Lab 5 Part 1

Contents

- 1.1 - Definition: What is the FFT?
- 1.2 - Background: How does the FFT "Read" Frequency Information?
- 1.3 - Exercise: Decomposing Signals into Sinusoids
- 1.4 - FFT of a Single Sinusoid with Integer Frequency
- 1.5 - FFT of Multiple Sinusoids with Integer Frequency


```
clear
```

1.1 - Definition: What is the FFT?

FFT is an acronym for the Fast Fourier Transform, an algorithm that computes the discrete Fourier transform (DFT) of a discrete-time signal. In other words, the FFT takes a discrete-time signal and extracts important frequency domain information from that signal. It is considered "fast" because its computational complexity is far less than that of the DFT, the Discrete Fourier Transform (see https://en.wikipedia.org/wiki/Fast_Fourier_transform). In this lab, we won?t care about the mechanics of the algorithm (hurrah!); we are simply concerned with how to use the FFT function in MATLAB to determine frequency domain information of signals. IMPORTANT: The FFT function performs the DFT of a signal; the FFT is specific to an input signal with length 2^n, but the FFT function in MATLAB can be used for input signals that do not meet this criteria: this just performs a DFT (see help fft for more information). The two terms are used interchangeably throughout the lab.

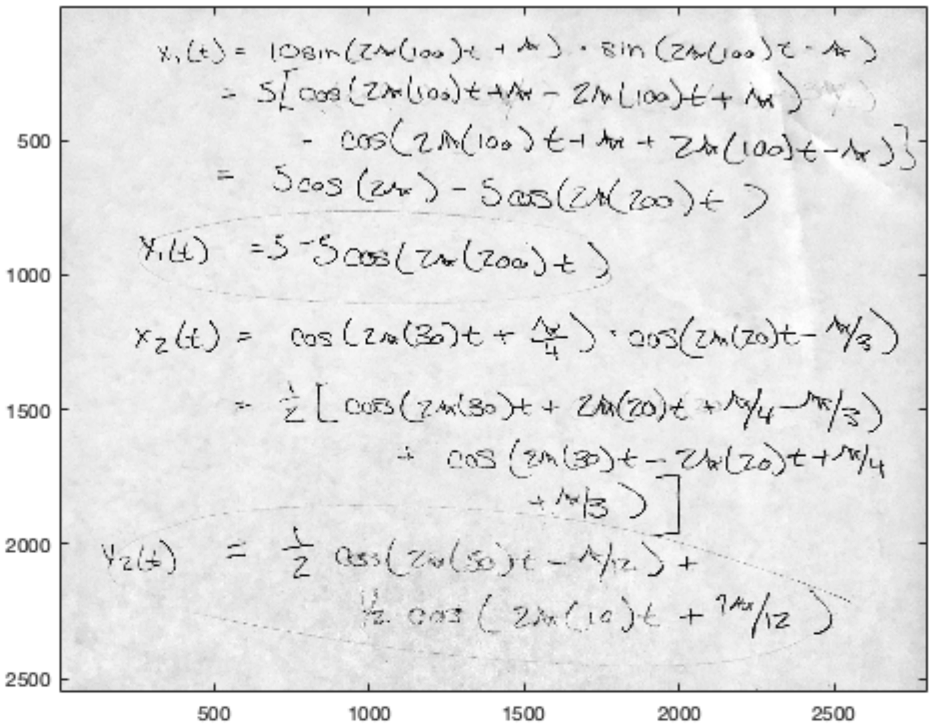
1.2 - Background: How does the FFT "Read" Frequency Information?

The FFT utilizes a series of sinusoidal basis signals of varying frequency. The FFT determines how closely the input signal "matches" these basis signals, and outputs a vector of complex numbers - one for each iteration of the algorithm. In this lab, we will call these complex measurements frequency bins, or simply bins.

Because the outputs of the FFT are complex, we need an easier way to visualize them. Simply plotting the vector of frequency bins will result in a complicated jumble of vectors in the complex plane. Therefore, we "read" the results of the FFT by plotting the magnitude spectrum of the frequency bins. The higher the magnitude of the frequency bin, the higher the correlation between the input signal and a sinusoid of that frequency. Therefore, the magnitude spectrum should spike at regions where the frequency of the sinusoidal basis signals match the input signal. Recall that any signal can be expressed as the sum of sinusoids of different frequencies. Consequently, the FFT is an extremely powerful tool because it can help us decompose a complicated discrete-time signal into its sinusoidal components.

1.3 - Exercise: Decomposing Signals into Sinusoids

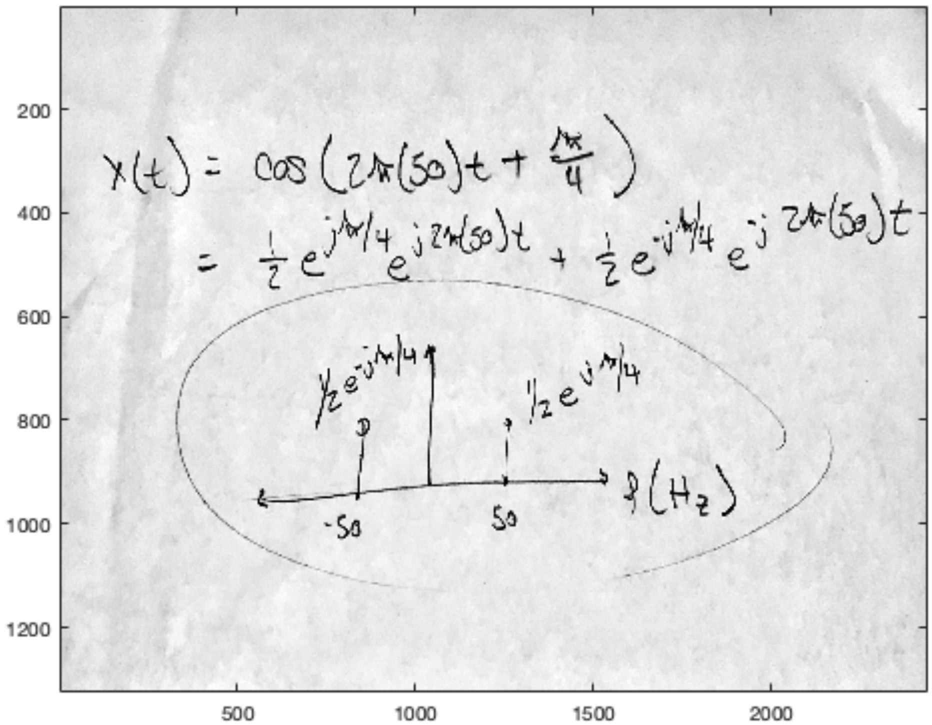
```
img = imread("5.1.1.3.jpg");
image(img);
```



1.4 - FFT of a Single Sinusoid with Integer Frequency

Plot the continuous-time frequency spectrum of this signal by hand.

```
img = imread("5.1.1.4.jpg");
image(img);
```



1.4.1 Define the variable fs for the sampling frequency, and assign it 1000 samples/second.

```
fs = 1000;
```

1.4.2 Construct a time vector using this sampling frequency that is exactly 1500 samples long, starting at zero.

```
tt = 0:1/1000:1/1000*1499;
```

1.4.3 Define x(t) , the sinusoid described above, as the vector x.

```
xx = cos(2*pi * 50 * tt + 0.25*pi);
```

1.4.4 Run the fft() function on x (see help fft).

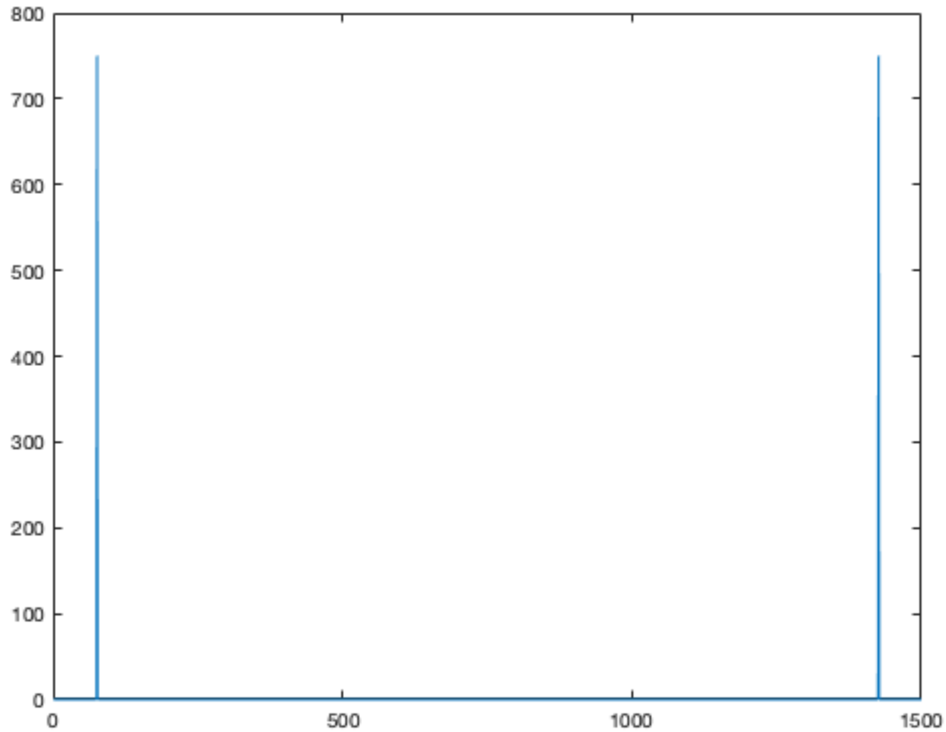
```
XX = fft(xx);
```

1.4.5

1.4.5.a

```
magX = abs(XX);
```

```
plot(magX)
```



1.4.5.b Compare this plot to your continuous-time spectrum. Question 1: What's different about it?

This is similar to my continuous-time spectrum as there are two lines, one representing -50 Hz and one representing 50 Hz. However, the magnitude spectrum is different as the magnitude of these lines are much larger.

1.4.5.c

```
maxValue = max(magX)
```

```
maxValue =

  750.0000
```

Question 2: What is the index corresponding to the left peak?

```
maxValueIndexes = find(magX == maxValue);
maxValueIndexes(1)
```

```
ans =
```

```
    76
```

Question 3: What is the value of the magnitude spectrum everywhere else?

Approximately zero.

```
magX(maxValueIndexes(1)-1)
```

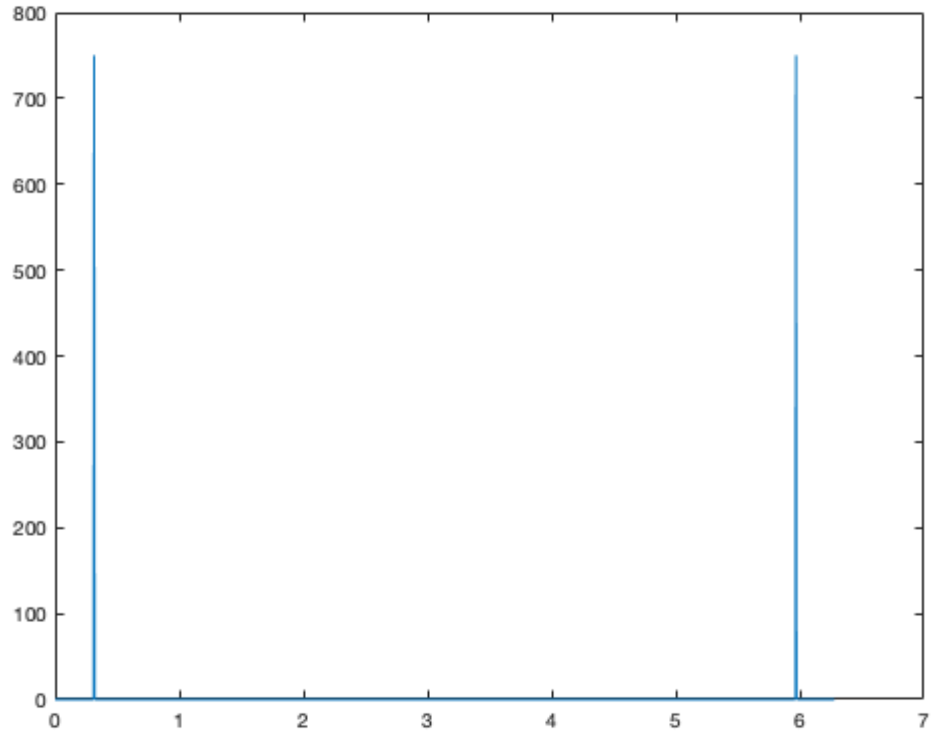
```
ans =

   1.3888e-11
```

1.4.5.d

The result of the Fourier transform of a sampled signal goes into frequency bins that correspond to the normalized radian frequency. Our convention thus far has been to draw magnitude spectra from -pi to pi. The FFT function, however, returns frequency bins ranging from 0 to 2*pi.

Plot FFT output against normalized radian frequency.

```
ww = 0:(2*pi/length(XX)):(2*pi-1/length(XX));
plot(ww,abs(XX));
```

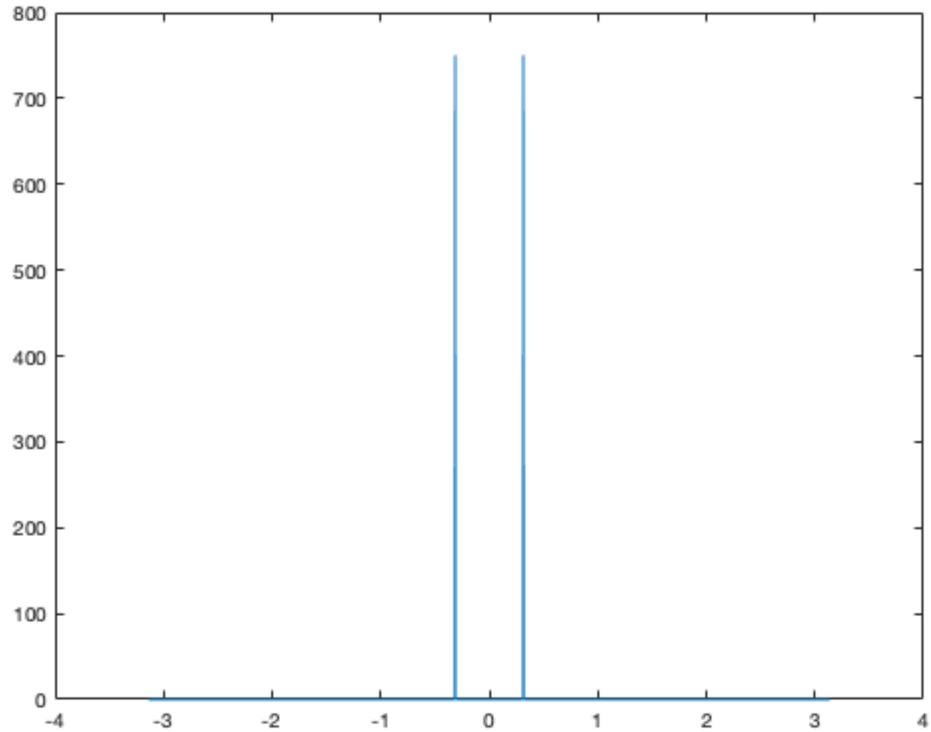


1.4.5.e

The fftshift() function (see help fftshift) is a way to "fix" the output of the FFT function in order to make it range from ?pi to pi.

Plot the shifted FFT output against normalized radian frequency:

```
ww = -pi:(2*pi/length(XX)):(pi-1/length(XX));
XX = fftshift(XX);
plot(ww,abs(XX));
```

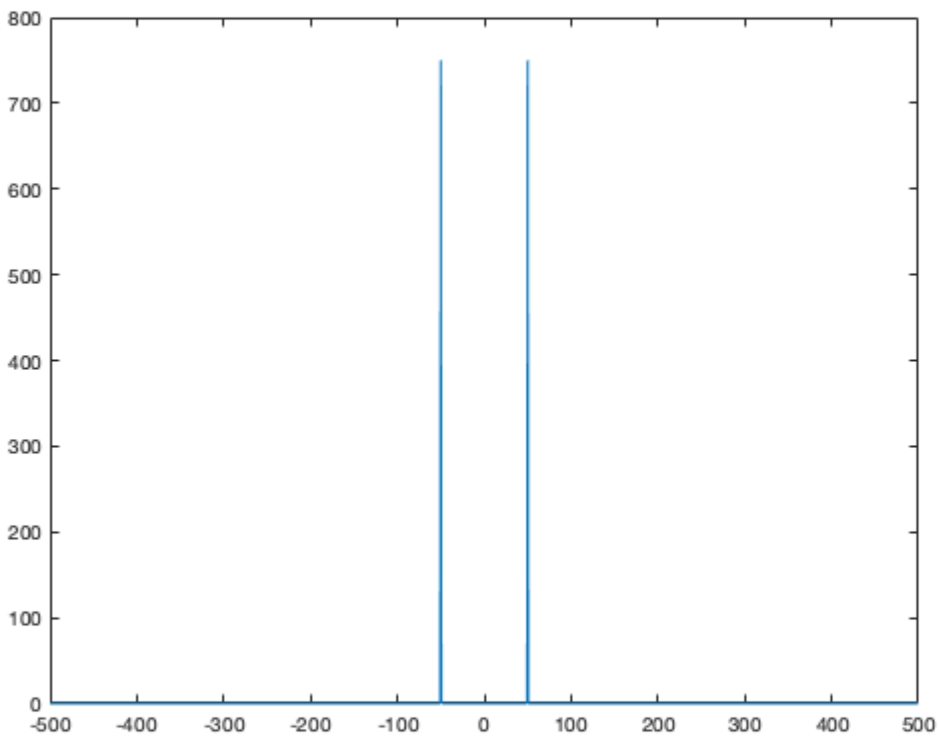


1.4.5.f

The normalized radian frequency is related to the frequency in Hz by the sampling frequency and a factor of 2*pi.

Plot the FFT of X against the Hertz frequencies of the bins.

```
ww = -pi:(2*pi/length(XX)):(pi-1/length(XX));
ff = fs * ww / (2 * pi);
plot(ff, abs(XX));
```

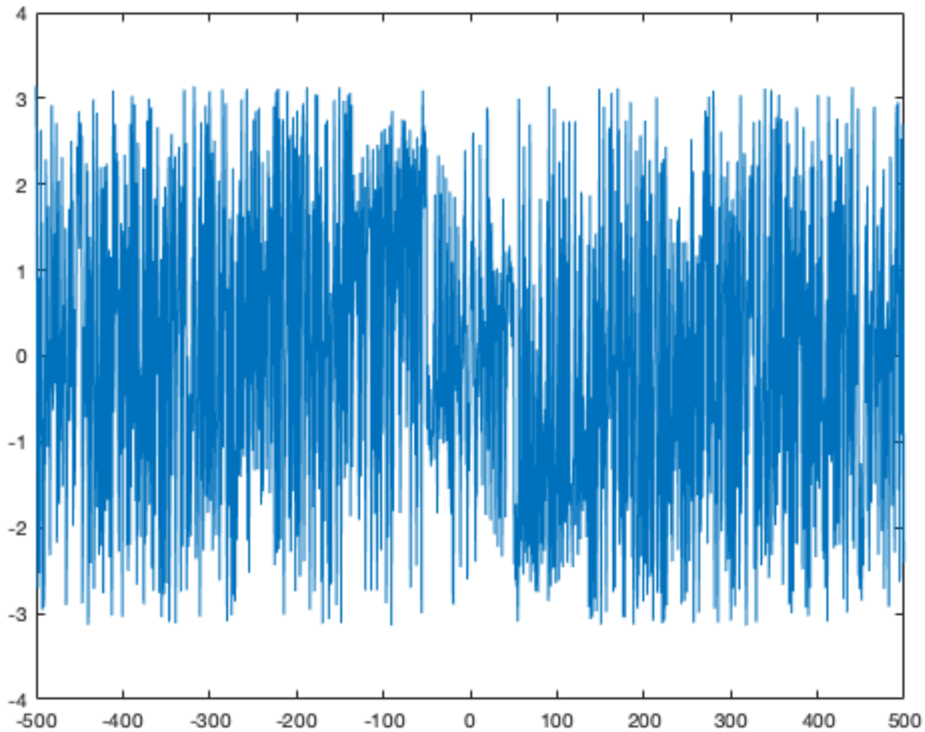


Question 4: What are the frequencies matching the peaks?

The frequencies mathcing the peaks are approximately +-50.

1.4.6 The FFT can also be used to extract phase information from the signal. From the complex amplitude of the FFT function, you can use the angle() function (see help angle) to find the phase in each bin. Plot the phase spectrum of the signal against Hertz frequency.

```
ff = fs * ww / (2 * pi);
plot(ff, angle(XX));
```



Question 5: What are the phases of the peaks?

```
ang = angle(XX);
peakIndex = find(abs(XX) == max(abs(XX)));
ang(peakIndex)
```

```
ans =

   -0.7854    0.7854
```

Question 6: Does this match what you would expect, and why?

Yes, this does match what I would expect. The decimal values +- 0.785 correspond to +- pi/4. This was the same phase that we deduced when we drew the continuous time frequency spectrum earlier.

1.4.7

In parts [1.4.5] and [1.4.6], we identified the frequency and phase of x(t). It turns out that we can also use the FFT to determine the exact magnitude of x(t), by finding the value of the magnitude spectrum in that frequency bin.

Determine the exact magnitude of the peak. Note that the two peaks will have the same magnitude, being complex conjugates of each other. Recall that the magnitude of the peak is directly related to the length of the DFT.

As was determined previously and can be seen below, the exact magnitude of both peaks is equal to 750.

```
max(abs(xx))
```

```
ans =

    0.9877
```

1.5 - FFT of Multiple Sinusoids with Integer Frequency

1.5.1

Using the same sampling frequency (1000 samples/s) and duration (1500 samples) from part 1.5, create the vectors x1 and x2 , storing 1500 samples of the signals x1(t) and x2(t).

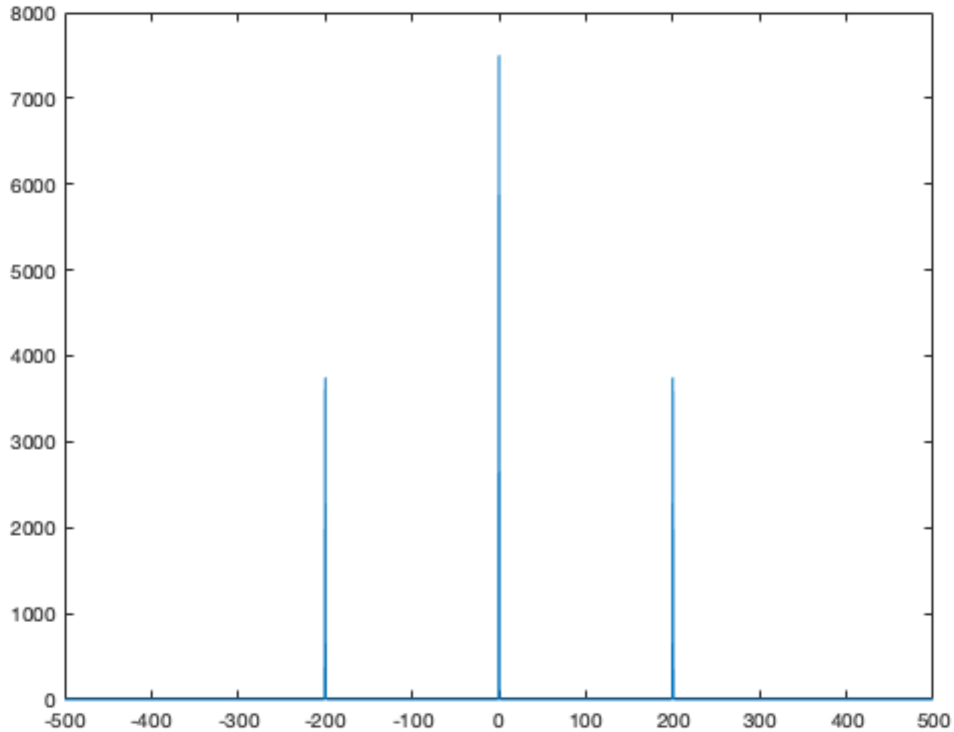
```
fs = 1000;
tt = 0:1/1000:1/1000*1499;

x1 = 10 * sin(2*pi*(100)*tt + pi).*sin(2*pi*(100)*tt - pi);
x2 = cos(2*pi*(30)*tt + 0.25*pi).*cos(2*pi*(20)*tt - (1/3)*pi);
```

1.5.2

Plot the magnitude spectra of x1 and x2 against Hertzian frequency. Unlike in 1.5, we should observe multiple peaks, each corresponding to a single sinusoidal component. Comment on the appearance of each spectrum.

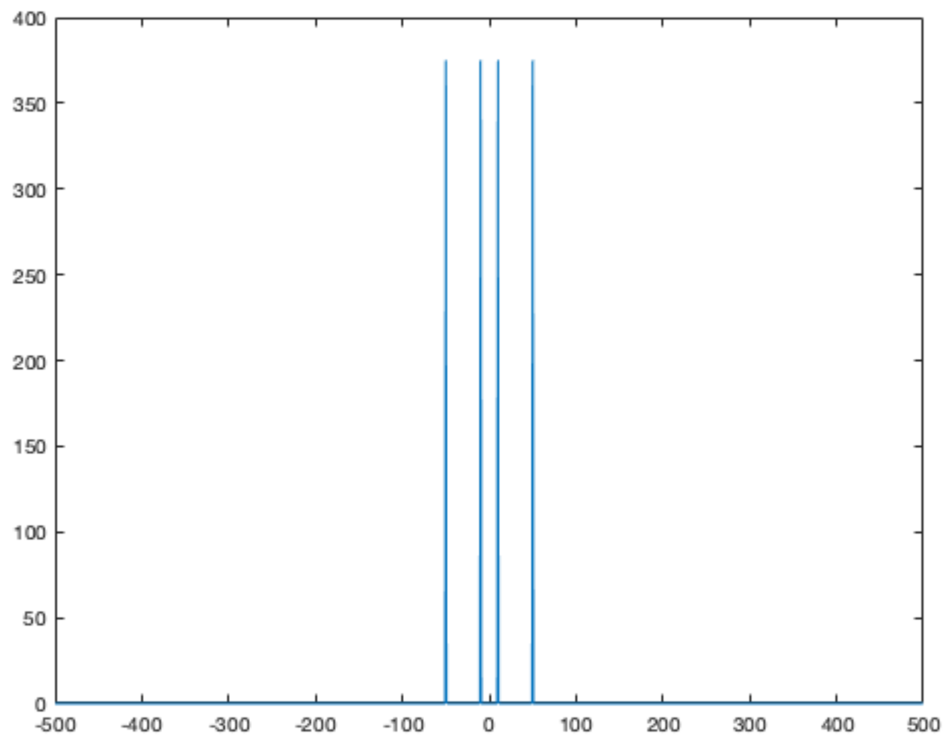
```
XX1 = fftshift(fft(x1));
XX2 = fftshift(fft(x2));

ww = -pi:(2*pi/length(tt)):(pi-1/length(tt));
ff = fs * ww / (2 * pi);
```

```
plot(ff, abs(XX1));
```



Here, there are three peaks. There is one large peak at 0 Hz, and two peaks at +-200 Hz.

```
plot(ff, abs(XX2));
```



Here, there are 4 peaks. One a little bit larger than 0 Hz, one a little bit smaller than 0 Hz, one at approximately 50 Hz, and one at approximately -50 Hz.

Question 7: Do you notice any differences in the number of peaks for these signals?

Yes I do, I see that there are three peaks for x1, while there are four peaks for x2.

1.5.3

Using MATLAB, find the frequencies, amplitudes, and phases of the sinusoidal components of x1 and x2. Compare these results to what you found by hand.

```
ang1 = angle(XX1);
mag1 = abs(XX1);
indexes1 = find(mag1 > 0.1); % the nonzero values of the mag1 response

ang1(indexes1)
mag1(indexes1)/1500
ff(indexes1)
```

```
ans =

    3.1416         0   -3.1416
```

```
ans =

    2.5000    5.0000    2.5000


ans =

 -200.0000   -0.0000  200.0000
```

From observing the above, one can see that we have a purelry real DC component with a phase of 0, a frequency of 0, and an amplitude of 5. We also have two complex sinusoids, and after applying the inverse Euler Formula, we see that we have another real sinusoid with a phase of pi, a frequency of 200 Hz, and an amplitude of 5.

```
ang2 = angle(XX2);
mag2 = abs(XX2);
indexes2 = find(mag2 > 0.1); % the nonzero values of the mag2 response

ang2(indexes2)
mag2(indexes2)/1500
ff(indexes2)
```

```
ans =

    0.2618   -1.8326    1.8326   -0.2618


ans =

    0.2500    0.2500    0.2500    0.2500


ans =

  -50.0000  -10.0000   10.0000   50.0000
```

Similairly, from observing the above, we can see that we have 4 complex sinusoids, forming pairs with the Inverse Euler Formula to create two real valued sinusoids. Both of these real sinusoids have an amplitude of 0.25*2 = 1/2. One has a phase of -pi/12 and a frequency of 50 Hz, while the other has a phase of 7pi/12 and a frequency of 10 Hz.

Published with MATLAB® R2019a