PHYS 391 – Lab 5: Fourier Transform

Key Concepts

- Fast Fourier Transforms
- Frequency Resolution of FFTs
- Aliasing
- Harmonic Overtones

5.1 Introduction

This lab will explore the experimental aspects of using the Fast Fourier Transform (FFT). This is going to be a much more qualitative lab than the past few labs in the sense that the idea here is to let you explore some of these concepts from videos and collected data files rather than just relying on a mathematical descriptions. There really isn't much data to collect in this lab, so you will be graded mostly on how well you can explain your answers to posed questions and assigned tasks **in boldface** and how you can relate this to the theory of the Fourier Transform.

With the exception of some coding tasks, the majority of work for this lab entails writing and drawing/labeling figures. Please find on our Canvas site a **lab response template** (named: *Lab5template.docx*) Plan to download this word document and use it to answer questions posed in this lab. You're welcome to neatly draw graphs and figures that support your answers on the document (allow space), then scan this and hand in the scan. This file is in addition to any coding work that you can turn in separately.

5.2 Vernier Setup

We will use Vernier Logger Pro connected to a microphone and voltage probe via a LabPro interface (or similar) to detect time-varying voltages supplied to a speaker by a Pasco Function Generator. Both data channels to Logger Pro (the microphone input and voltage from the probe) will be displayed as both time domain graphs and Fourier transform plots.

Recall that all observed data is discrete, and that *time series* are time-domain data sampled at regular time steps, Δt. This time-domain data is also sometimes called a "waveform". Recall, also, that the *sampling frequency* (ν_s) is related to Δt by the expression:

$$\nu_s = \frac{1}{\Delta t}$$

The output from the function generator has been wired to drive a basic speaker over which audio signals can be heard. The Vernier microphone probe senses these signals. The voltage probe is also connected so as to monitor, directly, the voltages produced by the function generator.

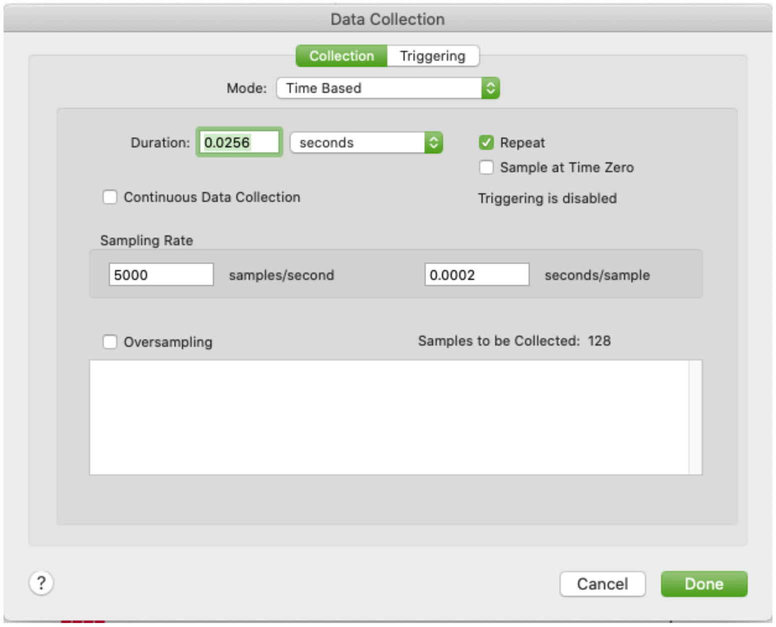
5.3 Frequency Domain Setup

We'll set up the LoggerPro by setting up the critical parameters controlling both the resolution of the signal in the time domain: ν_s and N. These are the sampling frequency and number of samples analyzed (and acquired in this case) per *window.* As LoggerPro doesn't let us control directly the value for N, we need to use the LoggerPro *Duration* setting in its *Data Collection* dialog to set this.

a) Question: for Figure 1, how can we derive the N value shown in the Data Collection dialog box in Figure 1

Using these acquisition parameters, we will first look at an input microphone signal driven at 500 Hz. The time-domain waveform is shown in Figure 2.

b) Verify from Figure 2 that the frequency is indeed 500 Hz. Measure the period from the graph and show your work to find the frequency (i.e. what did you measure to determine



Data Collection

Mode: Time Based

Duration: 0.0256 seconds

☑ Repeat

□ Sample at Time Zero

□ Continuous Data Collection

Triggering is disabled

Sampling Rate

5000 samples/second

0.0002 seconds/sample

□ Oversampling

Samples to be Collected: 128

Figure 1: An example *Data Collection* dialog for LoggerPro

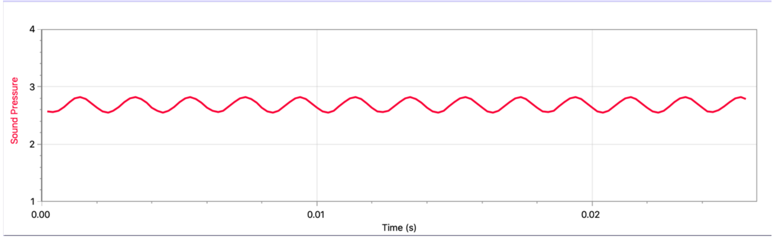


Figure 2: Time-domain data acquired with 500 Hz sound wave

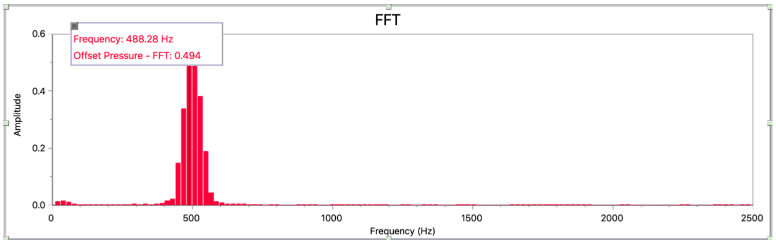


Figure 3: Frequency-domain data acquired with 500 Hz sound wave

this?)

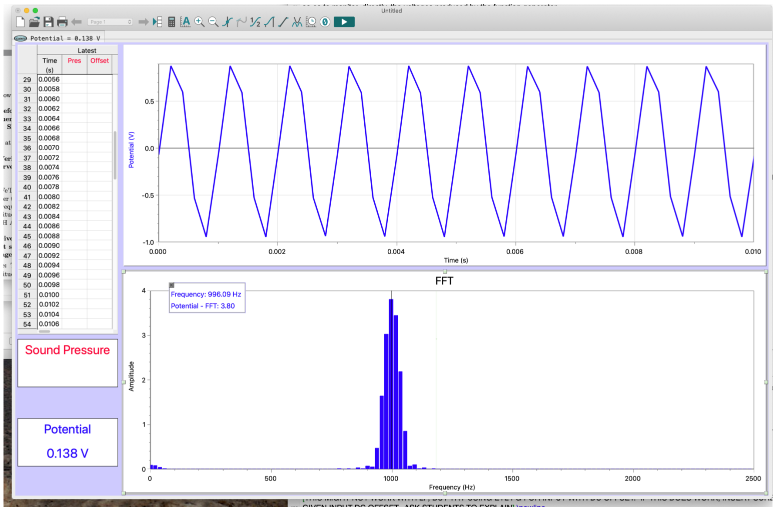
Now we'll look at the FFT (frequency domain) in addition to the time domain plot.

c) Before we do, given the sampling frequency ν_s and N value above, work out the expected frequency step between successive *Fourier coefficients* that will be viewable in the frequency-domain plot. Show your work.

Watch the short video **video5p3**. The time-domain signal in the video is shown in Figure 2, while the frequency-domain signal is shown in Figure 3.

d) Verify that your predicted frequency step $\Delta\nu$ between successive Fourier coefficients is observed. If it was different, explain why.

We'll now look at the input signal from the function generator directly with a Vernier voltage probe (rather than via an audio signal captured by the microphone). Figure 4 shows a screenshot of this signal for the frequency domain (FFT). We have used LoggerPro's (LP's) Analysis/Examine function to determine the amplitude of the peak (largest) Fourier coefficient.



Potential = 0.138 V

	Latest		
	Time (s)	Pres	Offset
29	0.0056		
30	0.0058		
31	0.0060		
32	0.0062		
33	0.0064		
34	0.0066		
35	0.0068		
36	0.0070		
37	0.0072		
38	0.0074		
39	0.0076		
40	0.0078		
41	0.0080		
42	0.0082		
43	0.0084		
44	0.0086		
45	0.0088		
46	0.0090		
47	0.0092		
48	0.0094		
49	0.0096		
50	0.0098		
51	0.0100		
52	0.0102		
53	0.0104		
54	0.0106		

Sound Pressure

Potential
0.138 V

Frequency: 996.09 Hz
Potential - FFT: 3.80

Figure 4: Data acquired with 1 kHz, 1V peak input signal

e) Given the amplitude for this peak Fourier coefficient, how does it relate to the 1V peak input signal from the function generator. Is the amplitude of the peak the same as the peak voltage, or something else. Explain.

Note: The RMS amplitude of a sine wave is defined as $V_{\mathrm{RMS}} = \sqrt{\overline{V(t)^2}} = V_0/\sqrt{2}$, where V_0 is the peak amplitude $V(t) = V_0 \sin \omega t$. In electricity, this is also called the AC voltage.

Note 2: This exercise points out a bug in Vernier's sofware setup.

f) Describe this bug. Cite the upper time-domain and lower frequency-domain plots as evidence.

g) What is the maximum frequency displayed on the plot? Why doesn't this go up to ν_s?

5.4 Frequency Fidelity

5.4.1 Frequency Leakage

Look at the two figures 5 and 6. They show data for two successive Fourier coefficients in the analysis boxes. The Vernier setup is as before, with a sampling frequency ν_s of 5000 Hz and $N = 128$. The input signal is exactly 500 Hz at 1V peak.

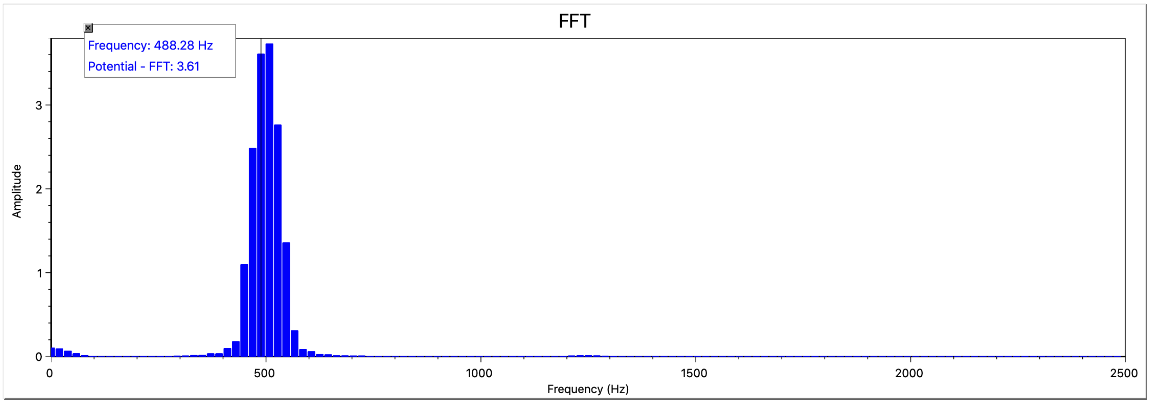


Figure 5: FFT graph for 500 Hz, 1V peak amplitude sine wave, one Fourier coefficient analyzed

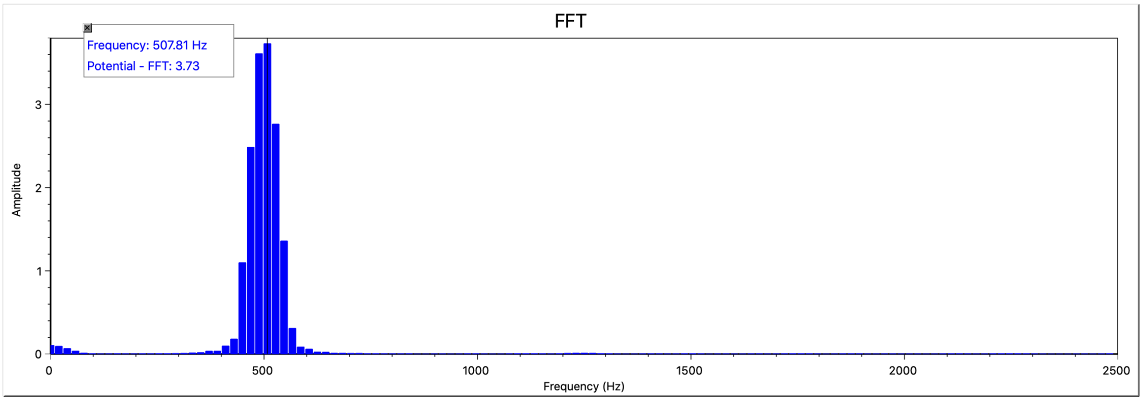


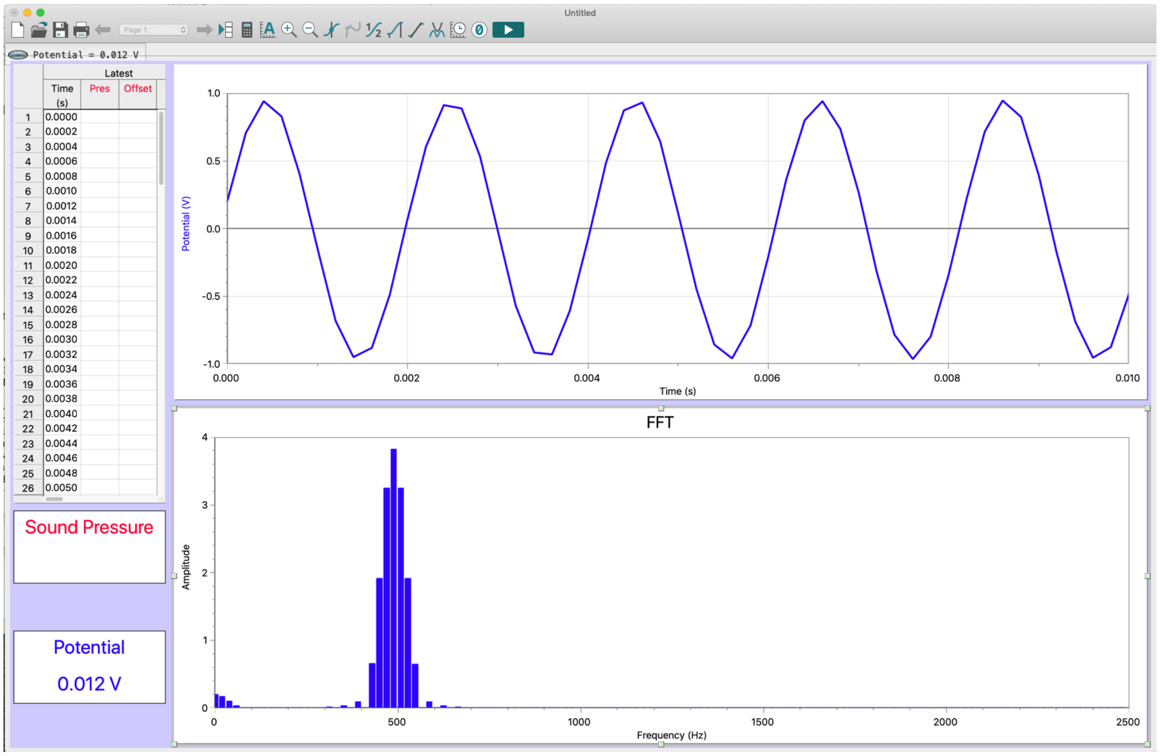
Figure 6: FFT graph for 500 Hz, 1V peak amplitude sine wave, the next Fourier coefficient analyzed

a) Describe in words what you see in the FFT plot. Is there a lone peak at 500 Hz? In other words, is all the amplitude in a single bin? If so, why, if not, why not? Explain.

b) To what frequency should we adjust the function generator in order to get one single bar (one appreciable Fourier coefficient) if we were using a boxcar window. Show your work supporting your answer.

c) Figure 7 shows the result of adjusting the input frequency to correspond to one of the Fourier coefficients. Describe what you see in the FFT (frequency domain) plot. Specifically, does this look like you would expect from your answer in b)?

d) Speculate as to why the Fourier coefficient bars adjacent to the peak aren't zero in Figure 7. Hint: LoggerPro is intended to take general-purpose frequency data. What might you want to include in a setup like this?



Untitled

Potential = 0.012 V

	Latest		
	Time (s)	Pres	Offset
1	0.0000		
2	0.0002		
3	0.0004		
4	0.0006		
5	0.0008		
6	0.0010		
7	0.0012		
8	0.0014		
9	0.0016		
10	0.0018		
11	0.0020		
12	0.0022		
13	0.0024		
14	0.0026		
15	0.0028		
16	0.0030		
17	0.0032		
18	0.0034		
19	0.0036		
20	0.0038		
21	0.0040		
22	0.0042		
23	0.0044		
24	0.0046		
25	0.0048		
26	0.0050		

Sound Pressure

Potential

0.012 V

Figure 7: FFT graph for a 1V peak amplitude sine wave after the frequency has been adjusted to match one of the Fourier components

Note that when the input signal's frequency doesn't match the frequency of one of the Fourier coefficients (given choices of ν_s and N), it becomes more challenging to estimate the true amplitude and frequency of the input signal from the Fourier coefficient amplitudes.

e) Qualitatively describe a method that would allow you to find the best value of frequency for a peak in the frequency-domain data.

5.4.2 Frequency Resolution

The frequency resolution (how well the FFT resolves small changes in input frequency) depends of course on the sample frequency, ν_s and the number of samples analyzed N.

a) Calculate how you would expect the frequency resolution ($\Delta\nu$) to change if you were to double the sampling frequency ν_s or double the sample size N. Show your work.

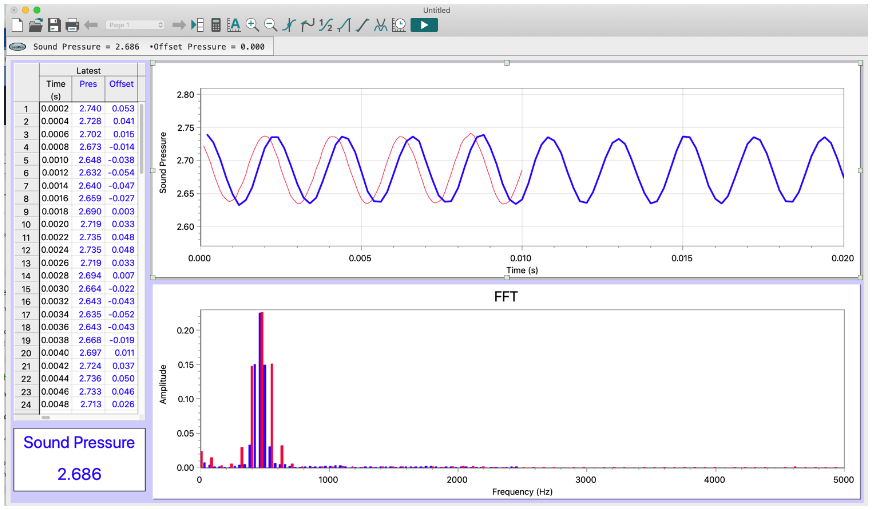
To verify your calculations, we first try halving ν_s (to 5 kHz) while keeping N fixed (so doubling *Duration*) and explore what happens to the frequency peak. Note that the input signal is 468.8 Hz at about 1V peak, and that we are using our 'audio' setup. Figure 8 gives before and after plots.

b) Describe, qualitatively, what happens to the frequency resolution, $\Delta\nu$ when the sample rate (ν_s) is halved. State a mathematical rationale for this. Note that we say the *resolution* is higher when $\Delta\nu$ is smaller. Is the resolution higher or lower when the sampling frequency is halved? Explain.

Examine, carefully, the shape of the time domain plots for both cases (10 kHz and 5 kHz sampling).

c) Describe and explain differences between these time domain plots.

Now we will keep the sampling frequency (ν_s) fixed, and instead, adjust N.



Sound Pressure = 2.686 •Offset Pressure = 0.000

	Latest		
	Time (s)	Pres	Offset
1	0.0002	2.740	0.053
2	0.0004	2.728	0.041
3	0.0006	2.702	0.015
4	0.0008	2.673	-0.014
5	0.0010	2.648	-0.038
6	0.0012	2.632	-0.054
7	0.0014	2.640	-0.047
8	0.0016	2.659	-0.027
9	0.0018	2.690	0.003
10	0.0020	2.719	0.033
11	0.0022	2.735	0.048
12	0.0024	2.735	0.048
13	0.0026	2.719	0.033
14	0.0028	2.694	0.007
15	0.0030	2.664	-0.022
16	0.0032	2.643	-0.043
17	0.0034	2.635	-0.052
18	0.0036	2.643	-0.043
19	0.0038	2.668	-0.019
20	0.0040	2.697	0.011
21	0.0042	2.724	0.037
22	0.0044	2.736	0.050
23	0.0046	2.733	0.046
24	0.0048	2.713	0.026

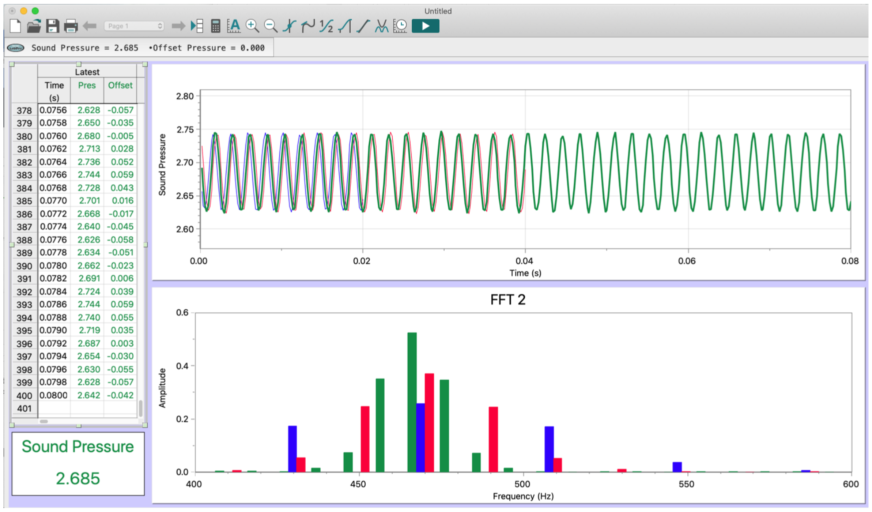
Sound Pressure

2.686

Figure 8: The same signal sampled at 10kHz (red) and 5kHz (blue) with N=100 in both cases

d) Given your "mathematical rationale" from above, what will happen when you double N with constant ν_s? Explain.

To verify your calculations, we've set ν_s to 5 kHz and started with N=100. Saving these plots, we then doubled N, and then doubled N again,. while keeping ν_s fixed. Note that the input signal is 468.8 Hz at about 1V peak, and that we are using our 'audio' setup. Figure 9 gives before and after plots.



Sound Pressure = 2.685 •Offset Pressure = 0.000

	Latest		
	Time (s)	Pres	Offset
378	0.0756	2.628	-0.057
379	0.0758	2.650	-0.035
380	0.0760	2.680	-0.005
381	0.0762	2.713	0.028
382	0.0764	2.736	0.052
383	0.0766	2.744	0.059
384	0.0768	2.728	0.043
385	0.0770	2.701	0.016
386	0.0772	2.668	-0.017
387	0.0774	2.640	-0.045
388	0.0776	2.626	-0.058
389	0.0778	2.634	-0.051
390	0.0780	2.662	-0.023
391	0.0782	2.691	0.006
392	0.0784	2.724	0.039
393	0.0786	2.744	0.059
394	0.0788	2.740	0.055
395	0.0790	2.719	0.035
396	0.0792	2.687	0.003
397	0.0794	2.654	-0.030
398	0.0796	2.630	-0.055
399	0.0798	2.628	-0.057
400	0.0800	2.642	-0.042
401			

Sound Pressure

2.685

Figure 9: The same signal sampled at 5 kHz for N's of 100 (blue), 200 (red) and 400 (green)

e) Did doubling N while keeping the sampling frequency fixed have the effect you predicted? Explain. What happens to the *frequency resolution* as N is increased. Why?

f) Can you think of a case where increasing N alone without increasing ν_s wouldn't actually lead to any improvement? What would happen if you had a transient, non-periodic signal like a short single pulse? What would the effective "sample time" be in this case?

5.5 Aliasing

Aliasing is an interesting phenomenon, for example, causing wagon wheels to appear to move counterclockwise in old westerns. We will explore aliasing by watching a pre-made video. Look at the short video **video5p5**. Write down the values for input (function generator) frequency and amplitude and those given for particular Fourier coefficients that are pointed out as the video progresses. as the video progresses. The video begins with the sampling frequency set to 1 kHz and $N = 100$.

Questions:
a) As we scanned the input frequency of the function generator up to the Nyquist frequency for this setup, what did you see? Particularly, what did you see when the input frequency was set to 200 Hz?

b) What happened as the input frequency approached the Nyquist limit?

c) Describe what happened in the frequency plot (FFT) after the input frequency surpassed the Nyquist limit and approached the sampling frequency. I asked a question about the apparent 400 Hz signal observed after the frequency surpassed the Nyquist limit in the video, answer it here.

d) Describe what happened after the input frequency exceeded the sampling frequency at 1 kHz.

e) Finally, does a pattern become apparent? If the input frequency is 750 Hz, what would the expected observed (and aliased) frequency be? Explain.

5.6 Frequency Overtones

Recall that when using *Fourier synthesis* in your homework (5) you could construct standard waveforms, for example triangle and square waves, by adding *overtones*, defined as integer multiples of some fundamental frequency, together. The trick was in determining whether *odd*, *even* or *both* integers (e.g. 3, 5, 7 times the fundamental) were needed. Another important piece of information is how the amplitudes of the *overtones* compared to that of the $fundamental$ (the latter being the frequency input by the function generator). **Aside: for an excellent simulator for the construction of waveforms using Fourier series, see this excellent web site, particularly: http://www.falstad.com/fourier/**

To explore frequency overtones, we'll set the sampling frequency (ν_s) to 1 kHz and $N = 1000$. We begin by setting input frequency at 200 Hz with 1V-P amplitude. Look at the short video **video5p6audio**. Here we explore the differences in observed waveforms given stated changes in the input signal between *sine*, *triangle*, and *square* waves, initially using our 'audio' setup.

Questions:
a) How faithful is the representation of the sine wave given our (audio: speaker + microphone) system for monitoring the input signal? Cite evidence observed in the video.

b) Does the triangle wave look like what you expected? Explain.

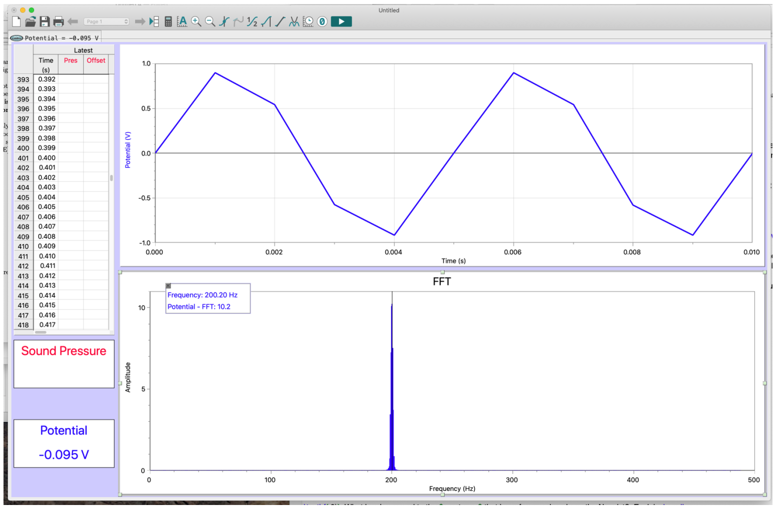
c) The square wave observed with our system does not look much like a square wave. Speculate as to why that might be.

Look at another short video **video5p6voltage**. Here we do the same exploration of waveforms, but using a voltage probe to monitor input signals.

d) is the observed waveform of the input square wave improved using the new setup, a voltage probe? Explain why this might be so.

Now to analyze these signals. We start with the same parameters as above (for the videos). In Figure 10 we show both time- and frequency-domain plots for a 200 Hz sine wave (amplitude $\sim$ 1 V peak) using the

Voltage Probe. In Figures 11-12 we show the same type of plot for a triangle wave with Fourier peaks picked at 200 Hz and 400 Hz. In Figures 13-15 we do the same for the square wave, but pick also the peak Fourier coefficient near 0 Hz as well.


Potential = -0.095 V

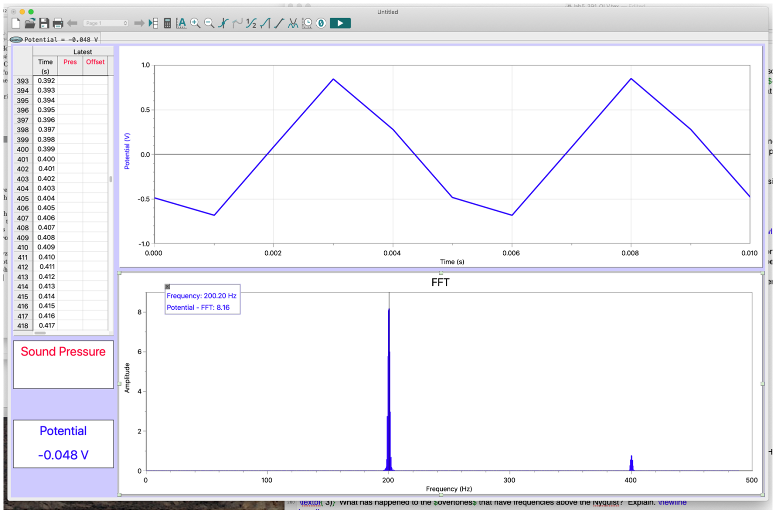
	Latest		
	Time (s)	Pres	Offset
393	0.392		
394	0.393		
395	0.394		
396	0.395		
397	0.396		
398	0.397		
399	0.398		
400	0.399		
401	0.400		
402	0.401		
403	0.402		
404	0.403		
405	0.404		
406	0.405		
407	0.406		
408	0.407		
409	0.408		
410	0.409		
411	0.410		
412	0.411		
413	0.412		
414	0.413		
415	0.414		
416	0.415		
417	0.416		
418	0.417		

Sound Pressure

Potential -0.095 V

Frequency: 200.20 Hz
Potential - FFT: 10.2

Figure 10: 200 Hz, 1V-P sine wave signal analyzed near 200 Hz.


Potential = -0.048 V

	Latest		
	Time (s)	Pres	Offset
393	0.392		
394	0.393		
395	0.394		
396	0.395		
397	0.396		
398	0.397		
399	0.398		
400	0.399		
401	0.400		
402	0.401		
403	0.402		
404	0.403		
405	0.404		
406	0.405		
407	0.406		
408	0.407		
409	0.408		
410	0.409		
411	0.410		
412	0.411		
413	0.412		
414	0.413		
415	0.414		
416	0.415		
417	0.416		
418	0.417		

Sound Pressure

Potential -0.048 V

Frequency: 200.20 Hz
Potential - FFT: 8.16

Figure 11: 200 Hz, 1V-P triangle wave signal analyzed near 200 Hz.

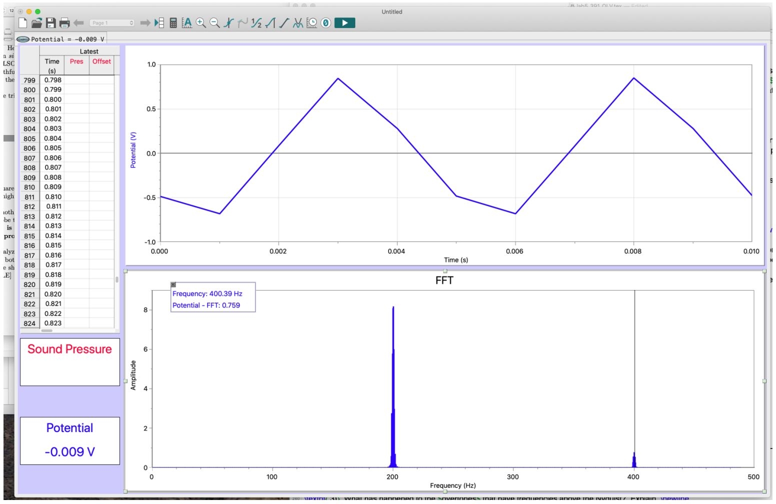
More Questions:

e) What are the expected overtone frequencies for the triangle wave (compared to that of the *fundamental* frequency of 200 Hz)? Find a mathematical expression for these and state it here (you can look this up).

f) What about the square wave? Note, you calculated this in HW5.

g) What has happened to the *overtones* that have frequencies above the Nyquist limit? Explain.

h) Does it seem like a good idea to have the Nyquist frequency be an integer multiple of your primary frequency?



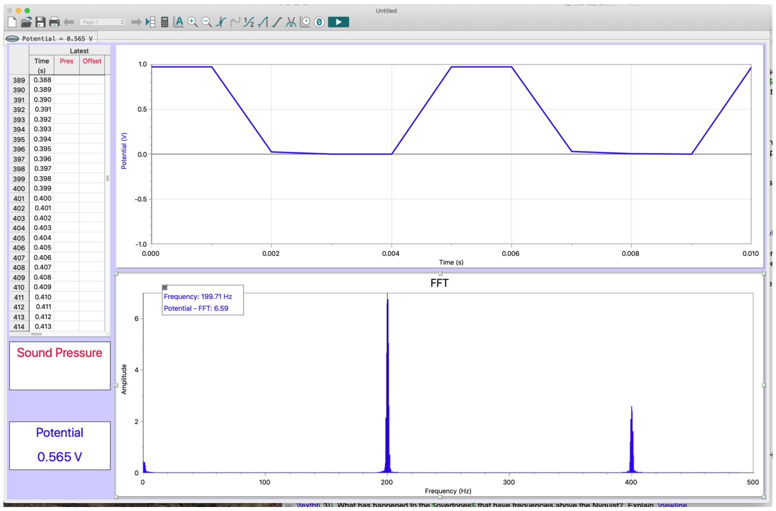
Potential = -0.009 V

	Latest		
	Time (s)	Pres	Offset
799	0.798		
800	0.799		
801	0.800		
802	0.801		
803	0.802		
804	0.803		
805	0.804		
806	0.805		
807	0.806		
808	0.807		
809	0.808		
810	0.809		
811	0.810		
812	0.811		
813	0.812		
814	0.813		
815	0.814		
816	0.815		
817	0.816		
818	0.817		
819	0.818		
820	0.819		
821	0.820		
822	0.821		
823	0.822		
824	0.823		

Sound Pressure

Potential
-0.009 V

Figure 12: 200 Hz, 1V-P triangle wave signal analyzed near 400 Hz.



Potential = 0.565 V

	Latest		
	Time (s)	Pres	Offset
389	0.388		
390	0.389		
391	0.390		
392	0.391		
393	0.392		
394	0.393		
395	0.394		
396	0.395		
397	0.396		
398	0.397		
399	0.398		
400	0.399		
401	0.400		
402	0.401		
403	0.402		
404	0.403		
405	0.404		
406	0.405		
407	0.406		
408	0.407		
409	0.408		
410	0.409		
411	0.410		
412	0.411		
413	0.412		
414	0.413		

Sound Pressure

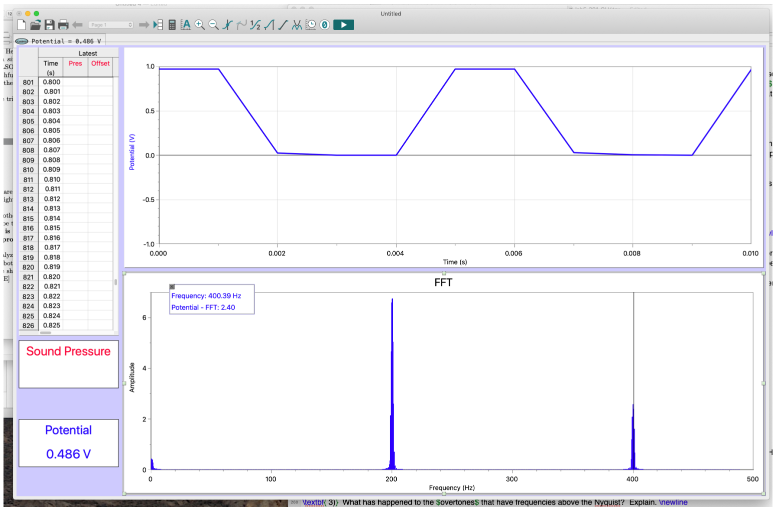
Potential
0.565 V

Figure 13: 200 Hz, 1V-P square wave signal analyzed near 200 Hz.

i) Explain why experimentalists design equipment to include a lo-pass filter between the signal detector and the Analog-to-Digital-Converter (ADC) to directly filter out any frequency above our Nyquist limit.

5.7 Coding to Analyze Musical (or other) Signals

Musical signals (time series of notes played) can be analyzed to observe their own *overtone sequences*. These, in turn, give information about the nature of the resonances set up inside various musical instruments so as to produce these notes. For example, the overtone sequences of notes played on string instruments are related to the fundamental frequency by both odd- and even-integer multiples. This reflects the fact that string instruments have displacement nodes at both ends of their strings that, in turn, dictates the types of resonance *modes* (set sequences of the fundamental and overtones) when resonance is excited on a string. Many string instruments are played by *plucking* the strings (e.g., guitars, pianos, *pizzacato* for orchestral string instruments). However, the overtone sequence of said, struck string (say that fast!), isn't constant after the string is struck.



Potential = 0.486 V

	Latest		
	Time (s)	Pres	Offset
801	0.800		
802	0.801		
803	0.802		
804	0.803		
805	0.804		
806	0.805		
807	0.806		
808	0.807		
809	0.808		
810	0.809		
811	0.810		
812	0.811		
813	0.812		
814	0.813		
815	0.814		
816	0.815		
817	0.816		
818	0.817		
819	0.818		
820	0.819		
821	0.820		
822	0.821		
823	0.822		
824	0.823		
825	0.824		
826	0.825		

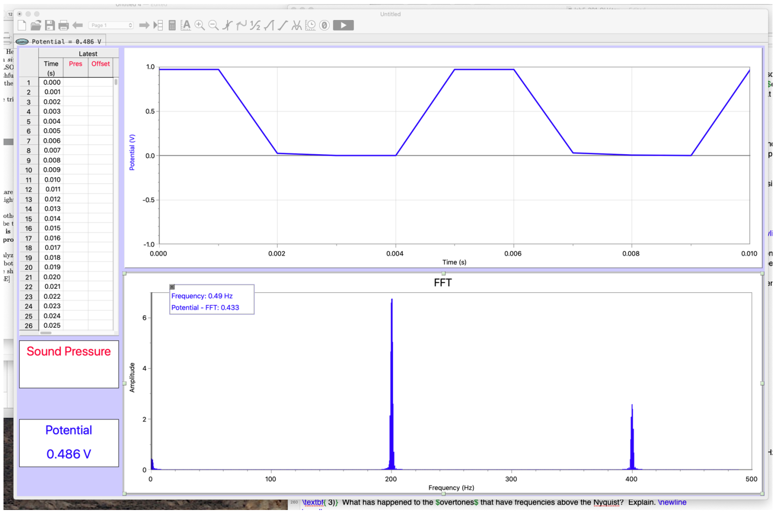
Sound Pressure

Potential
0.486 V

Frequency: 400.39 Hz
Potential - FFT: 2.40

260 \textbf{ 3)} What has happened to the $overtones$ that have frequencies above the Nyquist? Explain. \newline

Figure 14: 200 Hz, 1V-P square wave signal analyzed near 400 Hz.



Potential = 0.486 V

	Latest		
	Time (s)	Pres	Offset
1	0.000		
2	0.001		
3	0.002		
4	0.003		
5	0.004		
6	0.005		
7	0.006		
8	0.007		
9	0.008		
10	0.009		
11	0.010		
12	0.011		
13	0.012		
14	0.013		
15	0.014		
16	0.015		
17	0.016		
18	0.017		
19	0.018		
20	0.019		
21	0.020		
22	0.021		
23	0.022		
24	0.023		
25	0.024		
26	0.025		

Sound Pressure

Potential
0.486 V

Frequency: 0.49 Hz
Potential - FFT: 0.433

\textbf{ 3)} What has happened to the $overtones$ that have frequencies above the Nyquist? Explain. \newline

Figure 15: 200 Hz, 1V-P square wave signal analyzed near 0 Hz.

5.7.1 Waveform Analysis

On the course website there are several files with recordings of musical instruments along with a Jupyter notebook showing how to read the data from a WAV file into python. For your coding exercise, start with this notebook and add code to do the following:

- Read in the **GuitarA.wav** file and plot the amplitude vs. time (time domain) for a range of times that lets you see a few cycles of the waveform. This part is mostly done for you.
- Take the Fourier transform of this waveform and display the results as magnitude vs. frequency (frequency domain). You wrote code to do this in HW5, so you can just re-use that code. Adjust the x-axis scale so that you can better see the first few harmonics. I find a range up to 1 kHz works well.
- While these samples are long enough that windowing isn't a huge issue, you can apply a window to your data to make the peaks slightly more regular if you wish.
- Identify the fundamental frequency and the first few harmonics. Also, find the ratio of amplitude in each harmonic to the fundamental frequency. This is approximate, and you can just read the values off your plot.
- What features do you note about the spacing of the frequencies and their amplitudes? From the

harmonic components, do these look like square waves, triangle waves, or something more like what you would expect from standing waves?

Repeat this exercise for at least one more waveform. The piano and saxophone filess are both playing the same A note, so these are a direct comparison of same pitch, but with rather different toness.

5.7.2 Bonus Sonogram

This section is optional, but you can earn a few extra credit points for completing this section. For non-static waveforms, a common analysis method is to make a *waterfall plot* or *sonogram*. This shows the Fourier-transformed frequency components vs. time where the FFT is repeatedly applied to a short interval of data at a time. The LIGO waveform shown in class is one example of this sort of plot.

The challenge, then is to produce your own sonogram. The general idea is the following:

- Import a WAV file into python
- Construct a loop that:
  - chops the sound signal time series into *windows* of some length (e.g., $N = 512$ or 1024 points)
  - makes an FFT of this window of data (here you will likely need to apply a Hamming window to avoid the boxcar window artifacts)
  - saves the FFT magnitude vs. frequency output as one column in a 2D array
  - proceeds by '*sliding*' the window along the entire data set, continuing to add columns to the 2D array for each slice.
- Plot the array as a *waterfall* or *sonogram* plot of frequency vs. time, where the color indicates the magnitude.
- Adjust the window length and y-axis scaling so that you have enough frequency resolution to see the main harmonic peaks.

An alternate to writing your own loop is to use the matplotlib `specgram` function. This avoids having to write your own code, but you will have to figure out for yourself how to call this function and adjust the window parameters.

If you get this sonogram working, try looking at the guitar string and the didgeridoo recording.

What can you say about how the overtone content of the struck, guitar A string evolves over time (after it is struck)?

What to you notice about the frequencies in the didgeridoo? How is this different from the guitar?

5.8 Final Comments

This lab has only scratched the surface in demonstrating some of the useful things that can be done with the FFT experimentally, but it has illustrated several of the pitfalls which can easily be avoided. The issues of aliasing, leakage, and windowing are always present in any experimental frequency-domain measurement. Without an understanding of where these issues arise from and bit of care to mitigate their effects, one can be severely fooled by discrete sampled data. Hopefully this lab will save you an afternoon of grief some day when you have to do this for real.