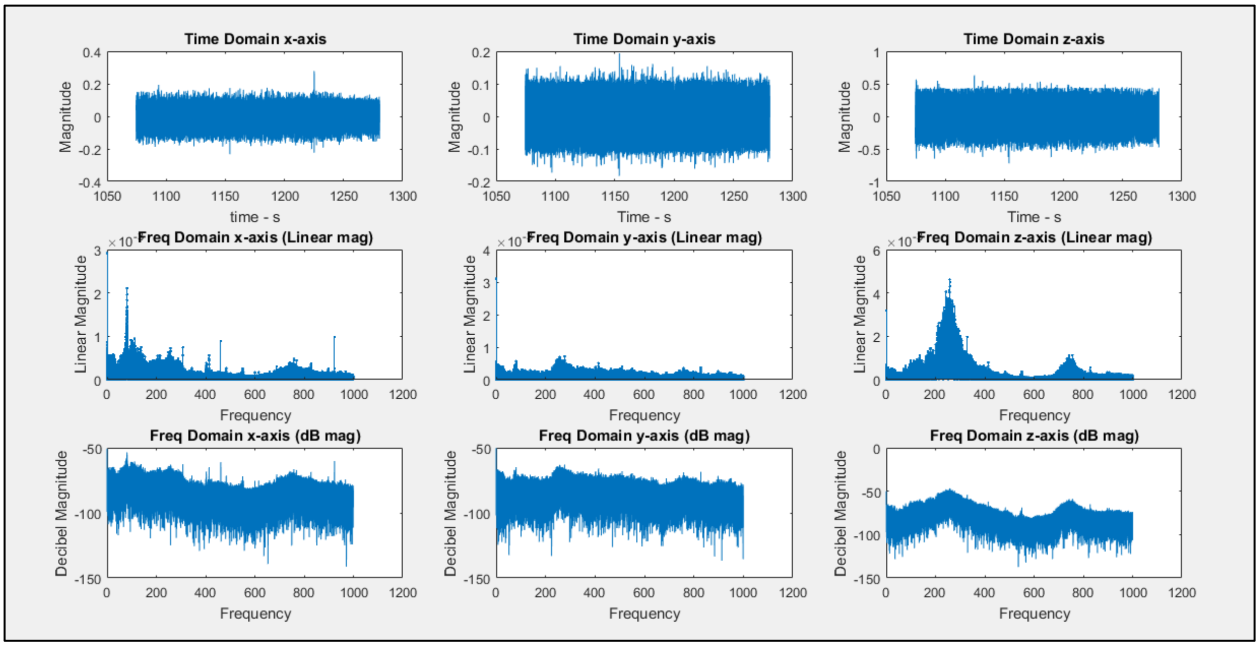
NOTE: Please find the problem 3 code in problem 2.



4. Convert the FFT linear magnitude to decibel magnitude and then plot the frequency domain for each axis. Plot should be frequency on the x-axis and decibel magnitude on the y-axis. Plot the entire frequency range from zero to the folding frequency. (Recommend *plot*)

MATLAB Example Code

```
%%%%%%%%%%%%Decibel Magnitude%%%%%%%%%%%%%%%
db_spec1 = mag2db(amp_spec1); %convert x-axis to dB and graph
subplot(3,3,7); plot(freq1, db_spec1(1:n1),'o');xlabel('Frequency'); ylabel('Decibel↙
Magnitude'); title('Freq Domain x-axis (dB mag)');
db_spec2 = mag2db(amp_spec2); %convert y-axis to dB and graph
subplot(3,3,8); plot(freq2, db_spec2(1:n2),'o');xlabel('Frequency'); ylabel('Decibel↙
Magnitude'); title('Freq Domain y-axis (dB mag)');
db_spec3 = mag2db(amp_spec3); %convert z-axis to dB and graph
subplot(3,3,9); plot(freq3, db_spec3(1:n3),'o'); xlabel('Frequency'); ylabel('Decibel↙
Magnitude'); title('Freq Domain z-axis (dB mag)');
```



5. Your client is interested in the vibrational profile that is produced, as they are designing a sensor package and need to understand the vibrations it will encounter. Using both the time and frequency domains for all three channels, create a report on what acceleration profile can be expected.

Seeing that the z-axis has the highest time and frequency domains, it can be expected that the acceleration profile will act sililar to that axis. The frequency domain graph shows us that there is a large peak in magnitude shortly after 200 Hz, and then another peak magnitude at 750 Hz. The time domain graph shows oscillating magnitudes of values between ± 0.5.

6. Using the data above, determine which channel has the largest vibrations. You will use this axis/channel for the remaining questions. Briefly discussion your selection.

The z-axis channel will have the largest vibrations. This was determined by looking at the magnitudes of the graphs. In Time Domain, Linear Frequency Domain and dB Frequency Domain, the z axis had the highest magnitudes. Next highest would be the x-axis, then the y-axis.

Short-Time Fourier Transform (STFT)

1. **Using Matlab or Excel, create the following complex waveform over three seconds with a time step of at least 10 kHz. You should include a copy of the code used.**

$$y(t) = \begin{Bmatrix} \sin(2\pi \cdot 440 \cdot t) & 0 \le t < 1 \\ \sin(2\pi \cdot 784 \cdot t) & 1 \le t < 2 \\ \sin(2\pi \cdot 587 \cdot t) & 2 \le t < 3 \end{Bmatrix}$$

```
% Short-Time Fourier transform (STFT)
% Part 1

t = 0:0.0001:1;
y = sin(2*pi*440*t);

subplot(2,2,1);
plot(t,y); grid on;
xlabel('Time (s)');
ylabel('Amplitude');
title('y = sin(2*pi*440*t)');

t = 1:0.0001:2;
y = sin(2*pi*784*t);

subplot(2,2,2);
plot(t,y); grid on;
xlabel('Time (s)');
ylabel('Amplitude');
title('y = sin(2*pi*784*t)');

t = 2:0.0001:3;
y = sin(2*pi*587*t);

subplot(2,1,2);
plot(t,y); grid on;
xlabel('Time (s)');
ylabel('Amplitude');
title('y = sin(2*pi*587*t)');
```

NOTE: Please find the time domain graphs in Problem 3.

2. Using Matlab, perform the FFT from 0 to 1 seconds, 1 to 2 seconds, and finally 2 to 3 seconds.

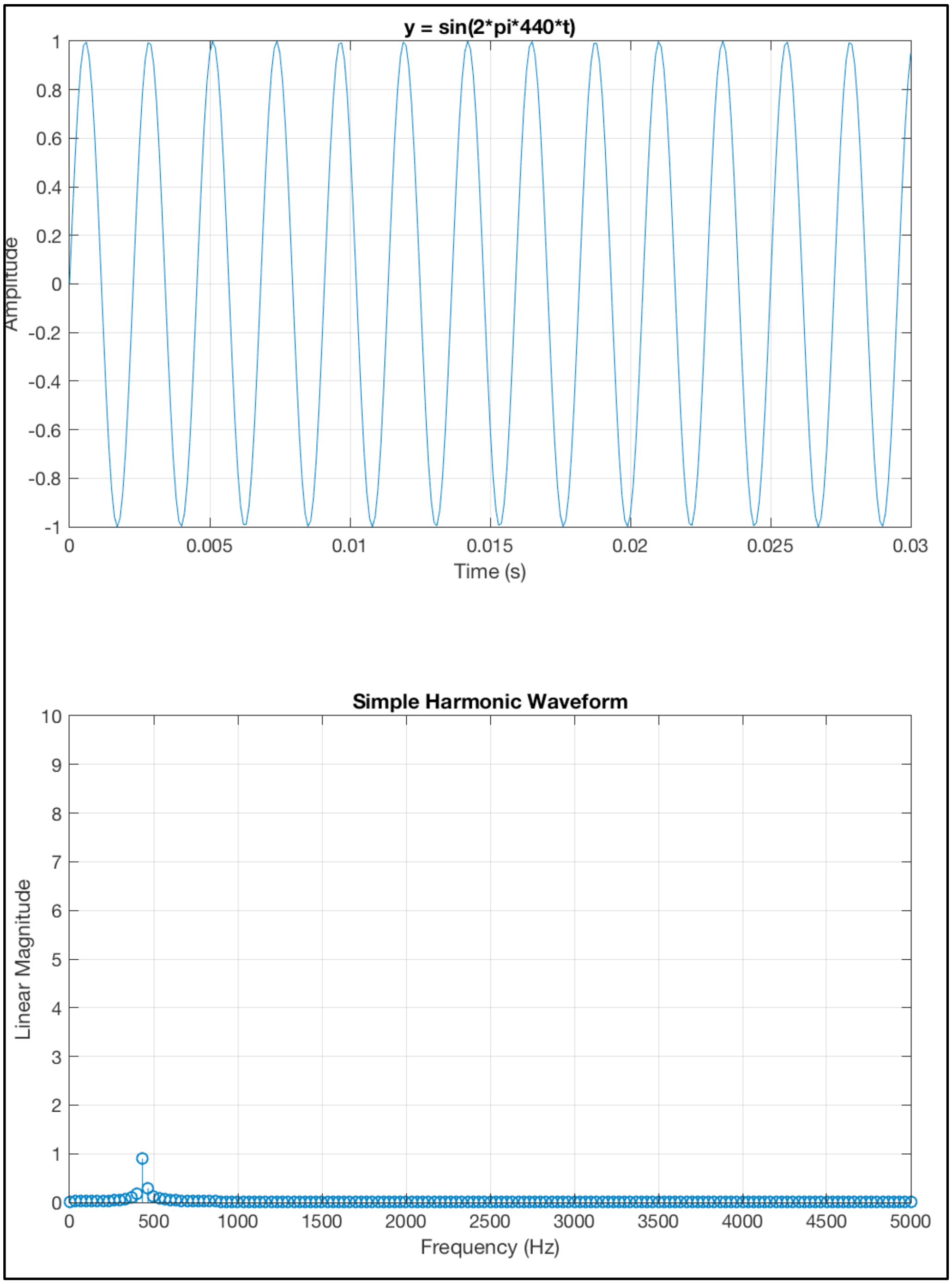
```
% Short-Time Fourier transform (STFT)
% Part 2 Equation 1; y = sin(2*pi*440*t)
figure(1)
subplot(2, 1, 1)
Fs = 10000;  % Sample Rate (Hz)
t = 0:0.0001:0.030;  % time

y = sin(2*pi*440*t);

plot(t,y)
grid on;
xlabel('Time (s)');
ylabel('Amplitude');
title('y = sin(2*pi*440*t)');

Y = fft(y); %compute fourier transform
n = length(y)/2; n = ceil(n); % n is the number of data points / 2
amp_spec = abs(Y)/n;    % amplitude is abs(fourier) / n
amp_spec(1) = amp_spec(1)/2;

freq = linspace(0,Fs/2,n);  % frequency range
subplot(2,1,2)
stem(freq,amp_spec(1:n)); grid on
axis([0 5000 0 10])
title('Simple Harmonic Waveform')
xlabel('Frequency (Hz)')
ylabel('Linear Magnitude')
```

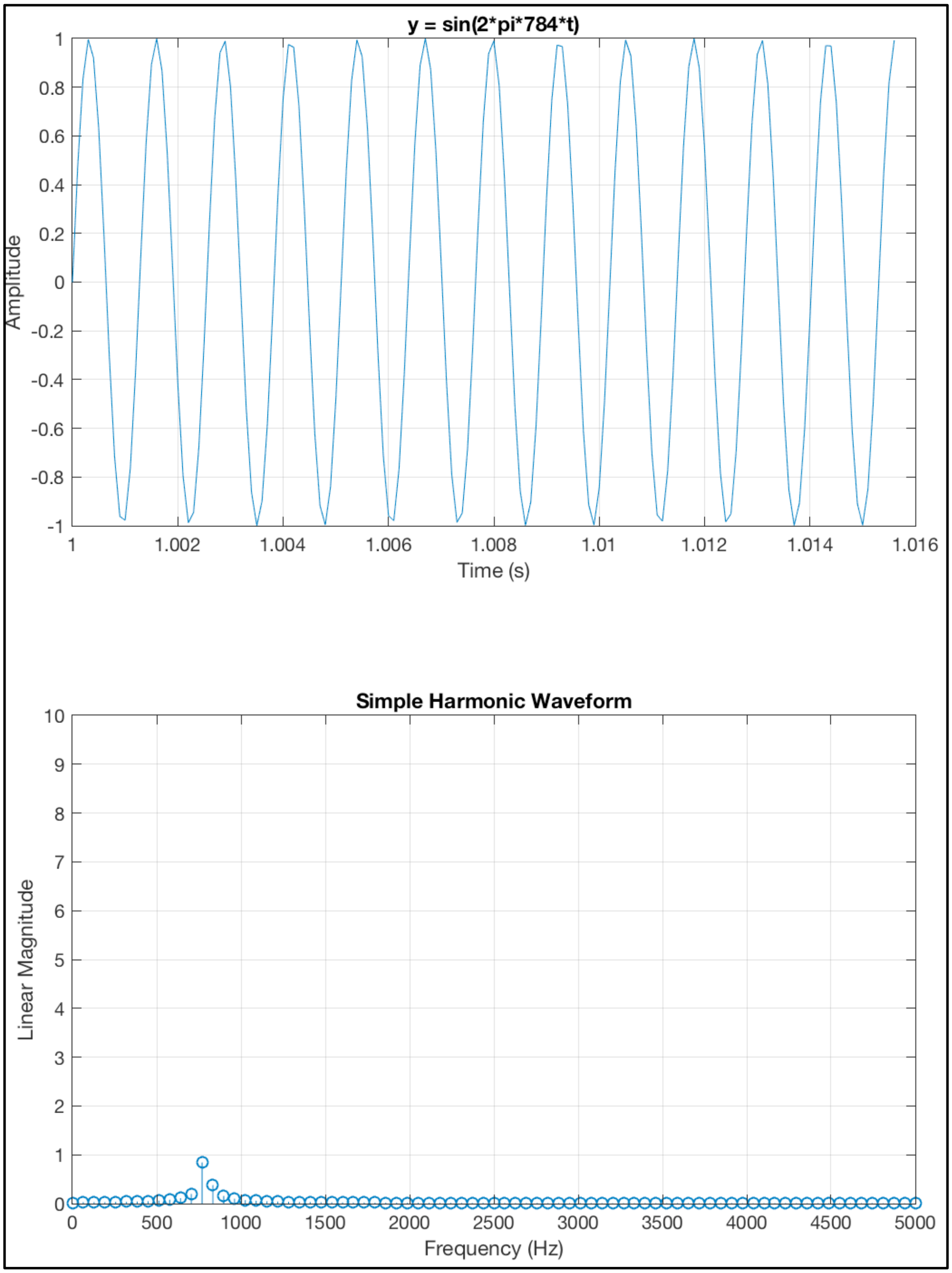
```
% Short-Time Fourier transform (STFT)
% Part 2; y = sin(2*pi*784*t)
figure(2)
subplot(2, 1, 1)
Fs = 10000;  % Sample Rate (Hz)
t = 1:0.0001:1.0156;  % time

y = sin(2*pi*784*t);

plot(t,y)
grid on;
xlabel('Time (s)');
ylabel('Amplitude');
title('y = sin(2*pi*784*t)');

Y = fft(y); %compute fourier transform
n = length(y)/2; n = ceil(n); % n is the number of data points / 2
amp_spec = abs(Y)/n;    % amplitude is abs(fourier) / n
amp_spec(1) = amp_spec(1)/2;

freq = linspace(0,Fs/2,n);  % frequency range
subplot(2,1,2)
stem(freq,amp_spec(1:n)); grid on
axis([0 5000 0 10])
title('Simple Harmonic Waveform')
xlabel('Frequency (Hz)')
ylabel('Linear Magnitude')
```

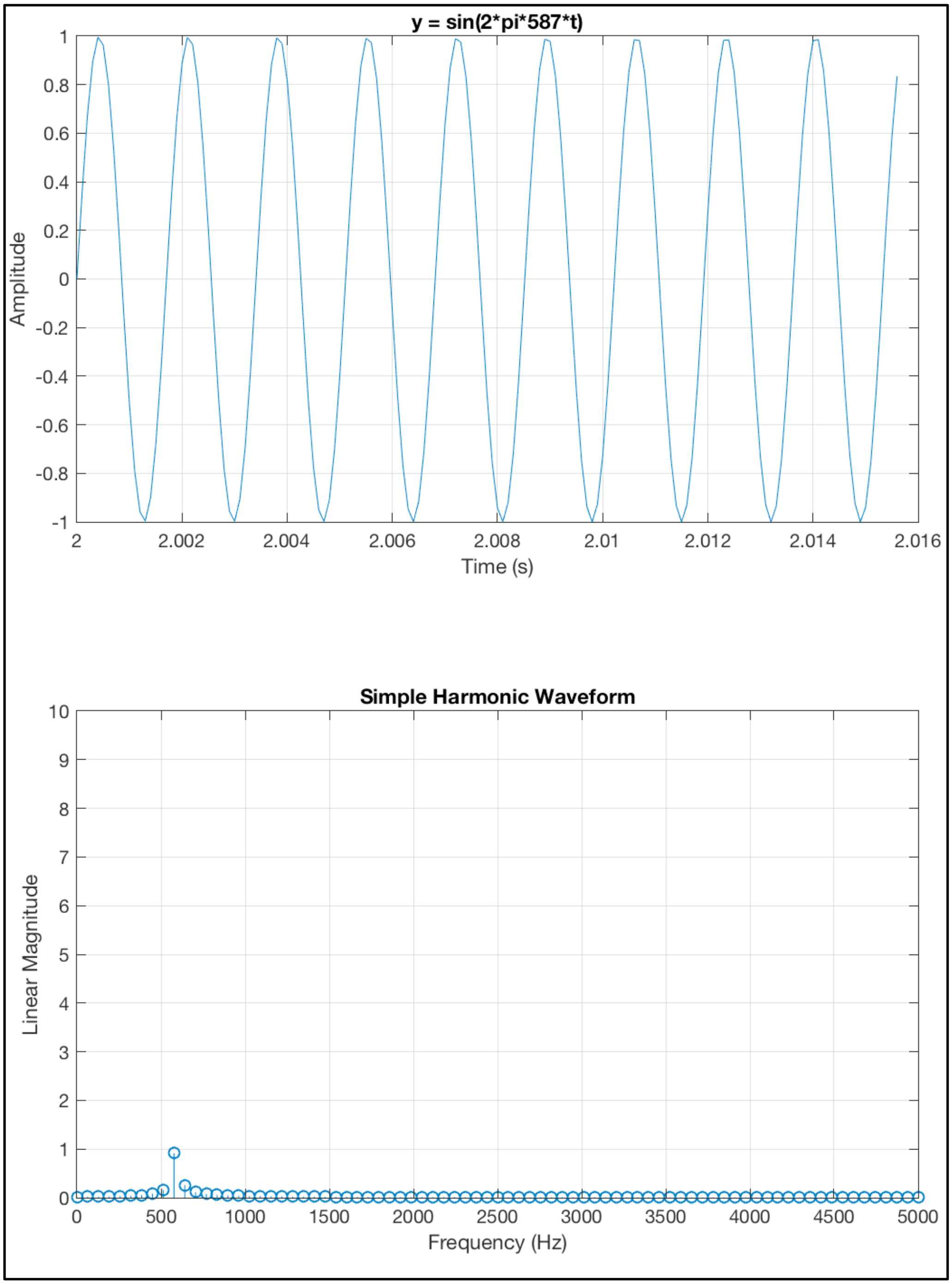
```
% Short-Time Fourier transform (STFT)
% Part 2 and 3; y = sin(2*pi*587*t)
figure(3)
subplot(2, 1, 1)
Fs = 10000;  % Sample Rate (Hz)
N = 10000;   % Number of data points
f = 587;    % frequency of signal
A = 4.5;    % amplitude of signal
t = 2:0.0001:2.0156;  % time

y = sin(2*pi*587*t);
plot(t,y)
grid on;
xlabel('Time (s)');
ylabel('Amplitude');
title('y = sin(2*pi*587*t)');

Y = fft(y); %compute fourier transform
n = length(y)/2; n = ceil(n); % n is the number of data points / 2
amp_spec = abs(Y)/n;    % amplitude is abs(fourier) / n
amp_spec(1) = amp_spec(1)/2;

freq = linspace(0,Fs/2,n);  % frequency range
subplot(2,1,2)
stem(freq,amp_spec(1:n)); grid on
axis([0 5000 0 10])
title('Simple Harmonic Waveform')
xlabel('Frequency (Hz)')
ylabel('Linear Magnitude')
```

NOTE: Please find the frequency domain graphs in Problem 3.

3. **Create a subplot for each of the three time ranges(0 to 1, 1 to 2, 2, to 3) that contains the time domain in one plot and the frequency domain in second plot.**
    a. **In the time domain the graph should be time on the x-axis and magnitude on the y-axis. You should trim our data down to show only five to eight typical oscillations.**
    b. **The frequency domain plot should be frequency on the x-axis and linear magnitude on the y-axis. Plot the entire frequency range from zero to the folding frequency.**




y = sin(2*pi*784*t)
Amplitude
Time (s)
Simple Harmonic Waveform
Linear Magnitude
Frequency (Hz)

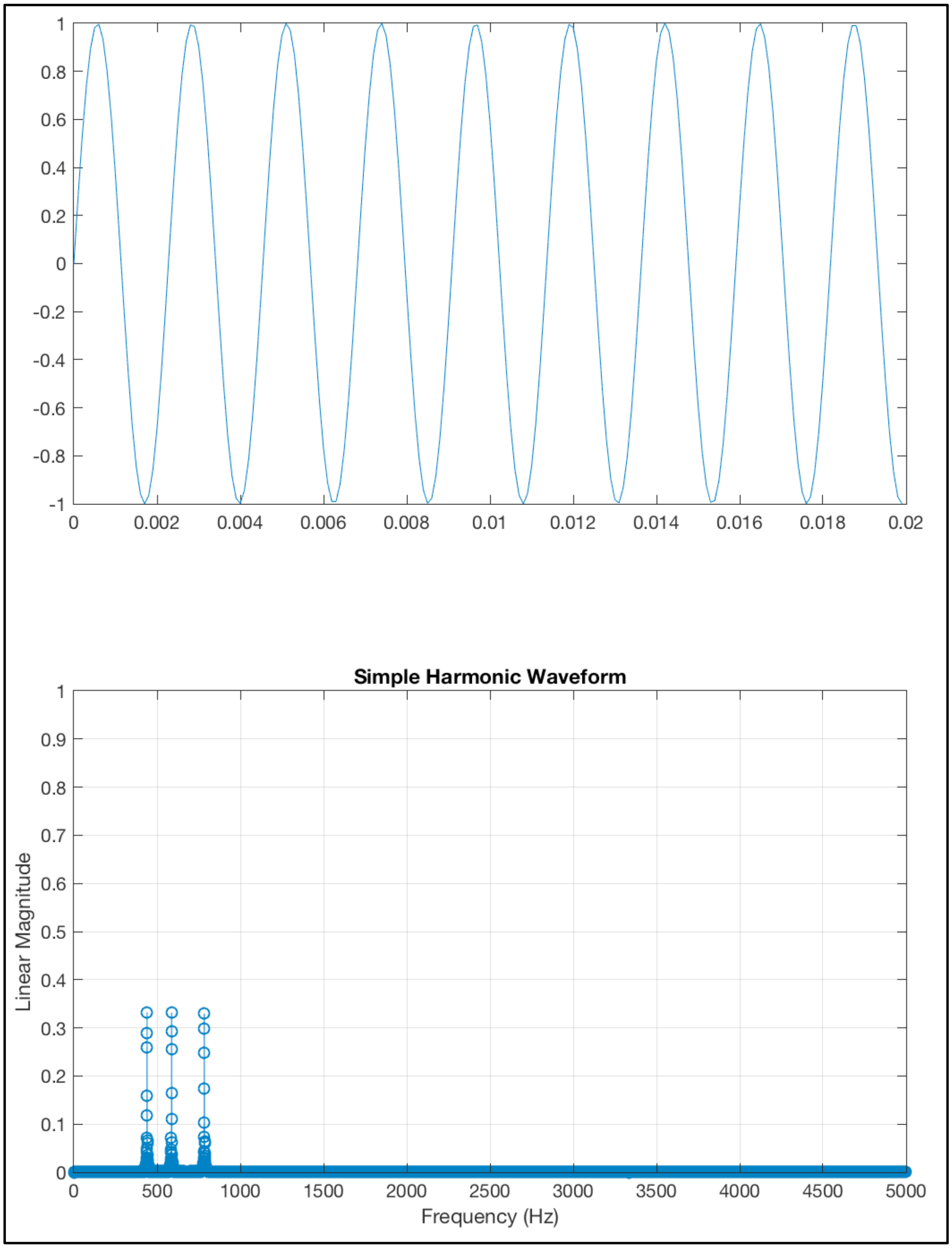

y = sin(2*pi*587*t)
Amplitude
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
2
2.002
2.004
2.006
2.008
2.01
2.012
2.014
2.016
Time (s)
Simple Harmonic Waveform
Linear Magnitude
10
9
8
7
6
5
4
3
2
1
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
Frequency (Hz)

4. Using Matlab, convert the entire signal above into the frequency domain using the FFT. You should include a copy of the code used.

```
% Short Time Fourier Transform
% Problem 4 Code

t1 = 0:0.0001:1; y1 = sin(2*pi*440*t1);
t2 = 1:0.0001:2; y2 = sin(2*pi*784*t2);
t3 = 2:0.0001:3; y3 = sin(2*pi*587*t3);
t11 = t1(1,:)'; t22 = t2(1,:)'; t33 = t3(1,:)';
y11 = y1(1,:)'; y22 = y2(1,:)'; y33 = y3(1,:)';

subplot(2, 1, 1)
grid on;
xlabel('Time (s)');
ylabel('Amplitude');
title('y = sin(2*pi*440*t)');

ab = t11((1:200),1); abb = y11((1:200), 1);
plot(ab,abb)

Y = fft(b); %compute fourier transform
n = length(b)/2; n = ceil(n); % n is the number of data points / 2
amp_spec = abs(Y)/n;    % amplitude is abs(fourier) / n
amp_spec(1) = amp_spec(1)/2;

freq = linspace(0,Fs/2,n);  % frequency range
subplot(2,1,2)
stem(freq,amp_spec(1:n)); grid on
axis([0 5000 0 1])
title('Simple Harmonic Waveform')
xlabel('Frequency (Hz)')
ylabel('Linear Magnitude')
```

NOTE: Please find the frequency domain graph in Problem 5.

5. **Create a subplot that contains the time domain in one plot and the frequency domain in the second plot.**
   a. **In the time domain the graph should be time on the x-axis and magnitude on the y-axis.**
   b. **The frequency domain plot should be frequency on the x-axis and linear magnitude on the y-axis. Plot the entire frequency range from zero to the folding frequency.**



6. Compare and contrast the information you can obtain from part 3 and part 5.

The frequency components of the combine Fourier combines all three frequencies, where the individuals solely plotted the frequency of its own equation.

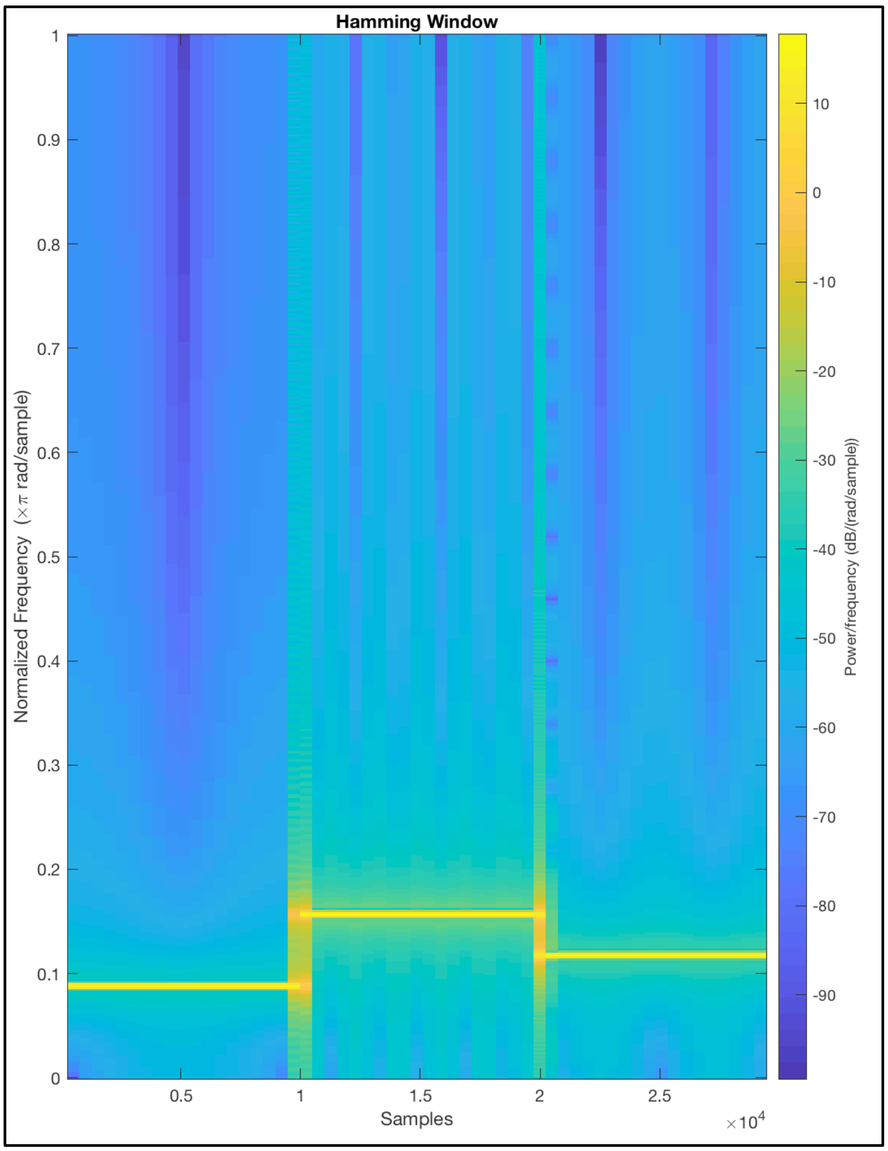
7. Perform a Short Time Fourier Transform on the signal, using the Matlab command *spectrogram.* With the following conditions:

a. Hamming window

b. Window size 0.25 seconds

c. 50 % overlap

d. No oversampling



8. Compare and contrast the information you can obtain from the basic FFT Results (parts 3 and 5) with that from the STFR (part 7).

The frequency signals for each signal show for the entirety of their sample range, so equation one pertains to 0 to 1 samples while equation two pertains to 1 to 2 samples, and equation 3 pertains to 2 to 3 samples.

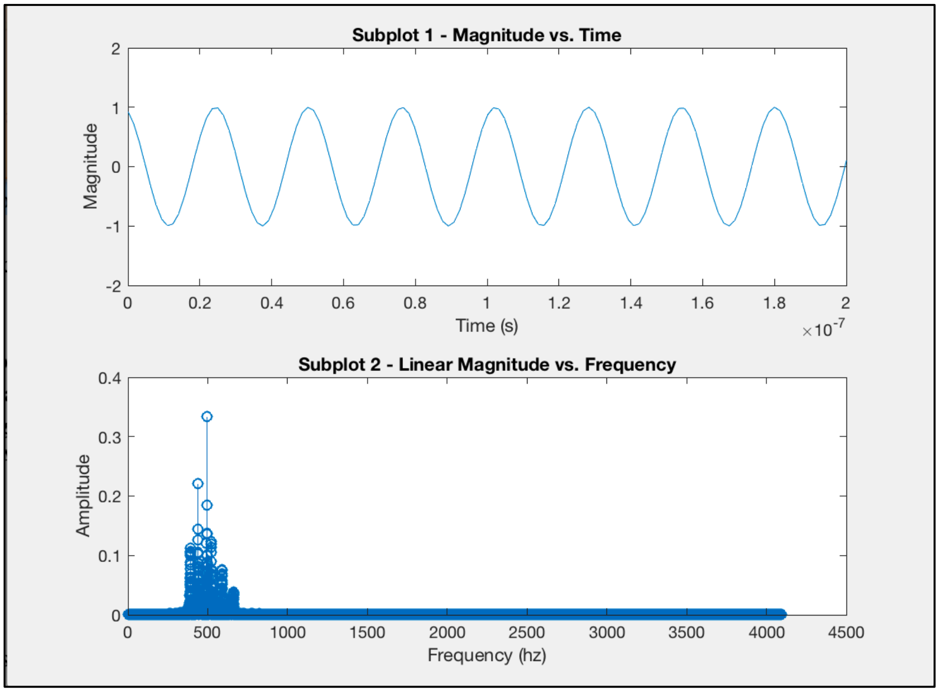
Victors Tone

1. **Using MATLab, convert the entire signal above into the frequency domain using the FFT. Include a copy of the code used.**

```
% Victor's Tone Section of Lab 6
% Colton Tubbs

Fs = 8192; % sample rate in hz
Y1 = fft(x); % compute fourier transform
n = length(x)/2;
t = linspace(0,1/Fs,n*2); % time array
n = ceil(n); % det number of points and cut in half
amp_spec1 = abs(Y1)/n; % abs val and normalize

freq1 = linspace(0, Fs/2, n); % det frequency resolution
% stem(freq1, amp_spec1(1:n)); % plot from 0 to folding freq.
% xlabel('Frequency (hz)')
% ylabel('Amplitude')


figure
subplot(2,1,1)       % add first plot in 2 x 1 grid
plot(t, x)
xlim([0 2e-7])
ylim([-2 2])
title('Subplot 1 - Magnitude vs. Time')
xlabel('Time (s)')
ylabel('Magnitude')

subplot(2,1,2)       % add second plot in 2 x 1 grid
stem(freq1, amp_spec1(1:n))      % plot using + markers
title('Subplot 2 - Linear Magnitude vs. Frequency')
xlabel('Frequency (hz)')
ylabel('Amplitude')
```

2. **Create a subplot that contains the time domain in one plot and the frequency domain in a second plot.**

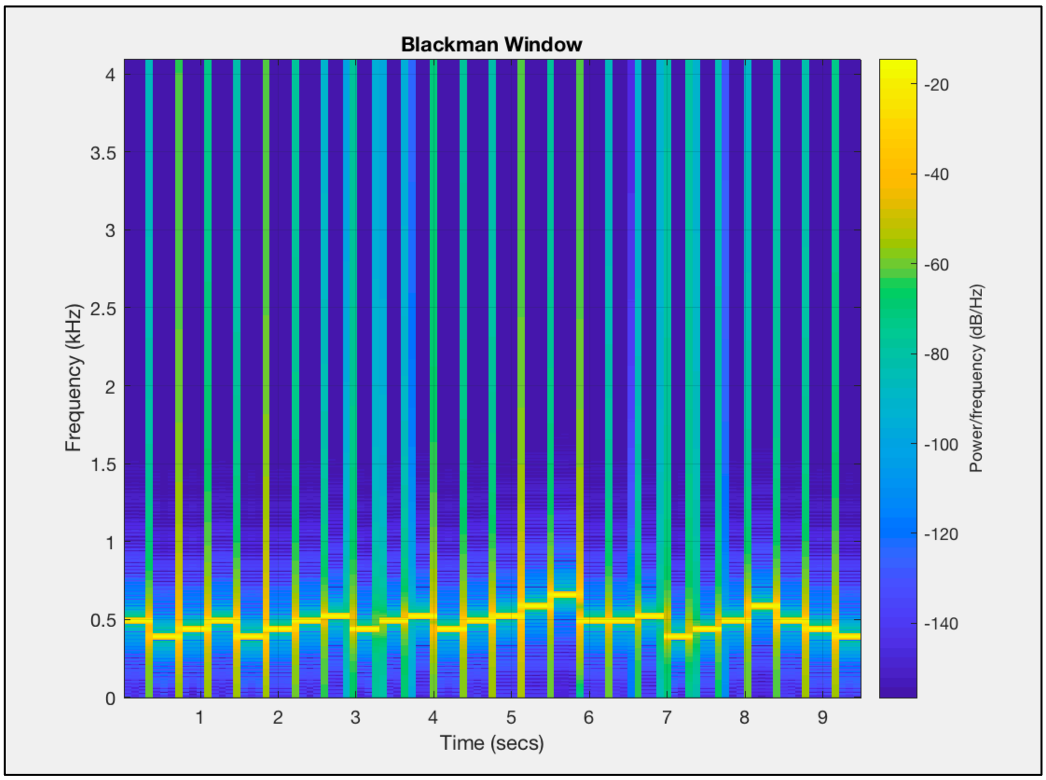
```
%Problem 2
figure
subplot(2,1,1)       % add first plot in 2 x 1 grid
plot(t, x)
xlim([0 2e-7])
ylim([-2 2])
title('Subplot 1 - Magnitude vs. Time')
xlabel('Time (s)')
ylabel('Magnitude')

subplot(2,1,2)       % add second plot in 2 x 1 grid
stem(freq1, amp_spec1(1:n))      % plot using + markers
title('Subplot 2 - Linear Magnitude vs. Frequency')
xlabel('Frequency (hz)')
ylabel('Amplitude')
```



Subplot 1 - Magnitude vs. Time
Magnitude
2
1
0
-1
-2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time (s)
×10^-7
Subplot 2 - Linear Magnitude vs. Frequency
Amplitude
0.4
0.3
0.2
0.1
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Frequency (hz)

3. **Perform a Short Time Fourier Transofrm on the signal, using the MATLab command *spectrogram.* Given some conditions.**

```
% Blackman Window
% Window Size: 0.125 seconds
size = 0.125*Fs;
% 25% overlap
% Double oversampling

figure
spectrogram(x, blackman(size), .25*size, n*2, Fs, 'yaxis');
title('Blackman Window')
grid on
```



4. **Using the results, determine the dominant frequency for each 0.125 seconds. Create a table that contains the following: time frame, dominant frequency.** ***For extra credit determine the notes that are played during each time interval and add them to the table.***

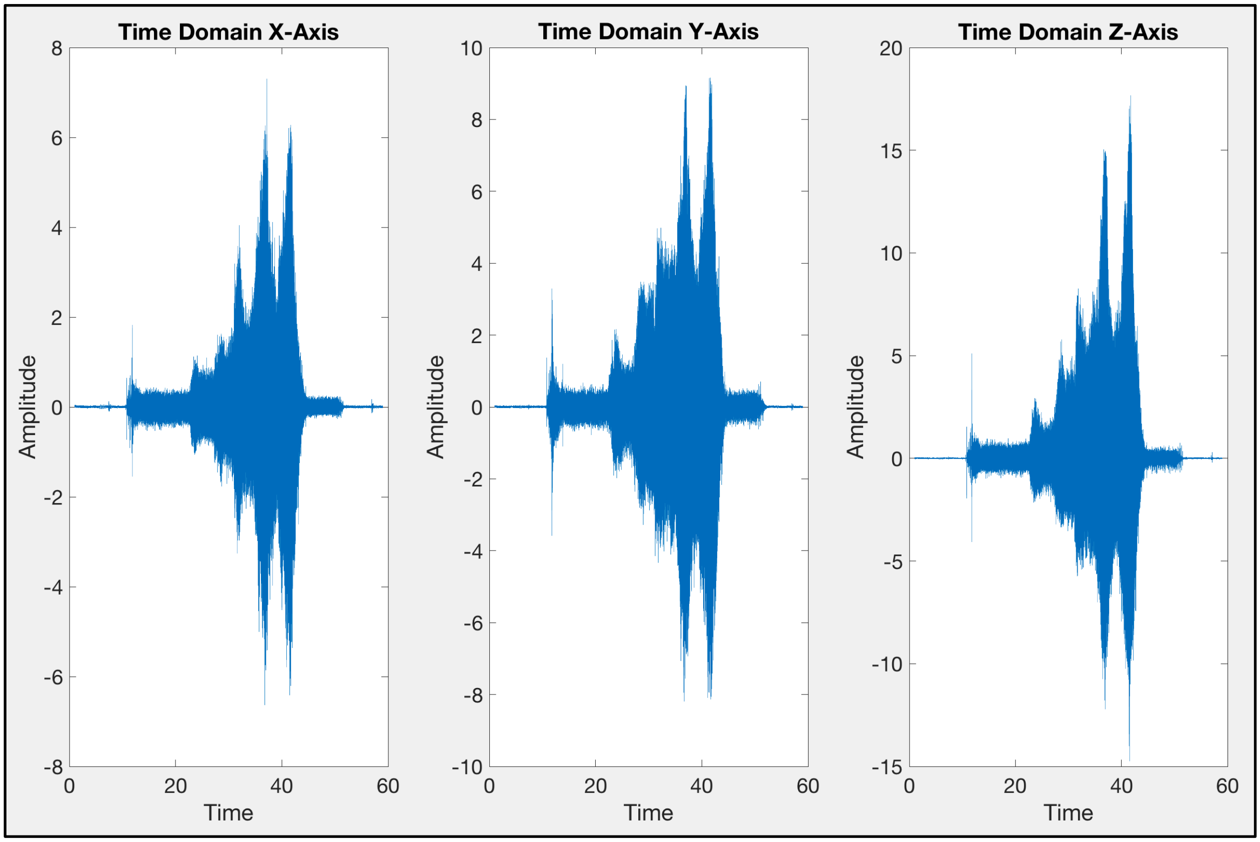
Number	Time	Frez (kHz)	Frez (Hz)	Note:
1	0.375	0.5	500	B
2	0.75	0.3866	386.6	G
3	1.125	0.439	439	A
4	1.5	0.5	500	B
5	1.875	0.3866	386.6	G
6	2.25	0.439	439	A
7	2.625	0.5	500	B
8	3	0.5196	519.6	C
9	3.375	0.439	439	A
10	3.75	0.5	500	B
11	4.125	0.5196	519.6	C
12	4.5	0.439	439	A
13	4.875	0.5	500	B
14	5.25	0.5196	519.6	C
15	5.625	0.5389	538.9	C#
16	6	0.6564	656.4	E
17	6.375	0.5	500	B

5. **Compare and contrast the information you can obtain from the FFT with that obtained from the STFT.**

While the FFT does take the signal and parse out what frequencies are most present, it is clear that the STFT is more powerful when analyzing the whole signal. It is much more powerful when looking at specific signals are most prominent *with respect to time,* which the FFT cannot do, as it looks at the whole signal. Picking out notes from the signal with respect to time would not have been possible if just using a FFT.

Car Engine Data

1. **Using the data create a subplot that has three rows and three columns for the entire data set. Use the MATLAB command *figure (1)* to control location of subplots to the figure containing the full data set.**

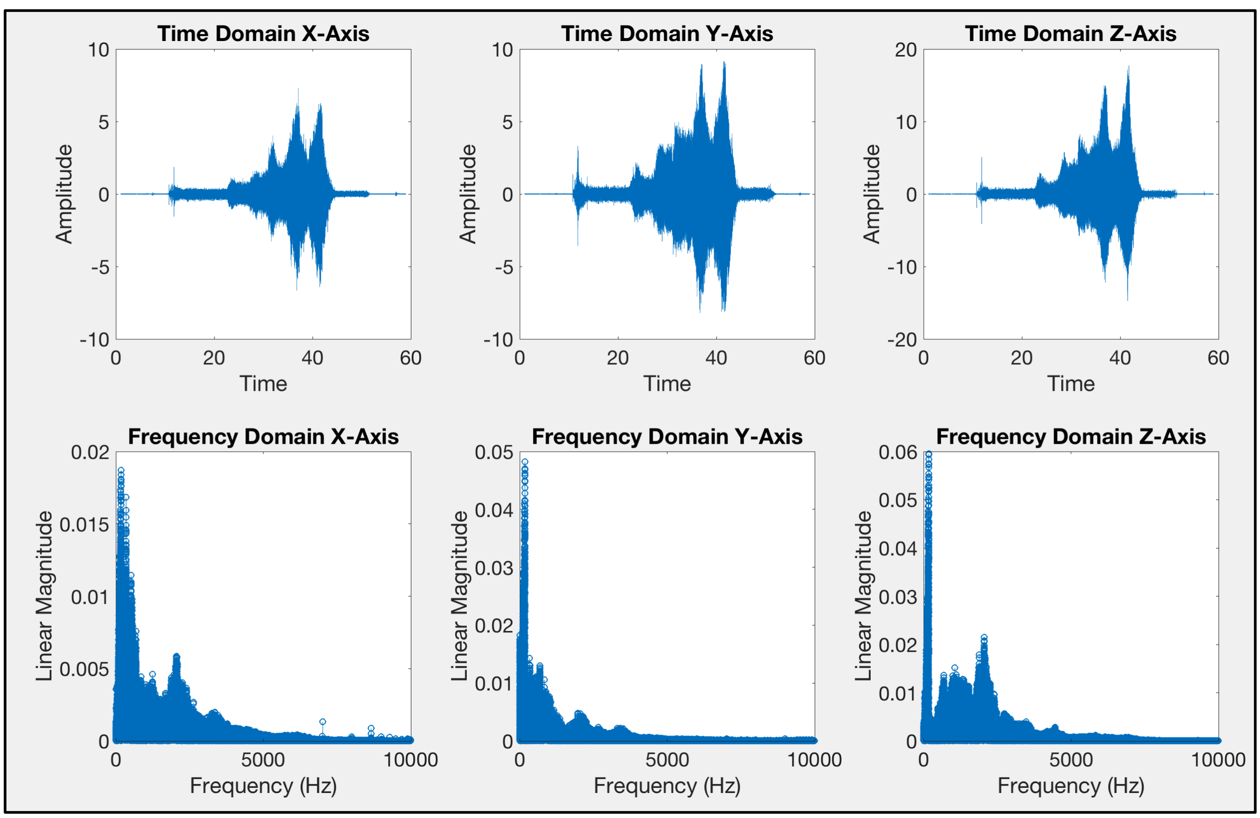


2. **Using MATLAB, convert all three channels for entire signal into the frequency domain using the FFT. You should include a copy of the code used.**

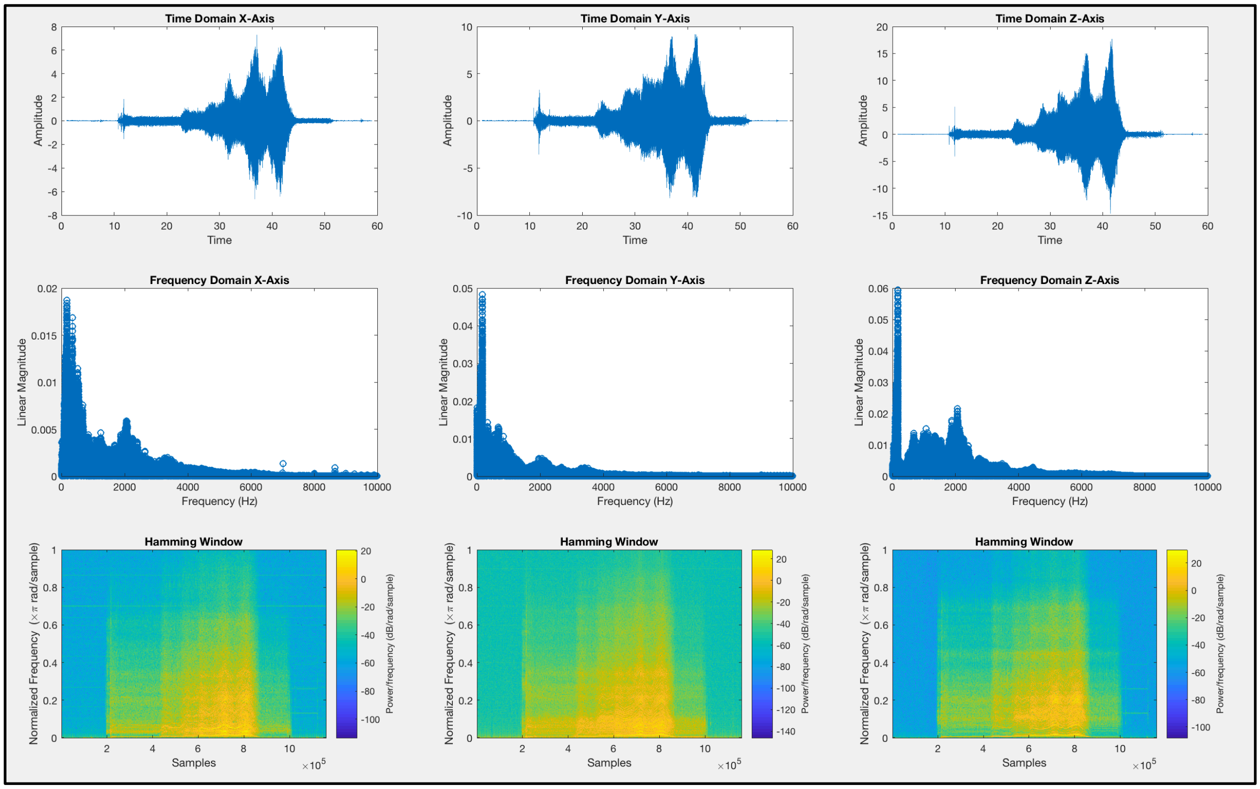
MATLAB Code Example

```
%Frequency Domain
subplot(2,3,4)
Yx = fft(acc_x);
n = length(acc_x)/2;
n = ceil(n);
freq = linspace(0,FoldingFreq,n);
amp_specx = abs(Yx)/n;
stem(freq,amp_specx(1:n));
xlim([0 10000]);
title('Frequency Domain X-Axis','fontsize',24);
xlabel('Frequency (Hz)','fontsize',22);
ylabel('Linear Magnitude','fontsize',22);
set(gca,'FontSize',20);
```

3. **Using the results plot the frequency domain for each axis. The plot should be frequency on the x-axis and linear magnitude on the y-axis. Plot the entire frequency range from zero to the folding frequency.**



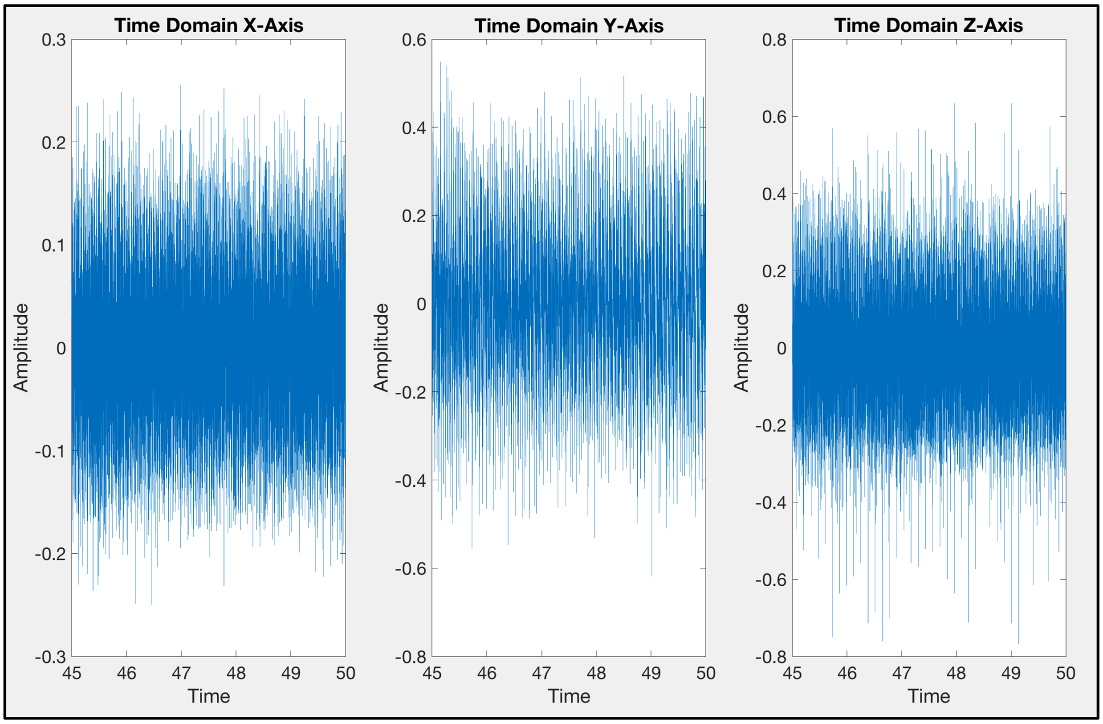
4. Using the data create a STFT for the entire data set, one for each channel: Hamming Window, Window size 1 second, 25% overlap, no oversampling.



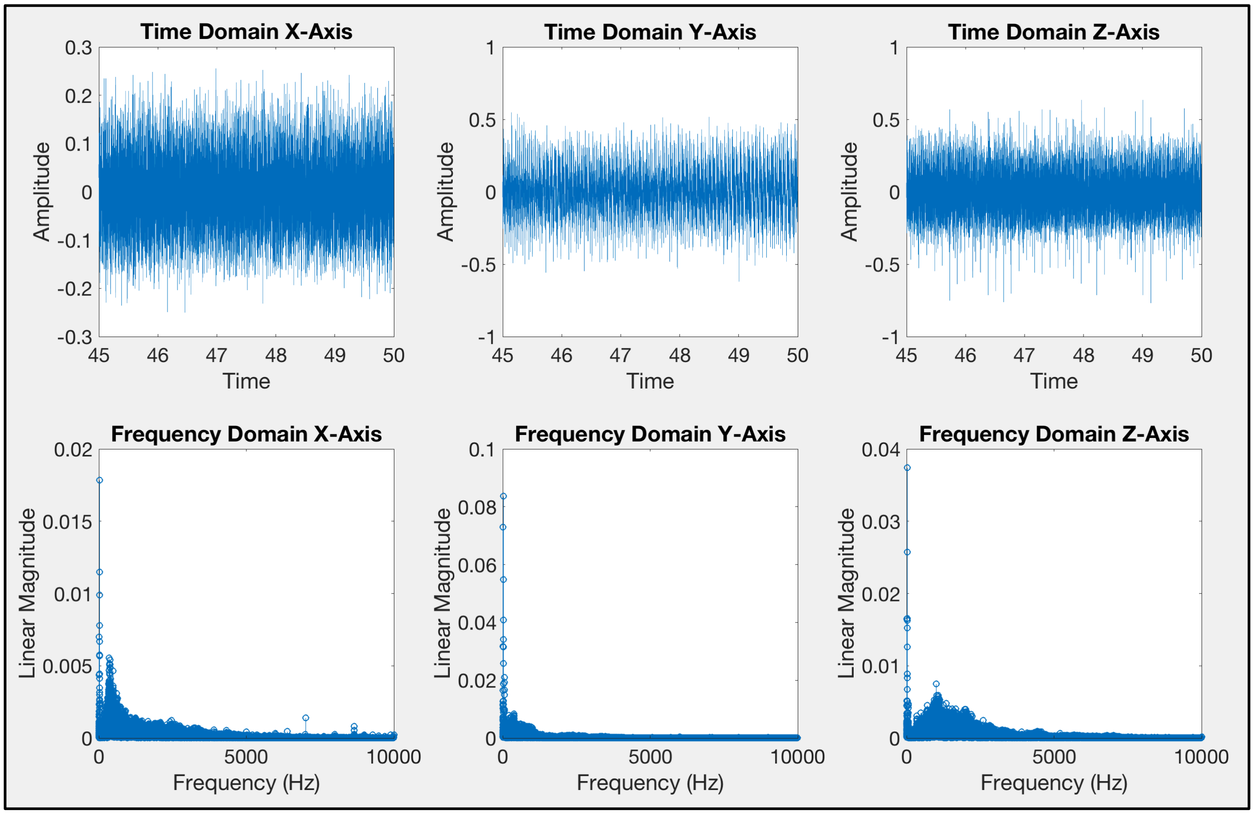
5. Using the information obtained from above, prepare a brief discussion on the overall vibrational profile one would expect to see under similar running conditions.

From the information obtained in problem 4, it appears that there are a significant amount of vibrations found in all directions, however, from the time domain graphs that there are more vibrations in the y and z direction compared to the x direction. For the time domain graphs, it is more difficult to determine exactly which direction contains the most vibrations. Taking the time domain graphs to the frequency domain provide a better understanding of the frequency components found in each direction. With the frequency domain plots, it appears that the z direction has the most vibrations followed by the x direction and then finally the y direction. In conclusion the time domain plots provide an acceptable idea of which direction contains the greater vibration, then taking the data to the frequency domain allows to gain a deeper understanding of the frequencies present, which allows for an easier way to determine the vibrational profile. With similar running conditions, one could expect there to be the most vibration occurring in the z direction followed by the x and y directions, respectively.

6. Between 45 and 50 seconds of the data set, the car was running at idle. With this data set, this occurs between data points (45 * 20000) and (50 * 20000). Use the MATLAB command *figure (2)* to control location of subplots to the figure containing the zoomed data set.



7. **Transform the data to the frequency domain, using the results plot the frequency domain for each axis. The plot should be frequency on the x-axis and linear magnitude on the y-axis. Plot the entire frequency range from zero to the folding frequency.**



8. **Using the information obtained from above, prepare a brief discussion on the overall vibrational profile one would expect to see while the car is idling.**

For the overall vibrational profile for the car at idle, there was a greater vibration in the y and z directions. This conclusion was determined because the amplitude in the y and z directions were approximately 67% greater than the x direction. These conclusions that were made about the vibrational profile of the system that were determined from the time domain graphs were not extremely clear, however, after reviewing the frequency domain plots it was more explicit that there was greater vibration in the y and z directions than the x direction. The frequency domain plots presented the different frequencies for the three orientations and there was indeed greater frequencies found in the z direction. In conclusion after reviewing the frequency domain plots for the three directions, the most vibrations occurred in the z direction.