ECE 300 Lab 6: Studying Aperiodic Signals

## Objective

The objective of this lab is to understand the difference between periodic and aperiodic signals. Some aperiodic signals, such as speech waveforms, actually consist of short bursts of periodic signals. In the case of speech, each letter’s sound is a periodic signal, but when you string all of the sounds together to make words and the words together to make sentences, the result is an aperiodic signal. Periodicity can depend upon the timescale at which one is observing the signal. We will be using the short-time Fourier Transform (STFT) to understand how the periodicity of a signal may change over time. The STFT is displayed in a special graph called a spectrogram.

* Observe the transition from a Fourier Series to Fourier Transform as a signal’s period increases to infinity
* Observe the periodic nature of various vocal sounds corresponding to letters
* Spectral characterization of aperiodic speech signals
* Observe and understand the differences between the Fourier Series, Fourier Transform, and the STFT

# Prelab:

Speech processing and speech recognition begin with understanding where speech sounds come from and what they “look” like.

1. Describe the difference between a voiced versus a voiceless sound. Provide two examples of phonetics that are voiced and two that are unvoiced, for example “f as in farm”.
2. Look up the definition of a spectrogram. Find an example of a birdsong spectrogram and describe in your own words what that birdsong would sound like by looking at the spectrogram.
3. Look up the standards for AMR-WB and AMR-NB for speech coding in telecommunication systems. Provide the bandwidth that is used for each of these codecs.

# Lab Experiments

## Part 1: From Fourier series to Fourier transform

We discussed in class that the relationship between the Fourier Series and Transform can be explained by making the period go to infinity for a periodic waveform. When this happens, the series no longer has discrete harmonics because the fundamental frequency is zero and the transform becomes a continuous function of ω. You will observe this phenomenon happen before your very eyes during this part of the lab.

1. Scope Ch1 = V\_OUT

FG Ch1 to SIG: Ramp, 500mVpp, 1kHz, offset=250mV

FG Ch2 to YIN: Pulse, 1Vpp, 500Hz, pulse-width=1ms, offset=500mV

Set Scope: MATH channel on, FFT, Ch1

Set Scope: Acquire -> High Res, 2ms/div

1. Set the board to have a gain of 1 (yes, even though a YIN signal is applied, you will start this part with a gain of 1). Set the other jumpers and switches to pass the signal through to the output unfiltered.
2. Look at the signal and its spectrum. Verify that this signal is consistent with Figure 1A below. Note where 0V is.
3. Set the gain switch over to YIN. Make sure that your scope looks like Figure 1B. Flip back and forth between a gain of 1 and the gain set to YIN. Make sure you understand the changes in the frequency domain.
4. Change the frequency of the YIN Pulse signal (make sure the FG is on the correct channel) from 500Hz to 250Hz to 125Hz to 62.5Hz. Make sure you get the scope images as seen in Figures 1B-D. Make sure you understand the changes in the spectrum for each of these different frequencies.
5. **Your Experiment:** Get a screen capture of the scope with the frequency domain at each of the 4 frequencies for the pulse signal from Part (e). Connect the speaker to the speaker driver output. Listen to the sound as you change the frequency of the Pulse signal on YIN through the 4 different frequencies mentioned previously. Insert your screen captures and answer the questions in the Data Memo.

|  |  |
| --- | --- |
| 1. Periodic Signal for x(t)   no_hz | 1. X(t) multiplied by pulse at 500Hz   500hz |
| 1. X(t) multiplied by pulse at 250Hz   250hz | 1. X(t) multiplied by pulse at 125Hz   125hz |

**Figure 1:** 4 of the 5 periodic signals that you will be observing in this part of the lab. Note how the triangle in each period remains a constant shape, but the triangles are just spaced further apart in time.

## Part2: Capturing speech waveforms

Speech is made up of continually varying sounds. While the individual vocal sounds are periodic in nature, when you string them together the resulting words are not periodic. You will begin by looking at the periodic nature and spectra of individual vocal sounds. Then you will look at the spectrum for an entire word.

1. Set the board to use the microphone input

Turn off the MATH function on the scope so that you only see the output signal  
Set the scope timescale to 100ms/div

Make sure that the scope is in “normal” mode of acquiring (***DO NOT*** use averaging or hi-res mode)

1. Say the “sss” sound (as in “SAVE”) into the microphone in your normal speaking voice. Try to keep the volume level and pitch of your voice as constant as possible (i.e. speak in a monotone voice). When you see a trace on the screen that doesn’t vary much in amplitude and appears to be fairly periodic, press the Run/Stop button on the scope to freeze the screen. Record the actual data values as an ASCII-XY text file (CSV format). Remember that the scope will only save the data that is currently displayed on the screen, so make sure you are not “zoomed” in when saving and that the waveform is not off of the edge of the screen. You should also make sure that only the channel that is displaying the output of the microphone is visible, and all other channels should be turned off. The scope will record all displayed channels, which makes it more difficult to get only the data that you want out of the file using MATLAB. **When you press the Save/Recall -> Save buttons, there is an option to change the Format of the data. Choose the ASCII-XY option. You should also change the number of points to save with Save/Recall -> Save -> Settings -> Length. Set the number of points to be 20000. Note that in order to change the Length setting the scope must be stopped (Run/Stop -> Stop (button is red instead of green)).**
2. Do the same as Part (b) for the long-a sound (as in “SAVE”).
3. Do the same as Part (b) for the “v” sound (as in “SAVE”).
4. Before doing the whole word, you need to change a number of settings on the scope.

Adjust the timescale to be 200ms/div

Trigger Mode/Coupling -> Mode -> Normal. This makes it so that the scope will not update the screen until the waveform crosses the trigger level.

Trigger Level -> Adjust the trigger level so that it is just above the trace, but will not trigger unless it receives a loud sound.

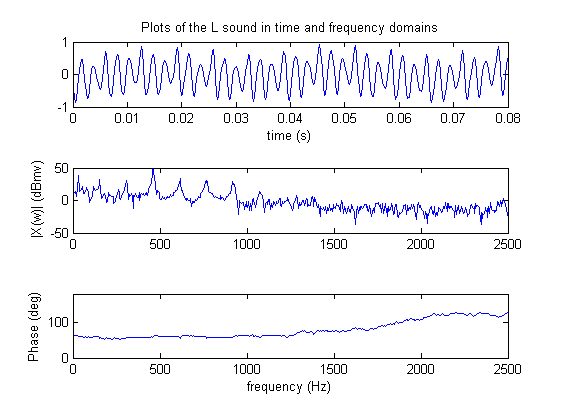
Horizontal Position -> Scroll the horizontal positioning until you see Delay=1.5 seconds

1. You may need a couple of practice rounds with this next part of the lab. First, make sure the scope is in Run mode by making sure the Run/Stop button is green. Then knock on the table or snap loudly to start the trigger. As soon as you see the trace start to appear on the screen, say the word “SAVE” into the microphone using your monotone speaking voice. When you say the word, instead of saying it in normal speaking rhythm, draw out the sound of each letter. For example, instead of “These” say “SSSSSSSSSSSAAAAAAAAAAAAAVVVVVVVVVVVVVE”. The screen will display 2s worth of data, so see if you can string out the word over as much of the 2s as possible without going over. When you have a good recording, press the Run/Stop button to freeze the screen. **Record the scope screen as an image file and the data points with 50000 points.**

## Part 3: Analyzing the speech signals in MATLAB

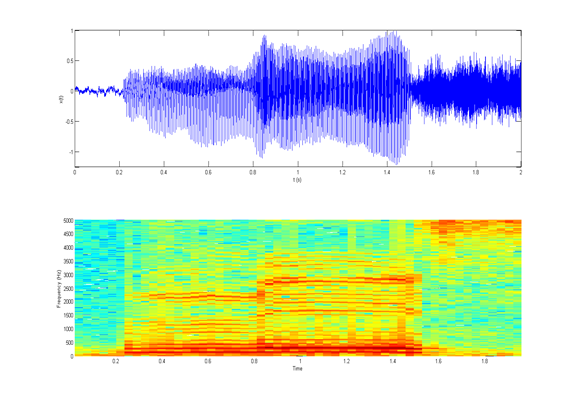
### Plotting the data in MATLAB

1. Create a new script file that will load your data and plot the magnitude and phase of the spectrum of the data. Use the “b=importdata(‘filename’,’,’,2)” command to import the data into MATLAB as a structure called “b”. Note that if your filename has an extension such as .csv, you must include this extension in the filename. Within any structure there is a matrix called “data” that stores all of the values. For files that are imported from the scope, the first column in this “data” matrix is the time vector, the next is channel 1, and so on. You can now access each column of the “data” matrix as b.data(:,X) where X is the column number. You should rename each column to its own variable, for example, “t=b.data(:,1)” to create the time vector, t.
2. Once the data is loaded, you can use the baf\_fft.m script to plot the spectrum of each signal as the magnitude and phase. Open the file baf\_fft.m in matlab and look at the header of the script to understand the various inputs and outputs. One of the inputs you have to supply to the script is the time difference between two samples, dt = t(2)-t(1), where t(2)and t(1), are the second and first elements of the time vector for your data. You should be able to determine this using MATLAB without having to open the data file.
3. Use the “subplot(3,1,X)” to make 4 different figures with 3 subplots each. The top subplot should be the time series of the phonetic or the word “These”. The middle plot should be the magnitude of the Fourier Transform of the corresponding time series in dBmV=20\*log10(sqrt(2)\*Xmag/.001). The bottom plot should be the phase in degrees. **Only the positive frequencies should be used for the magnitude and phase. An example of such a plot is shown in Figure 2 below. Limit the frequency range so that you can see all interesting frequencies.**



**Figure 2:** Plots relating to the L sound made in speech. This is the first type of MATLAB plot that you will produce for each of the phonetics and the whole word “save”. The top trace is the time domain version of the waveform captured by the microphone. Note that the timescale is such that you can clearly see the periodic nature of the sound. The middle trace is the magnitude of the Fourier Transform of the waveform. The bottom trace is the phase of the Fourier Transform. **You will produce one such plot for each of the sounds “s”, “a”, “v”, and the word “save”.**

1. Create a new Matlab script that will display the time-frequency plot for the three phonetics plus the word “These” using the spectrogram command. For reasonably good spectrogram plots I would use the command **spectrogram(y,[1:512],[],2048,Fs,’yaxis’**), but that’s just me. Note that Fs=1/dt from Part(c), is the frequency at which the time samples are taken by the scope. A slice of the 3D spectrogram plot at a given time index gives the spectrum of the signal over a short time window that is centered at that point in time. It is essentially a view similar to a spectrum analyzer display at that time during the signal. In this way, the evolution of the signal’s spectrum in time is displayed in the spectrogram. (Read this paragraph again until you understand it!)
2. Using the **subplot** command, create 4 figures with 2 subplots each. The top subplot should be the signal versus time and the bottom plot should be the spectrogram. **You should have one such plot for each of the phonetic sounds and then the whole word**. Figure 3 illustrates what one of these figures should look like but is for the whole word “These”. Can you see the different parts of the word in the time domain and in the spectrogram?



EE

ZZ

TH

Figure 3: Plot of a waveform in the time domain (top) and its corresponding spectrogram (bottom) for the whole word “these”. Note that the x-axis for each plot is time. For the spectrogram, the y-axis is frequency, and the z-axis (denoted by color) is the magnitude of the spectrum. You can see the change from one sound to the next in the word as the harmonics and frequencies change.

Data Collection Memo for Lab 6

Names:

Date:

Section:

# Part 1: From the Fourier Series to the Fourier Transform

1. Provide the screen captures that include the frequency domain for the periodic signal by itself and when multiplied by the pulse signal at each of the frequencies 500Hz, 250Hz, 125Hz, and 62.5Hz.

|  |  |
| --- | --- |
| Periodic signal with T0=1ms | When multiplied by pulse at 500Hz |
| When multiplied by pulse at 250Hz | When multiplied by pulse at 125Hz |
| When multiplied by pulse at 62.5Hz |  |

1. Using the tables, find the Fourier transform of the single triangle that you saw when the pulse signal was at 62.5Hz.
2. Sketch an outline of the spikes that appear in the power spectrum in each of your scope captures. What shape is this outline?
3. Explain why the sound from the speaker had a lower pitch as you decreased the frequency of the pulse signal.

# Parts 2 and 3: Looking at Speech Signals

It is strongly advised that you create ALL of your plots in lab. Before you leave the lab, you MUST have the instructor verify at least one magnitude plot and one spectrogram plot.

1. Provide the MATLAB plots of the signals with respect to time and the corresponding magnitude and phase as shown in Figure 2.

|  |  |
| --- | --- |
| Phonetic “s” sound time and FFT figure | Phonetic “a” sound time and FFT figure |
| Phonetic “v” sound time and FFT figure | Whole word “save” time and FFT figure |

1. Provide the MATLAB plots of the signals with respect to time and the corresponding spectrogram as shown in Figure 3.

|  |  |
| --- | --- |
| Phonetic “s” sound time and spectrogram figure | Phonetic “a” sound time and spectrogram figure |
| Phonetic “v” sound time and spectrogram plot | Whole word “save” time and spectrogram figure |

1. Why does the Magnitude of the FFT for the “a” and “v” sounds have clear peaks, while the “s” sound does not?
2. How is the information in the Magnitude plot of the whole word “save” different from the information in the spectrogram for the word “save”?
3. Indicate on the spectrogram for the whole word “save” at which times each of the phonetics is being said.
4. Given the bandwidths of the AMR-WB and AMR-NB codecs that you found in your prelab, what parts of your speech signal would be rejected by each of these codecs.