GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

ECE 2026 Spring 2025

Lab #3: Display and Analysis of Signals with Spectrograms and Fourier Analysis

Date: 10 – 21 Feb. 2025

Special Lab Instructions in Spring 2025: The labs will be held in person (or remotely through BlueJeans when needed). Students registered in a particular lab session are required to be present at the designated times. Attendances will be taken before each class. A total of six labs will be conducted. Each lab will typically last two weeks. The first week is devoted to students' **Q&As** and the second is for students' **demos** to the instructors for codes and lab results. Students will be grouped into teams of up to 5 by instructors before starting Lab 1. For each lab session, there will be two instructors administrating all activities. Each team will be given a breakout period of 15 minutes in each session for Q&As and verifications. Students are encouraged to discuss lab contents in teamwork. At the end of each lab, each student are required to turn in an individual lab report in a single pdf fill, containing answers to all lab questions, including codes and plots. Georgia Tech's **Honor Code** will be strictly enforced. See CANVAS Assignments for submission instructions.

Forgeries and plagiarism are a violation of the honor code and will be referred to the Dean of Students for disciplinary action. You are allowed to discuss lab exercises with other students, but you cannot give or receive any written material or electronic files. In addition, you are not allowed to use or copy material from old lab reports from previous semesters. Your submitted work must be your own original work.

1 Overview

Please read through the information below prior to attending your lab. The objective of this lab is to introduce more complicated signals that are related to the basic sinusoid. These signals which implement frequency modulation (FM) and amplitude modulation (AM) are widely used in communication systems such as radio and television. In addition, they can be used to create interesting sounds that mimic musical instruments. There are a number of demonstrations on the CD-ROM that provide examples of these signals for many different conditions. The resulting signal can be analyzed to show its time-frequency behavior by using the *spectrogram.*

CD-ROM
FM Synthesis

This lab studies visual display of spectral information over time in signals using *spectrogram* which shows their time-frequency content. An underlying objective of the lab is to learn more about the spectrogram. There are several specific steps that will be considered in this lab:

1. Synthesize a music signal with a MATLAB M-file, and display its spectrogram.

2. Study the frequency resolution of the spectrogram for two closely spaced sinusoids.

3. *Spectrogram:* Make empirical observations of the spectrogram as the section length is changed.

4. Synthesize a linear-FM chirp with a MATLAB M-file, and display its spectrogram.

5. *Spectrogram:* Create a spectrogram that displays negative frequencies, as well as positive ones.

6. Synthesize a frequency-modulated (FM) signal to match a given spectrogram. i.e, match specific time-frequency spectral content.

2 Pre-Lab

We have spent a lot of time learning about the properties of sinusoidal waveforms of the form:

$$x(t) = A\cos(2\pi f_0 t + \varphi) = \Re\left\{\left(Ae^{j\varphi}\right)e^{j2\pi f_0 t}\right\} \tag{1}$$

In this lab, we will extend our treatment of sinusoidal waveforms to more complicated signals composed of sums of sinusoidal signals, or sinusoids with changing frequency, i.e., frequency-modulated sinusoids.

2.1 Review: MATLAB Synthesis of Chirp Signals

CD-ROM
Spectrograms & Sounds: Wideband FM

A linear-FM *chirp* signal is a sinusoid whose instantaneous frequency changes linearly from a starting value to an ending one. The formula for such a signal can be defined by creating a complex exponential signal with a quadratic angle function $\psi(t)$. Mathematically, we define $\psi(t)$ as

$$\psi(t) = 2\pi\mu t^2 + 2\pi f_0 t + \varphi \tag{2}$$

The derivative of $\psi(t)$ yields an instantaneous *cyclic* frequency that changes *linearly* versus time.

$$f_i(t) = 2\mu t + f_0 \quad \text{(hertz)} \tag{3}$$

The slope of $f_i(t)$ is equal to 2μ and its $t = 0$ intercept is f_0. The frequency variation in (3) produced by the time-varying angle function is called *frequency modulation,* so these signals are called FM signals. Finally, since the linear variation of the frequency (3) can produce an audible sound similar to a siren or a bird chirp, linear-FM signals are also called *chirps.*

If the signal starts at time $t = t_1$ s with a frequency of f_1 Hz, and ends at time $t = t_2$ s with a frequency of f_2 Hz, then the slope of the line in (3) will be

$$\text{SLOPE} = 2\mu = \frac{f_2 - f_1}{t_2 - t_1} \tag{4}$$

Note that if the signal starts at time $t = 0$ s, then $f_1 = f_0$ is also the starting frequency. Otherwise, $f_0 = ?$

In MATLAB signals can only be synthesized by evaluating the signal's defining formula at discrete instants of time. These are called *samples* of the signal. For the chirp we use the following:

$$x(t_n) = A\cos(2\pi\mu t_n^2 + 2\pi f_0\, t_n + \varphi)$$

where t_n is the n^{th} time sample. The following MATLAB code will synthesize a linear-FM chirp:

```
   fSamp = 8000;   %-Number of time samples per second
   dt = 1/fSamp;
   tStart = 0;
   tStop = 1.5;
   tt = tStart:dt:tStop;
   mu = 600;
   fzero = 400;
   phi = 2*pi*rand;   %-- random phase
%
%%   psi = ????;    <=================== FILL IN THE CODE HERE
%
   cc = real( 7.7*exp(j*psi) );
%   soundsc( cc, fSamp );   %-- uncomment to hear the sound
   plotspec( cc+j*1e-12, fSamp, 256 ), colorbar, grid on   %-- with negative frequencies
```

(a) Determine the total duration of this synthesized signal in seconds, and also the length of the `tt` vector. Use MATLAB's `size` command to check that the signal vector `cc` has the expected size.

(b) Determine the range of frequencies (in hertz) that will be synthesized by the MATLAB script above, i.e., determine the minimum and maximum frequencies (in Hz) that will be heard. This will require that you relate the parameters μ, f_0, and φ to the minimum and maximum frequencies. Examine the MATLAB spectrogram to determine the instantaneous (cyclic) frequency $f_i(t)$ versus time. Zoom in to verify the correct starting and ending frequencies.

(c) The spectrogram usually shows only the frequency components for $f \geq 0$, but with the "tiny imaginary part" trick `plotspec` will show the negative frequency components. We will called this a *two-sided spectrogram.* Since the chirp signal is *real-valued*, the spectrum must have conjugate symmetry, so the magnitudes of the negative frequency components are a mirror image of those in the positive frequency region.

(d) Use `soundsc()` to listen to the signal in order to determine whether the signal's frequency content is increasing or decreasing. Notice that `soundsc()` needs to know two things: the vector containing the signal samples, and the rate at which the signal samples are to be played out. This rate should be the same as the rate at which the signal values were created (`fSamp` in the code above).
For more information do `help sound` and `help soundsc` in MATLAB.

(e) The test case above generates a chirp sound whose frequency starts low and chirps up. Modify the parameters so that the chirp starts at 3500 Hz and chirps down to 500 Hz.

2.2 Spectrograms: Time-Frequency Display of Signal Content

It is often useful to think of a signal in terms of its spectrum as discussed in Chapter 3. A signal's spectrum is a representation of the frequencies present in the signal. For a constant frequency sinusoid, the spectrum consists of two spikes, one at $\omega = 2\pi f_0$, the other at $\omega = -2\pi f_0$. For a more complicated signal the spectrum may be very interesting, e.g., the case of FM, where the spectrum components are time-varying. One way to represent the time-varying spectrum of a signal is the *spectrogram* (see Chapter 3 in the text). A spectrogram is produced by estimating the frequency content in short sections of the signal. The magnitude of the spectrum over individual sections is plotted as intensity or color over a two-dimensional domain of frequency and time.

When unsure about a command, use `help`.

There are some additional important things to know about spectrograms following last week's preliminary exercises:

1. In MATLAB the function `spectrogram` will compute the spectrogram. Type `help spectrogram` to learn more about this function and its arguments. The `spectrogram` function used to be called `specgram`, and had slightly different defaults—the argument list had a different order, and the output format always defaulted to frequency on the vertical axis and time on the horizontal axis.

2. A common call to the MATLAB function is `spectrogram(xx,1024,[],[],fs,'yaxis')`. The second argument[1] is the *section length* (or window length) which could be varied to get different looking spectrograms. The spectrogram is able to "see" very closely spaced separate spectrum lines with a longer (window) section length,[2] e.g., 1024 or 2048.

[1] If the second argument of `spectrogram` is made equal to the "empty matrix" then the default value used, which is the maximum of 256 and the signal length divided by 8.

[2] Usually the window (section) length is chosen to be a power of two, because a special algorithm called the FFT is used in the computation. The fastest FFT programs are those where the FFT length is a power of 2.

3. **(Negative) Frequency Range:** Normally the spectrogram image contains only positive frequencies. However, you can produce a spectrogram image containing negative frequencies ***if you use the function*** `plotspec` ***and if you make the input signal complex.*** Even if your signal is real, you can add a very tiny imaginary part, e.g., `xx = xx + j*1e-14`, to make it seem to be complex-valued.
Warning: This trick works nicely with the *SP-First* function called `plotspec`. However, when used with `spectrogram` the result does not have the negative frequency region in the proper location.

4. **Section Length of Spectral Analysis Windows:** A spectrogram is formed by taking successive short sections of a signal and performing an FFT analysis of each of those sections to get the spectrum. Since this is done repeatedly, the result is the spectrum versus time, where time is the location of the short sections. For a specific example, assume that the section length is 100, and the signal is a MATLAB vector `xx.` Then the first short section will be `xx(1:100).` The sections are usually overlapped and the default in `plotspec` is 50% overlap, so the second short section is `xx(51:150)`, the third `xx(101:200)`, and so on.

   The spectrogram image is, in effect, the spectrum versus time, so we need a reference time for each short section. In `plotspec` this reference time is the "midpoint" of the section. For the length-100 section, the reference index is 50, which is then converted to a time (in secs) by using the sampling rate (f_s). When the spectrogram is displayed as an image, these reference times are used along the horizontal axis.

In order to see a typical spectrogram, run the following code:

```
fs=8000; xx = cos(2000*pi*(0:1/fs:0.5)); plotspec(xx,fs,1024); colorbar
```

Notice that the spectrogram image contains one horizontal line at the correct frequency of the sinusoid.
To obtain a spectrogram with negative frequencies, try the following

```
xx = cos(2000*pi*(0:1/fs:0.5)); plotspec(xx+j*1e-9,fs,1024); colorbar
```

2.3 More Spectrogram: Decibels and Section Length

The common log scale used in engineering is decibels (dB), which is defined as $20\log_{10}(A)$ where A is amplitude. The built-in MATLAB `spectrogram` M-file uses a dB scale for amplitude when displaying its spectrogram image. The decibel has two notable features:

1. *Ratios become Differences:* On a dB scale, a numbers are represented with logarithms, so the ratio P/Q becomes
$$20\log_{10}(P/Q) = 20\log_{10}(P) - 20\log_{10}(Q)$$
If $A_2 = (1/10)A_1$ then A_2 is 20 dB lower than A_1, because with logs, we get
$$20\log_{10}(A_2) = 20\log_{10}((1/10)A_1) = 20\log_{10}(A_1) + 20\log_{10}(0.1) = 20\log_{10}(A_1) - 20\,\text{dB}.$$
Since ratios become differences, dB is most often used to compare the relative size of values.

2. The dB range must be restricted because $20\log_{10}(0) = -\infty$. If we want to map the linear amplitude range $[0,1]$ into dB, we must define a minimum dB level. Since $20\log_{10}(1) = 0\,\text{dB}$ is the maximum, other dB values for $[0,1]$ will be negative. The minimum dB level will chop off the bottom of $[0,1]$ and make it equal to $[\epsilon, 1]$ where ϵ is very small. For example, a dB range of 80 dB would define the minimum to be -80 dB, so $\epsilon = 10^{-80/20} = 10^{-4} = 0.0001$.

- Express the numbers 0.1, 1, 2, 5, 10 and 100 in dB.
- Convert -6 dB, -60 dB, and -80 dB to numbers.

Another topic is displaying spectrogram of signals is *section length* which determines the time and frequency resolution. Figure 1 shows a sinusoid analyzed with a constant section length, typically in power of 2 such as 32 or 512, overlapping 50% in time expressed in different colors and its corresponding spectrogram. It is clear that at the signal transition regions the frequency content is not just a single spectral line.

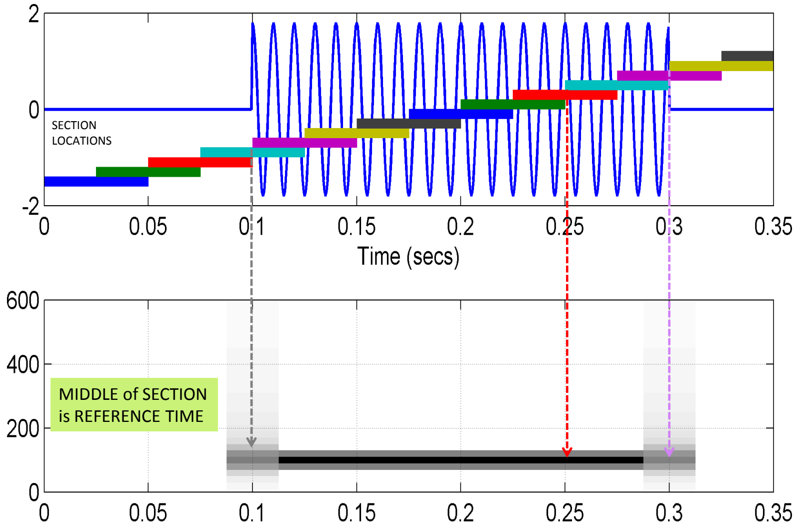


Figure 1: An illustration of the edge effect in spectrogram display.

2.4 Beat Note Spectrograms and Frequency Resolution

Beat notes as as a sum of two sinusoids provide an interesting way to investigate the time-frequency characteristics of spectrograms. Although some of the mathematical details require further study beyond this course, it is not difficult to appreciate the following issue: there is a fundamental trade-off between knowing which frequencies are present in a signal's spectrum and knowing how those frequencies vary with time. As discussed previously, a spectrogram estimates the frequency content over short sections of the signal; this is the *Section Length* parameter.[3] If we make the section length very short we can track rapid changes in the signal, usually changes in the frequency content. The tradeoff, however, is that shorter sections may not provide enough data to do an accurate frequency measurement. On the other hand, long sections allow the spectrogram to perform excellent frequency measurements, but fail to track sudden frequency changes. For example, if a signal is the sum of two sinusoids whose frequencies are nearly the same, a very long section length is needed to "resolve" the two sinusoidal components. This trade-off between the section length (in time) and the frequency resolution is akin to Heisenburg's Uncertainty Principle in physics. We can summarize this discussion by stating the following hypothesis:

> The *frequency resolution* of the spectrogram is *inversely proportional to the Section Length.* In other words, when the true spectrum has two lines (at f_1 and f_2) these two lines will be visible

[3] The section length is often called the window length; the two terms are used interchangeably in DSP.

as distinct lines in the spectrogram if $|f_1 - f_2| \approx C/T_{\text{SECT}}$ where C is a proportionality constant and T_{SECT} is the section duration in secs.

Note: When using `plotspec(xx,fs,Lsect)`, the section length ***in samples*** is an input argument to the spectrogram function. We can use the sampling rate to convert to duration, $T_{\text{SECT}} = L_{\text{SECT}}/f_s$.

We will use beat note signals which consist of two closely spaced spectral lines to confirm this hypothesis. A beat note signal may be viewed as a single frequency signal whose amplitude varies with time, *or* as the sum of two sinusoidal signals with different constant frequencies. Both views can be used to explain the effect of (window) section length when finding the spectrogram of a beat signal.

(a) Use the MATLAB code written in earlier lab to create and plot a beat signal defined via:

$$b(t) = 10\cos(2\pi(f_\Delta)t + \varphi_\Delta)\cos(2\pi(1024)t + \varphi_c),$$

with a duration of 5 s, and a sampling rate of $f_s = 8000$ samples/sec. The frequency f_Δ should be set to 4 Hz, but will be varied in later parts. The phases can be random.

(b) When $f_\Delta = 4$ determine the locations of the two spectrum lines that you expect to see in the spectrogram. In other words, derive (mathematically) the spectrum of the signal defined in part (a).

(c) Make the spectrogram of $b(t)$ using a (window) section length of $L_{\text{SECT}} = 256$ using the commands[4]:

```
plotspec(xx,fSamp,256); colorbar, grid on, zoom on
```

Comment on what you see. Are there two spectral lines, i.e., (horizontal lines across the spectrogram)? If necessary, use the zoom tool (in the MATLAB figure window), or `zoom on`, to examine the important regions of the spectrogram.

(d) It should not be possible to see both spectrum lines with $L_{\text{SECT}} = 256$. In order to get both lines a longer section length is needed, so try doubling the section length. Try $L_{\text{SECT}} = 512$, then $L_{\text{SECT}} = 1024$, and so on until you can discern two spectrum lines.[5] Then reduce the value of L_{SECT} little by little to get the smallest L_{SECT} that will work. Getting a value of L_{SECT} to the nearest 500 is sufficient.

As before, use zooming to examine the important regions of the spectrogram. Once you have two spectrum lines, record the value of L_{SECT} and determine whether the frequencies present in the spectrogram are correct. In addition, convert L_{SECT} to the section duration in seconds, T_{SECT}.

2.5 Fourier Series

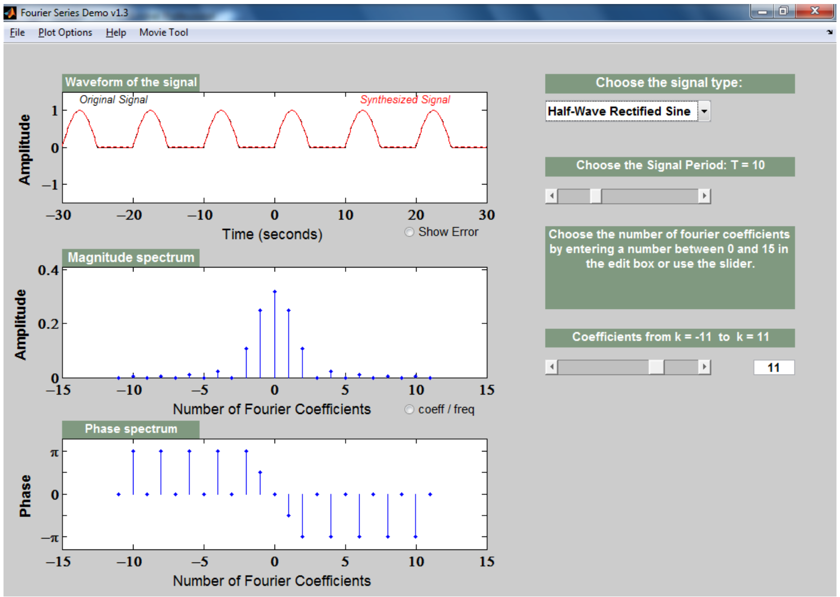
Another objective of this lab is to demonstrate usage of the `fseriesdemo` GUI. If you have installed the *SP-First* Toolbox, you will already have this demo on the `matlabpath`.

In this MATLAB GUI, you can choose the signal from a few pre-determined signal types and change the signal period. You can also decide the number of Fourier coefficients to compute and the corresponding magnitude and phase will be displayed accordingly.

Figure 2 shows the interface for the `fseriesdemo` GUI.

[4] Use `plotspec` instead of `specgram` in order to get a linear amplitude scale rather than logarithmic.

[5] Usually the window (section) length is chosen to be a power of two, because a special algorithm called the FFT is used in the computation. The fastest FFT programs are those where the FFT length is a power of 2.



Choose the signal type:

Half-Wave Rectified Sine

Choose the Signal Period: T = 10

Choose the number of fourier coefficients by entering a number between 0 and 15 in the edit box or use the slider.

Coefficients from k = -11 to k = 11

11

Figure 2: The **fseriesdemo** MATLAB GUI interface.

2.6 Triangle Wave and Its Fourier Series

The periodic triangular wave has a known Fourier Series. After consulting the text in Appendix C-2.2 on Fourier Series, we can write:

$$a_k = \begin{cases} \frac{-2}{\pi^2 k^2} = \frac{2}{\pi^2 k^2} e^{j\pi} & \text{for } k \text{ odd} \\ \frac{1}{2} & \text{for } k = 0 \\ 0 & \text{for } k \text{ even} \end{cases} \tag{5}$$

• Evaluate the coefficients for $k = 1$, 3, 5 and 15. Then compute the ratios a_3/a_1, a_5/a_1 and a_{15}/a_1.

Comment: Here's a general question: **are the Fourier Series coefficients independent of time scaling ?** First of all, $y(t) = x(bt)$ is scaling. For example, a plot of $x(3t)$ will be squeezed along the horizontal axis by a factor of $b = 3$.

What are the Fourier coefficients of $y(t) = x(bt)$ in terms of the Fourier coefficients of $x(t)$? If $x(t)$ has a period equal to T, then the period of $y(t)$ is T/b because $x(t)$ is squeezed by b. Thus, the fundamental frequency of $y(t)$ is $\frac{2\pi}{(T/b)} = b(\frac{2\pi}{T})$, i.e., it is scaled by b.

Now we write the Fourier Series integral for $y(t)$

$$(1/(T/b)) \int_{-T/2b}^{T/2b} y(t) e^{-j((2\pi k)/(T/b))t} dt = (b/T) \int_{-T/2b}^{T/2b} x(bt) e^{-j(2\pi k/T)(bt)} dt$$

Make a change of variables: $\lambda = bt$, and with $d\lambda = b(dt)$, you get

$$b(1/T) \int_{-T/2}^{T/2} x(\lambda) e^{-j(2\pi k/T)\lambda} (1/b) d\lambda = (1/T) \int_{-T/2}^{T/2} x(\lambda) e^{-j(2\pi k/T)\lambda} d\lambda = a_k$$

The scaling factor b cancels to give the RHS. So, time scaling doesn't change the $\{a_k\}$ coefficients. The Fourier Series coefficients of $y(t)$ are exactly equal to the Fourier Series coefficients of $x(t)$.

Use the GUI to do the following exercises. The parameters of the input signal are its frequency f_0 in Hz, and its phase φ in rads. The amplitude is one. The sampling rate for both the A/D converter and the D/A converter is f_s in samples/sec.

3 In-Lab Exercise

For the lab exercise, you will synthesize some signals, and then study their frequency content by using the spectrogram. The objective is to learn more about the connection between the *time-domain* definition of the signal and its *frequency-domain* content.

For the instructor verification, you will have to demonstrate that you understand concepts in a given subsection by answering questions from your lab instructor (or TA).

3.1 Spectrogram for a Chirp with Aliases

Use the code provided in the pre-Lab section as a starting point in order to write a MATLAB script or function that will synthesize a "chirp" signal. Then use that M-file in this section.

(a) What happens when we make a signal that "chirps" up to a very high frequency, and the *instantaneous frequency goes past half the sampling rate?* Generate a chirp signal that starts at 625 Hz when $t = 0$ s, and chirps up to $8,125$ Hz, at $t = 5$ s. Use $f_s = 2500$ Hz. Determine the parameters needed in Equation (2).

(b) Generate the chirp signal in MATLAB and make a spectrogram with a short section length, L_{SECT}, to verify that you have the correct starting and ending frequencies.[6] For your chosen L_{SECT}, determine the section duration T_{SECT} in secs.

(c) Explain why the instantaneous frequency seen in the spectrogram is *goes up and down* between zero and $f_s/2$, i.e., it does not chirp up to 8,125 Hz. There are two effects that should be accounted for in your explanation.
Note: If possible listen to the signal to verify that the spectrogram is faithfully representing the audio signal that you hear.

Completion of Lab Results (on separate Report page)

3.2 Spectrogram of Periodic Full-Wave Rectified Sinusoid

A periodic signal is known to have a Fourier Series, which is usually described as a *harmonic line spectrum* because the only frequencies present in the spectrum are integer multiples of the fundamental frequency. With the spectrogram, it is easy to exhibit this harmonic line characteristic. More detail about the materials below can be found in Lectures 7 & 8, and the Fourier Series text in Section 3-5 and Appendix C.

(a) Write a simple MATLAB script that will generate a periodic full-wave rectified sine wave once the period is given. The peak amplitude should be equal to 1. Here is a MATLAB one-liner that can form the basis of this script:

```
tt=0:(1/fs):tStop;xx=Amp*abs(sin(2*pi*tt/T));
```

The values of `fs, tStop, T, Amp` will have to be determined.

[6] There is no single correct answer for L_{SECT}, but you should pick a value that makes a smooth plot and lets you easily see the changing nature of the instantaneous frequency.

(b) Generate a full-wave rectified sine wave with T=1 sec, using a sampling rate of $f_s = 1000\,\text{Hz}$. The duration should be 5 secs, and Amp = 1.

(c) Make a spectrogram with a long section duration.[7] It is important to pick a section duration that is equal to an integer number of periods of the periodic full-wave rectified sine waveform created in the previous part. Define T_{SECT} to get exactly 5 periods, and then determine the section length L_{SECT} (an integer) to be used in `plotspec`.

(d) You should expect to see a "harmonic line spectrum" in the spectrogram. Since frequency is along the vertical axis, the harmonic lines will appear as horizontal lines in the spectrogram. Make a list of all the harmonic frequencies that you can see in the spectrogram. Note the Fourier series coefficient can be found as (for detail, see Section 3-5.2 in the textbook): $a_k = \frac{2}{\pi(1-4k^2)} = \frac{2}{\pi(4k^2-1)}e^{j\pi}$.

(e) Determine the fundamental frequency for the harmonic lines. Note here the fundamental frequency doubles the original frequency of the sine wave after the full-wave rectification operation.

(f) Measure the magnitudes of the first and third harmonic lines by using the fseriesdemo GUI first (see Lecture 7 slides for an illustration. Another example is in Figure 2 in the Pre-Lab part) and then confirm your values with the Fourier Series coefficient formula: $a_k = \frac{2}{\pi(1-4k^2)} = \frac{2}{\pi(4k^2-1)}e^{j\pi}$ (for odd k harmonics).

Completion of Lab Results (on separate Report page)

3.3 Decibels (dB): Seeing Small Values in the Spectrogram

The periodic full-wave rectified sine wave has a known Fourier series:

$$a_k = \frac{2}{\pi(1-4k^2)} = \frac{2}{\pi(4k^2-1)}e^{j\pi}$$

Thus, in the spectrogram we should see all harmonic lines. Furthermore, there should be an infinite number of harmonic frequency lines.

Where did all the harmonics go?

The answer is that the higher harmonics have amplitudes that are too small to be seen in a spectrogram that displays values with a *linear amplitude*. Instead, a logarithmic amplitude scale is needed.
Consult Section 2.3 in the Pre-Lab for a discussion of *decibels,* and then answer the following questions about decibels:

(a) In the language of dB, a factor of two is "6 dB." In other words, if B_2 is 6 dB bigger than B_1, then it is twice as big (approximately). Explain why this statement is true.

(b) Determine the dB difference between a_1 and a_3. In other words, a_3 is how many dB below a_1. Furthermore, explain why the dB difference.

Completion of Lab Results (on separate Report page)

[7] A long section duration in the spectrogram yields what is called a *narrowband spectrogram* because it will provide excellent resolution of the frequency components of a signal.

3.4 Spectrogram in dB

A variation of the *SP-First* function `plotspec` has been written to incorporate the dB amplitude scale. This new function is called `plotspecDB`, and its calling template is shown below:

```
function him = plotspecDB(xx,fsamp,Lsect,DBrange)
%PLOTSPECDB   plot a Spectrogram as an image
%        (display magnitude in decibels)
%  usage:   him = plotspec(xx,fsamp,Lsect,DBrange)
%      him = handle to the image object
%       xx = input signal
%    fsamp = sampling rate
%    Lsect = section length (integer, power of 2 is a good choice)
%             amount of data to Fourier analyze at one time
%  DBrange = defines the minimum dB value; max is always 0 dB
```

(a) Create a "dB-Spectrogram" for the 5-sec periodic full-wave rectified sine wave generated in Sect. 3.2. Use a `dBrange` equal to 80 dB. Notice that many more spectrum lines are now visible. List the frequencies of all the harmonic spectrum lines, or give a general formula.

(b) Generate a second full-wave rectified sine wave by changing the period in 3.2 to T=2 sec. Then make the dB-Spectrogram of this 5-sec full-wave rectified sine wave, being careful to select the section duration as an integer number of periods. Try different values of T_{SECT} with multiplier other than 5 times the new period. Identify one value of T_{SECT} that gives the "best" dB-spectrogram.

Completion of Lab Results (on separate Report page)

3.5 Linear and dB Spectrograms of Your Voice

Using the voice signal that you recorded for Lab #1, make a linear and a dB spectrogram with a long section length. Since the vowel region of the recording is a signal that is quasi-periodic, the spectrogram should exhibit a harmonic line characteristic during that time. Try different section lengths until the harmonic lines are easy to see in the spectrogram.

1. Turn in a print out of each the two spectrograms. Annotate the spectrogram to show the vowel region, as well as parameter values and measurements such as:
   The values of T_{SECT} and L_{SECT} used.
   The measured frequency separation of the (horizontal) harmonic lines that are seen in the vowel region.

2. From the frequency separation of the harmonic lines, measure the fundamental frequency in the vowel region. Also determine the fundamental period for this fundamental frequency.

3. Recall the value of the pitch period (in secs) measured in Lab #1. Compare the fundamental period determined in the previous part.

4. Comment on similarities and differences between the two period measurements.

Completion of Lab Results (on separate Report page)

ECE-2026 Lab #3 Spring-2025
LAB COMPLETION REPORT

Name: ____________________________ gtLoginUserName: ______________ Date: _______

Part 3.1 Chirp Aliasing

(a) Chirp formula: $\cos(\psi(t)) =$

(b) $L_{\text{SECT}} =$ and $T_{\text{SECT}} =$

(c) Guess why "ups and downs" in spectrogram (more about **sampling and aliasing** in Lab #4):

Part 3.2 Spectrogram of periodic full-wave rectified sine wave using a long section duration.

(b) $T_{\text{SECT}} =$ and $L_{\text{SECT}} =$

(d) List frequencies of visible harmonics =

(e) Fundamental Frequency =

(f) $|a_1| =$ $|a_3| =$ and dB difference =

Part 3.3 Questions about decibels:

(a) Explain 6 dB.

(b) dB difference between $|a_1|$ and $|a_3|$ =

Part 3.4 dB-Spectrogram of periodic full-wave rectified sine wave using a long-duration section.

(a) List frequencies of harmonic lines =

(b) Best value of T_{SECT} =

Part 3.5 Using the voice signal that you have recorded for Lab #1, make a linear and a dB spectrogram with a long section length, e.g., $L_{\text{SECT}} = 512$. Since the vowel region of the recording is a signal that is quasi-periodic, the spectrograms should exhibit a harmonic line characteristic (parllel horizontal lines) during that time. Try different section lengths L_{SECT} until the harmonic lines are easy to see in the spectrograms.