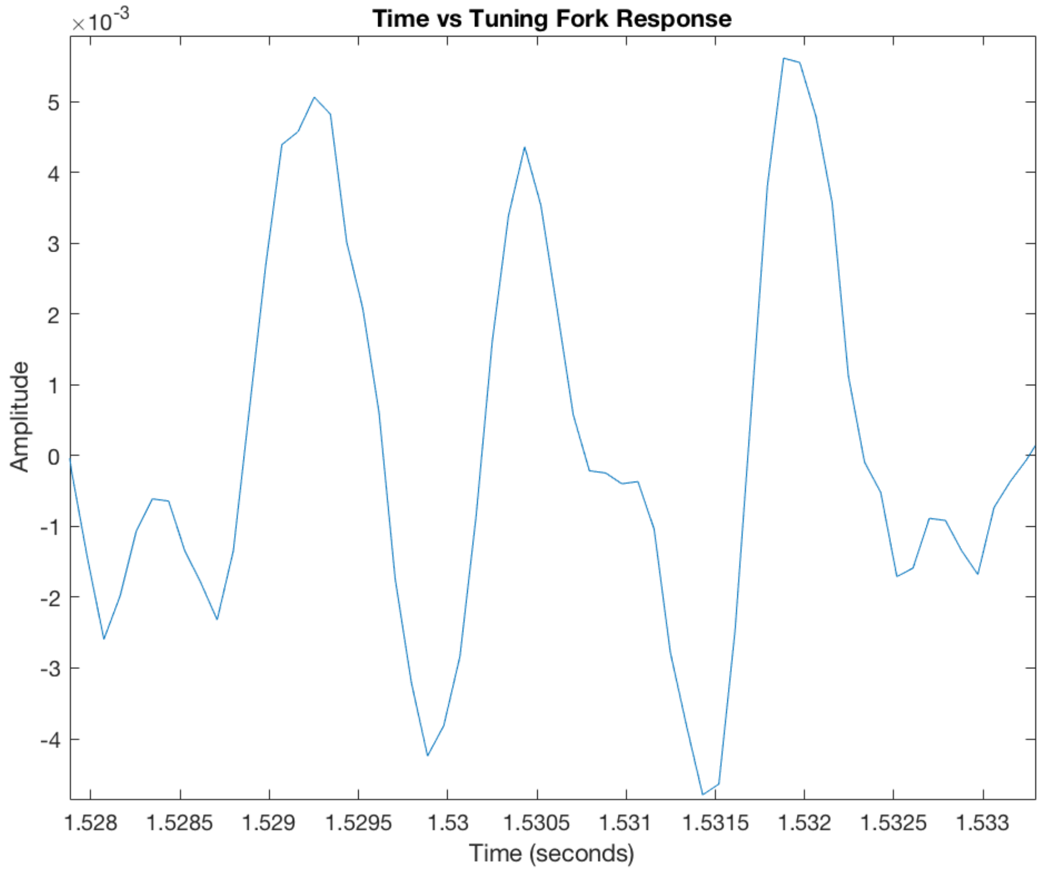
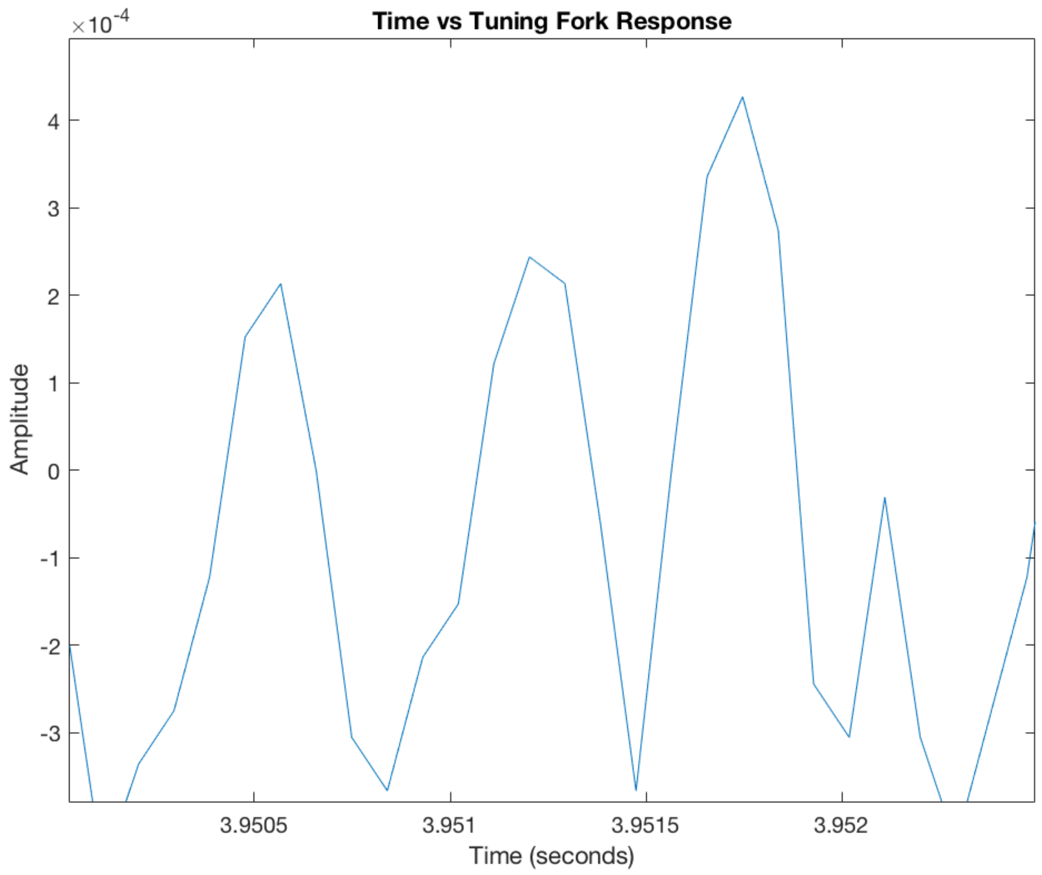
Introduction: In this lab, we used tuning fork to generate a sinusoidal signal. We then used microphone connected to a computer to record those signals, as a sinusoidal curve. It's characteristics such as period T and frequency can be related by curve as, f=1/T.

4.A





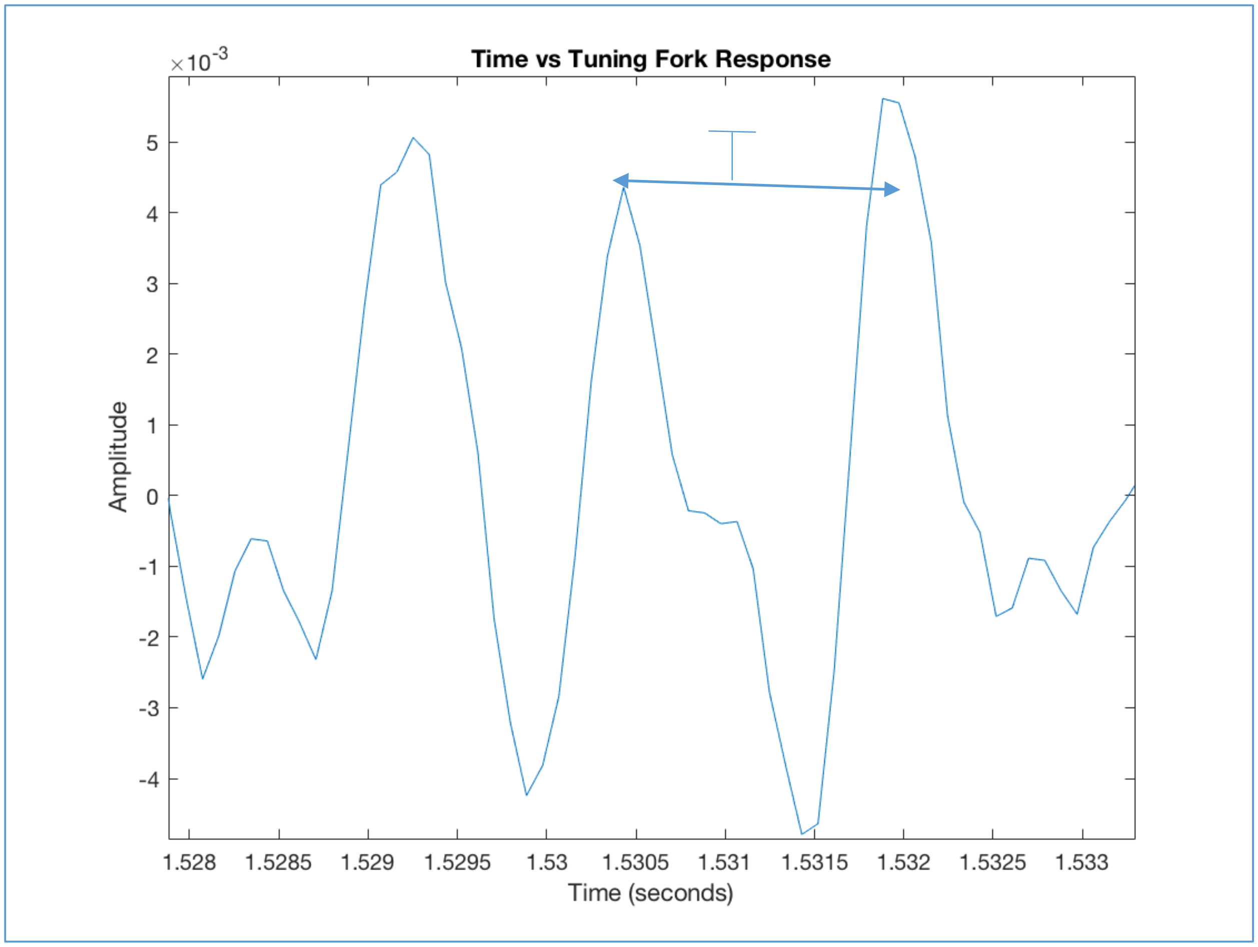
4.B T_1 = 1.532-1.5305 = 0.0015 seconds
f_1 = 1/T_1 = 1/0.0015 = 666.67 Hz

T_2 = 3.9513-3.9505 = 0.0008 seconds
f_2 = 1/T_2 = 1/0.0008 = 1250 Hz

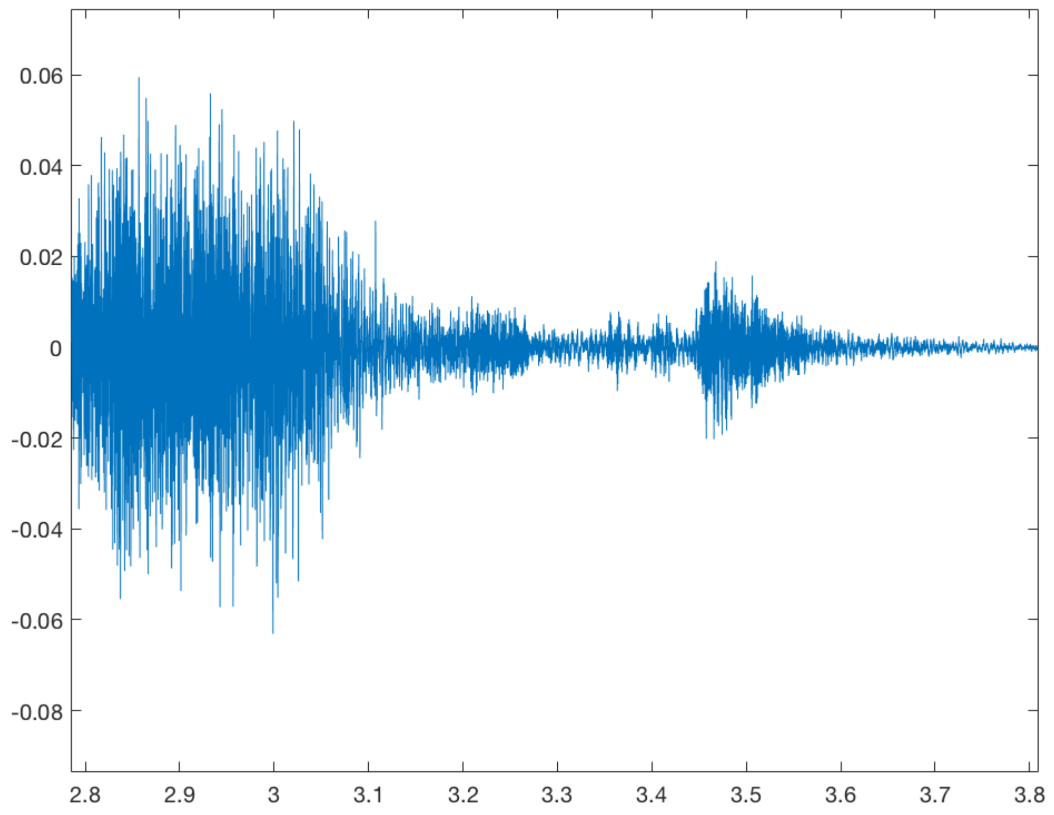
4.C The common possible source of error could be the background noise in the lab. The other possibility is placing of tuning fork too far, or too close to the microphone.

4.D The energy in the signal falls short off with respect to time, with middle accounting for the energy vibration/ longer peaks.

For the time period, the measuring of the signal took from peak to peak.



4.E Using sampling frequency of 44,100, our frequency estimated are little more accurate, due to the Nyquist theorem which states that "the maximum frequency that can be represented at any given sampling rate is half the sampling rate".



4.F A_1 = 5*10^-3
A_2 = 4*10^-4

4.G The other signals present in the spectrum are due to the distortions. I don't see a nice clean delta function like in the notes due to the distortions, and presence of background noise, while capturing the signals through microphone.

5.A

Signal Source	Frequency estimate from Time plot	Magnitude Decay Rate V/sec	Frequency estimate from frequency plot
Tuning fork 1 at 11,025 samp/sec	666.67 Hz	0.05	661
Tuning fork 2 at 11,025 samp/sec	1250 Hz	0.04	1246

5. B function [z] = compmag5(x,y);
%%%%%%%%%%
% this function computes the magnitude of the complex number x+jy and returns it in variable z.

```
% Inputs:
% x ? real part of complex number
% y ? imaginary part of complex number
%Outputs:
%z ? magnitude of complex number times 5
%written by J.A. Dickerson, January, 2005
a=5;
x=3;
y=4;
% compute magnitude and multiply
z =a* sqrt(x^2+y^2);
end
```

<u>For tuning fork 1:</u>

For sampling rate at 11,025 samp/sec;

```
Fs = 11025;
nbits = 16;
device = 1;
recObj = audiorecorder(Fs,nbits,device);

disp('Start Recording');
recordblocking(recObj, 5);
disp('End of Recording.');

play(recObj);

tuningfork1 = getaudiodata(recObj);

T = (0:length(tuningfork1)-1)* (1/Fs);
plot(T,tuningfork1);

xlabel('Time (seconds)');
ylabel('Amplitude');
title('Time vs. Tuning Fork Response');
```

For sampling rate at 44,100 samp/sec;

```
Fs = 44100;
nbits = 16;
device = 1;
recObj = audiorecorder(Fs,nbits,device);

disp('Start Recording');
recordblocking(recObj, 5);
```

```
disp('End of Recording.');

play(recObj);

tuningfork1 = getaudiodata(recObj);

T = (0:length(tuningfork1)-1)* (1/Fs);
plot(T,tuningfork1);

xlabel('Time (seconds)');
ylabel('Amplitude');
title('Time vs. Tuning Fork Response');
```

For frequency plot:

```
freqz(tuningfork1,[1],Fs);
```

For tuning fork 2

For sampling rate at 11,025 samp/sec;

```
Fs = 11025;
nbits = 16;
device = 1;
recObj = audiorecorder(Fs,nbits,device);

disp('Start Recording');
recordblocking(recObj, 5);
disp('End of Recording.');

play(recObj);

tuningfork2 = getaudiodata(recObj);

T = (0:length(tuningfork2)-1)* (1/Fs);
plot(T,tuningfork2);

xlabel('Time (seconds)');
ylabel('Amplitude');
title('Time vs. Tuning Fork Response');
```

For sampling rate at 44,100 samp/sec;

```
Fs = 44100;
nbits = 16;
```

```
device = 1;
recObj = audiorecorder(Fs,nbits,device);

disp('Start Recording');
recordblocking(recObj, 5);
disp('End of Recording.');

play(recObj);

tuningfork2 = getaudiodata(recObj);

T = (0:length(tuningfork2)-1)* (1/Fs);
plot(T,tuningfork2);

xlabel('Time (seconds)');
ylabel('Amplitude');
title('Time vs. Tuning Fork Response');
```

For frequency plot:

```
freqz(tuningfork2,[1],Fs);
```