

## Lab 02: An Experiment with Tuning Fork Synthesis of Sinusoidal Signals

### Introduction

For lab 02, we became familiar with tuning forks. Tuning forks are systems that produce sinusoidal signals when vibrated. We first took a recording with a microphone using the `ainput` function, which was then generated into an electrical signal. The signal was used for the various plots at different sampling frequencies seen later in this lab report.

### Signal Collection: Step 1

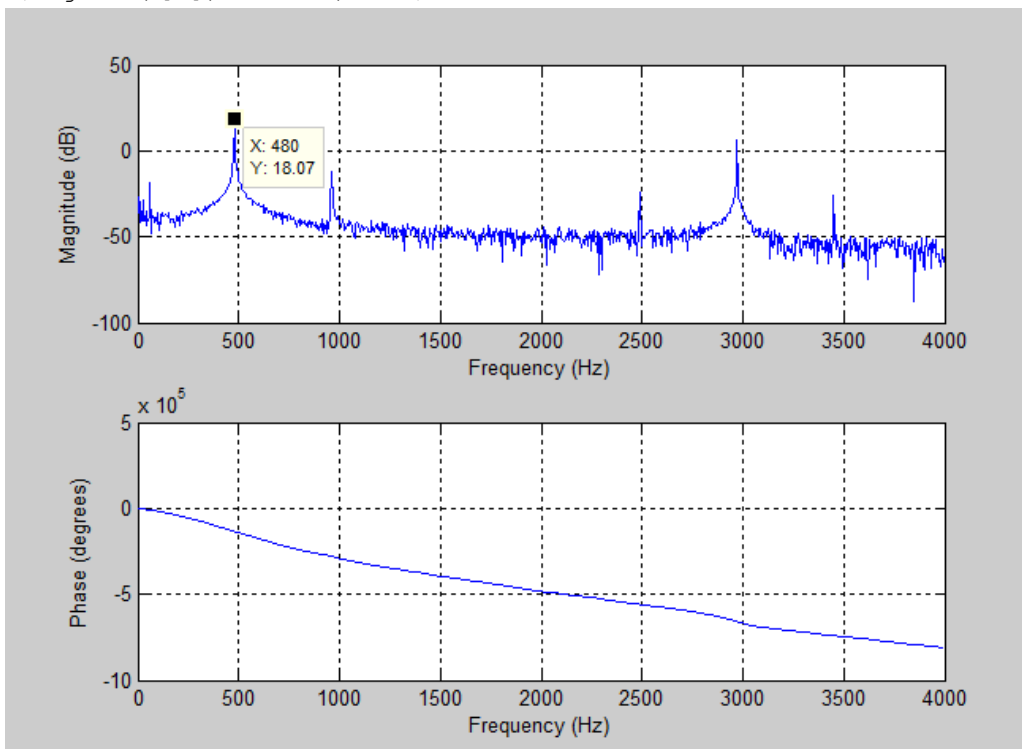
In this step, we used the `ainput` function given to us in order to save and collect the tuning fork signals. The two tuning forks I used for my frequencies were 480 and 271 Hz. I decided that one second was sufficient for my sampling period. Below, you will find how I executed the `ainput` file using `signal1` for my 480 Hz signal and `signal2` for my 271 Hz signal.

```
signal1 = ainput(8000,1,'fork1');  
signal2 = ainput(8000,1,'fork2');
```

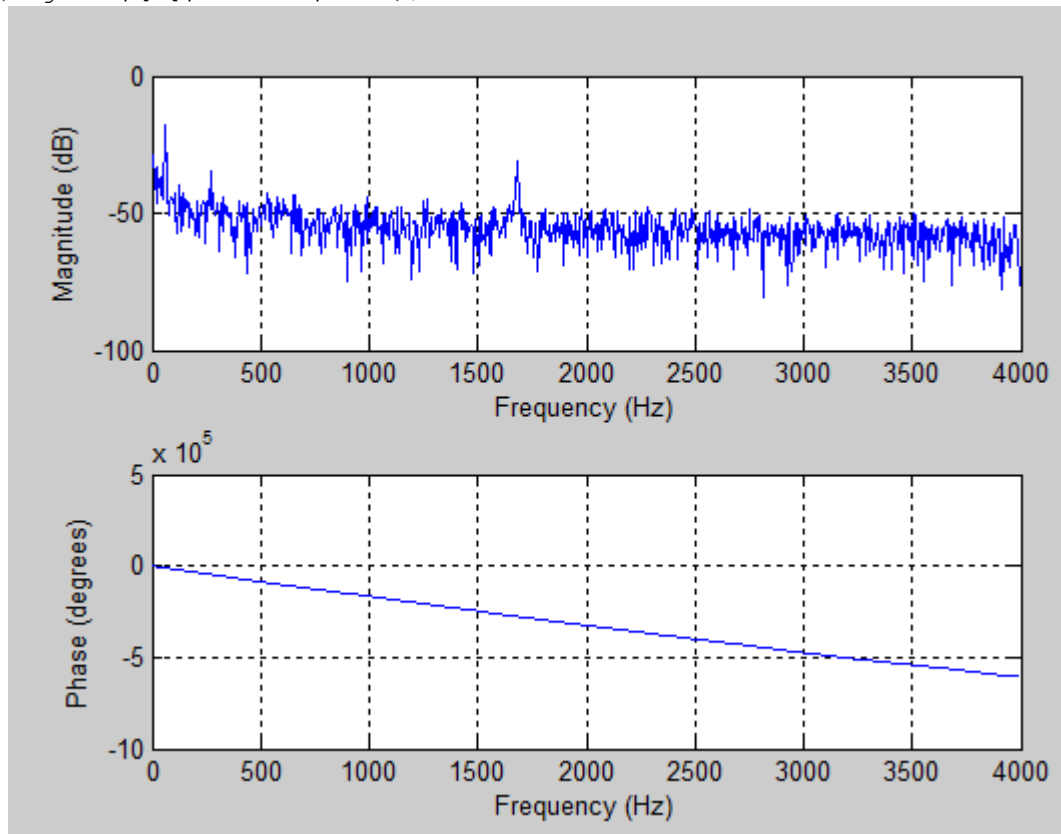
### Signal Collection: Step 2

Next, we measured the frequencies of these signals using a plot of both signals. I did this by using plotting the Time vs. Tuning Fork Response. Also, you will find the `freqz` function used to plot the frequency spectrum of a signal. I made a point to label all axes along with the time units. Below these frequency graphs, you will find my calculations of how I measured the signals. In addition, you will find the code I used to find the frequency response vs. time for both signals to be able to compare them.

```
freqz(signal1,[1],8000/10,8000);
```

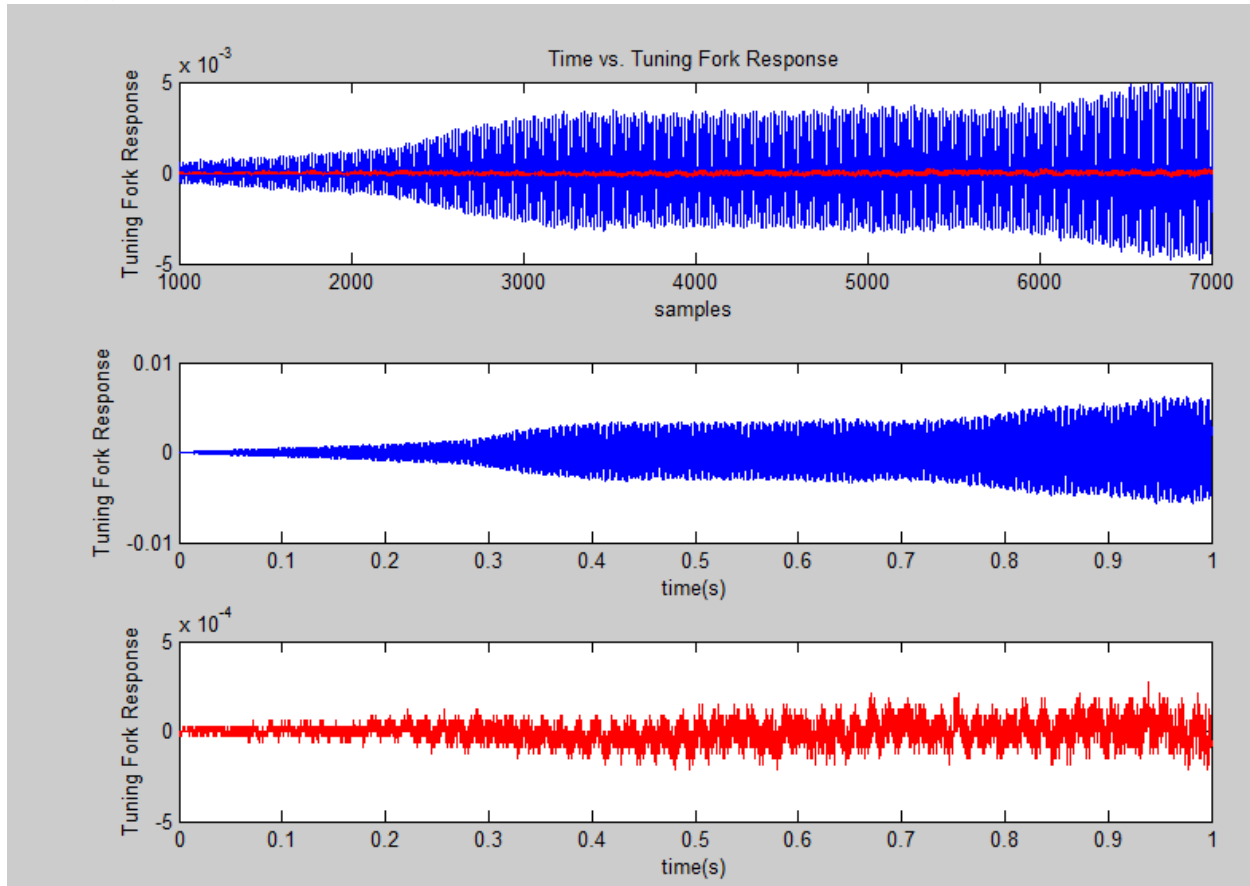


```
freqz(signal2, [1], 8000/10, 8000);
```




I used the following code to develop a Time vs. Tuning Fork Response graph to measure the frequency.

```
subplot(3,1,1);  
plot(signal1, '-b');  
hold on;  
plot(signal2, '-r');  
xlabel('samples');  
ylabel('Tuning Fork Response');  
title('Time vs. Tuning Fork Response');  
axis([1000,7000,-0.005,0.005]);  
tscale1=(1:length(signal1))*(1/8000);  
tscale2=(1:length(signal2))*(1/8000);  
  
subplot(3,1,2);  
plot(tscale1,signal1);  
xlabel('time(s)');  
ylabel('Tuning Fork Response');  
  
subplot(3,1,3);  
plot(tscale2,signal2, '-r');  
xlabel('time(s)');  
ylabel('Tuning Fork Response');
```



For the calculation section, I zoomed in on two peaks of both the signal1 and signal2 graphs. I then found the value of that peak and subtracted it from the next peak. This gives me the time in-between two peaks. Since  $T=1/F$ , I can just find  $1/(\text{time between peaks})$  to get the frequency for that signal. Below, you will find frequency calculations for signal 1 and signal 2:

\*1:   $\rightarrow .002$  is time between 2 peaks

$$\frac{1}{.002} = \text{frequency for sig1}$$

$$= 500 \text{ Hz vs } 480$$

\*1  $.5344 - .538 \rightarrow .0036 \rightarrow \frac{1}{.0036} = F \text{ for sig2}$

$$= 277.78 \text{ Hz vs } .271 \text{ Hz}$$

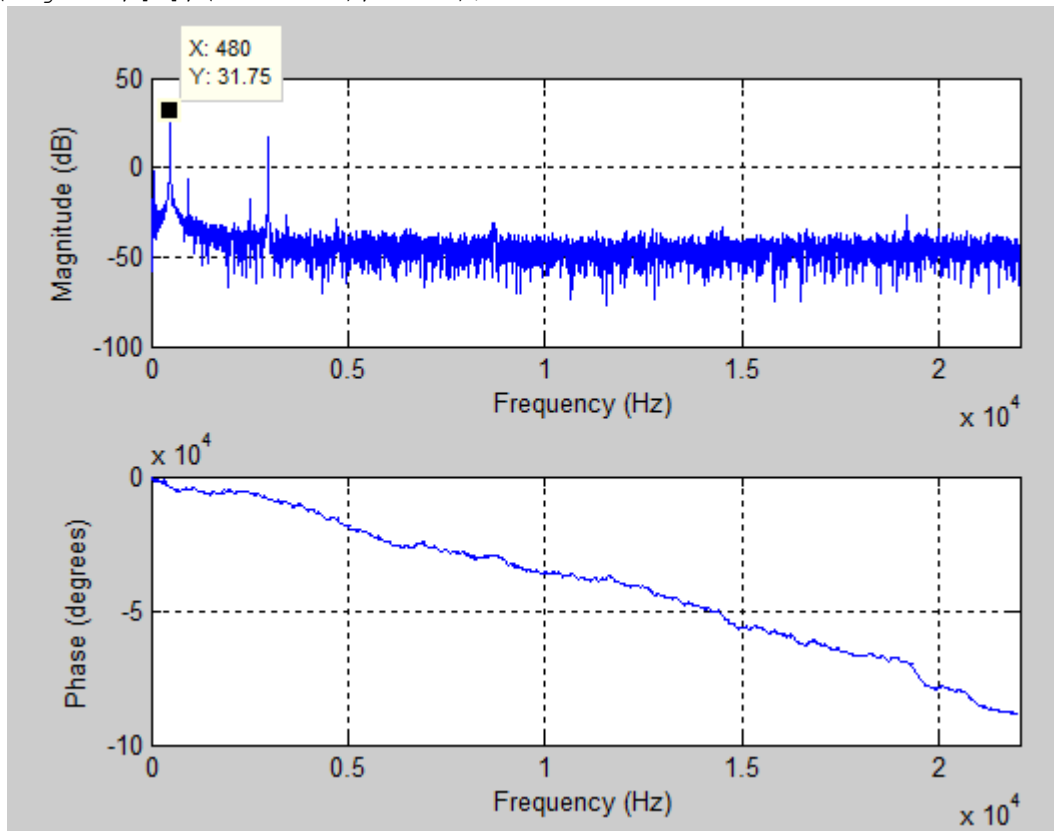
### Signal Collection: Step 3

The measured frequency vs. the tuning fork frequency was very close with very little error. The error for signal1 frequency was only 4.167% and the error for signal2 frequency was 2.50%. There may have been many sources of error during this lab. For instance, noise of the signal may have affected the measurements. Also, the recording may not have been in the quietest environment. Human error and machine error could have also taken place.

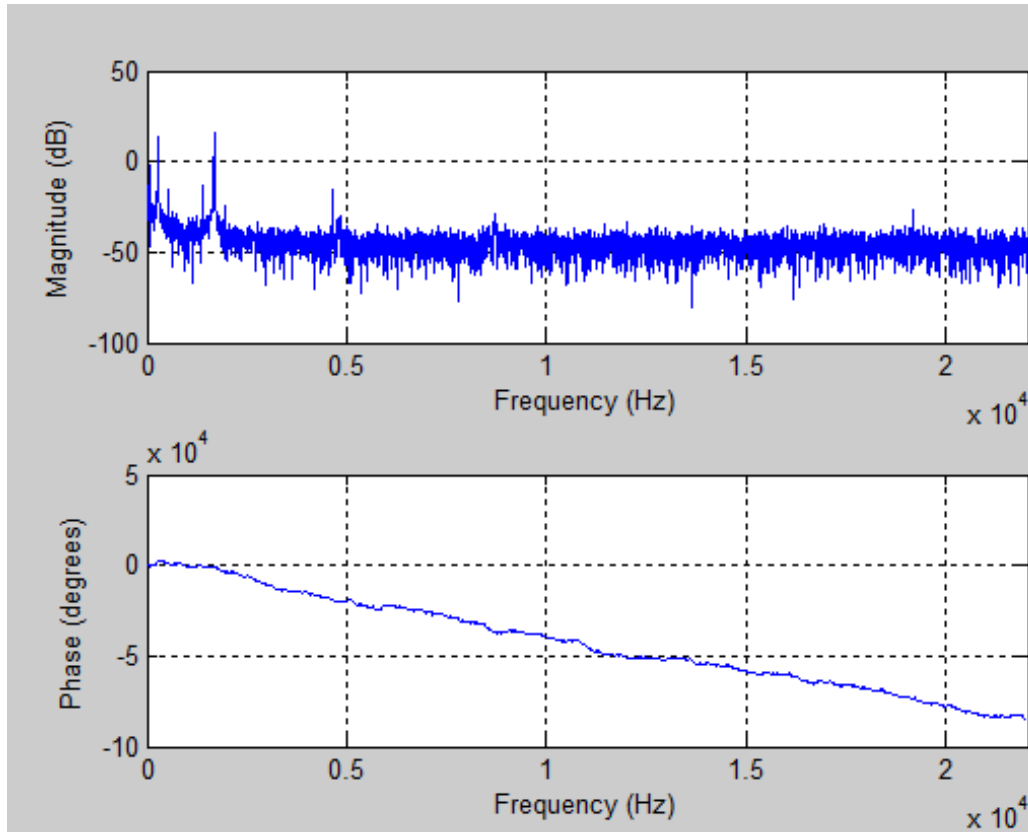
### Signal Collection: Step 4

Once steps 1-2 were finalized, I repeated those steps with the same tuning forks, but using a sampling frequency of 44,100. I would consider my estimates to be more accurate with a higher sampling frequency because there is more data to work with. However, it does get difficult determining the correct peaks to measure from because there are so many.

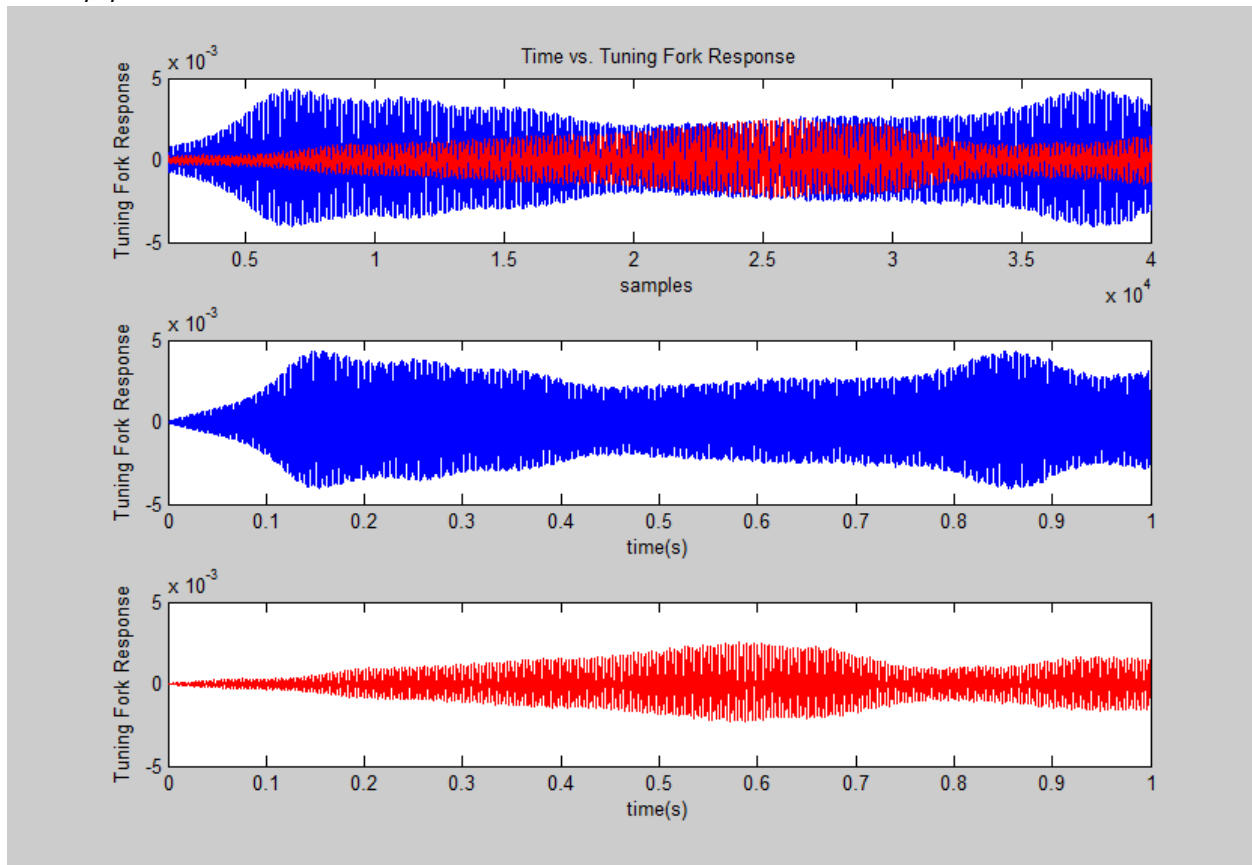
```
signal3 = ainput(44100,1,'fork3');  
freqz(signal3,[1],(44100/10),44100);
```



```
signal4 = ainput(44100,1,'fork4');  
freqz(signal4,[1],44100/10,44100);
```



```
subplot(3,1,1);  
plot(signal3, '-b');  
hold on;  
plot(signal4, '-r');  
xlabel('samples');  
ylabel('Tuning Fork Response');  
title('Time vs. Tuning Fork Response');  
axis([2000,40000,-0.005,0.005]);  
tscale3=(1:length(signal3))*(1/44100);  
tscale4=(1:length(signal4))*(1/44100);  
  
subplot(3,1,2);  
plot(tscale3,signal3);  
xlabel('time(s)');  
ylabel('Tuning Fork Response');  
  
subplot(3,1,3);  
plot(tscale4,signal4, '-r');  
xlabel('time(s)');  
ylabel('Tuning Fork Response');
```



Calculations:

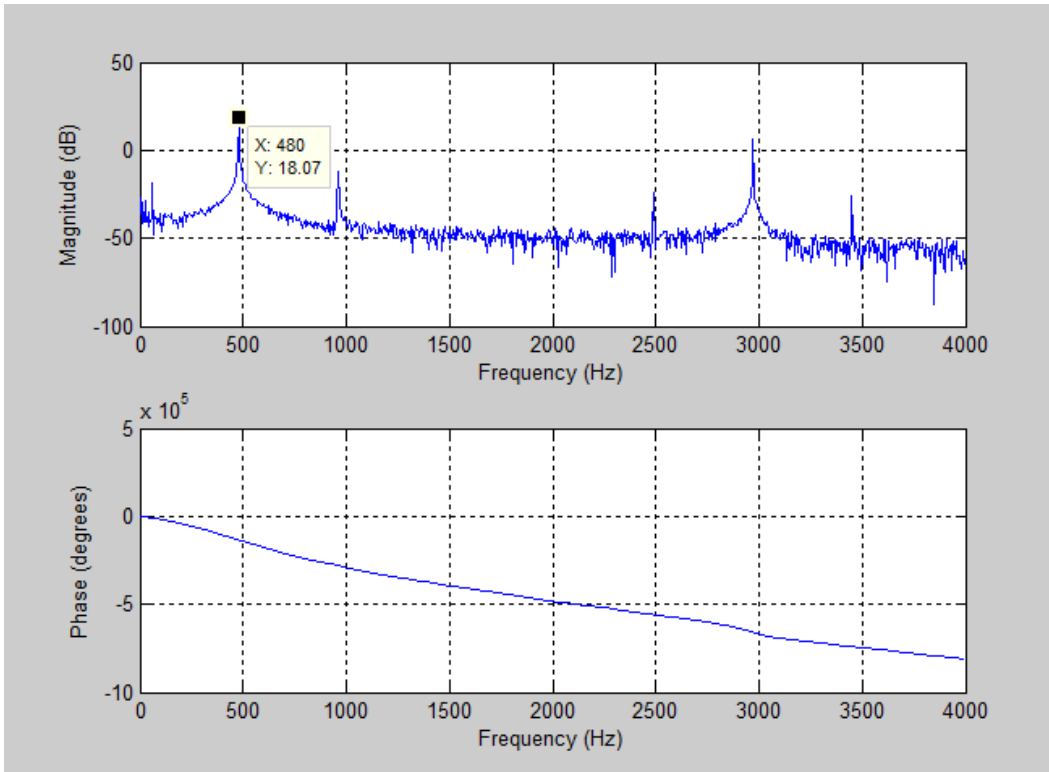
$$.5316 - .5337 \Rightarrow .0021 \quad \frac{1}{.0021} = 476.19 \text{ Hz}$$
$$.3742 - .3777 \Rightarrow .0035 \quad \frac{1}{.0035} = 285.714 \text{ Hz}$$

### Frequency Analysis: Step 5

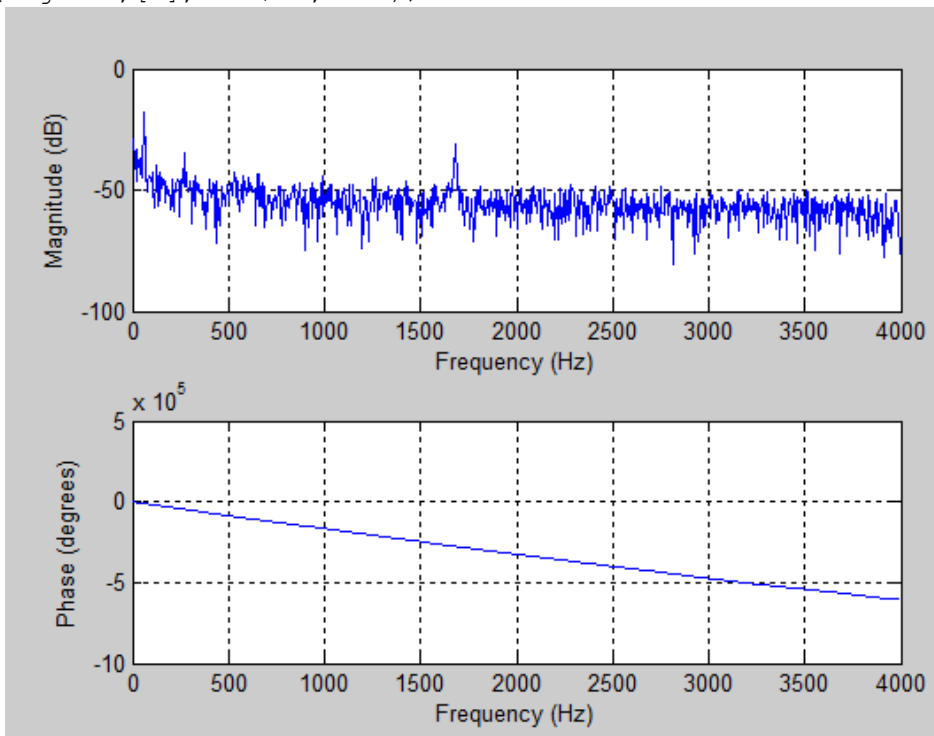
The magnitude spectrum of the signals in step 1 can be seen by using the freqz function. As you can see for the first frequency, the magnitude peaks at 480 Hz, which is exactly the frequency of 480 Hz tuning fork used. The amplitude in decibels of the signal in each case can be seen below. The amplitude in the first case is 18.07 dB, and the second signal has a -19.89 dB amplitude. The tuning fork frequency estimated by the frequency spectrum with the one estimated using the time signal are very close considering one is 480 vs. 500 Hz, respectively. In addition, the second signal's frequency can be seen as 272 vs. 277.78 Hz. Therefore, I would

consider the frequency estimated by the frequency spectrum a little more accurate than with the one estimated using the time signal.

```
freqz(signal1,[1],8000/10,8000);
```



```
freqz(signal2,[1],8000/10,8000);
```



**Frequency Analysis: Step 6**

Many of the other signals present in the spectrum come from noise for the most part. This noise is random fluctuation in the electrical signal. In this case, noise is something that is part of an error or considered an undesired random disturbance. You wouldn't see a nice clean delta function like in the notes because of this noise. The dirac delta function strictly focuses on calculations. However, in practice, all electrical signals have some sort of noise that the delta function has no way of including.