GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

EE 2025 Fall 2009
Lab #3: AM and FM Sinusoidal Signals

Date: 8–14 Sept. 2009

You should read the Pre-Lab section of the lab and do all the exercises in the Pre-Lab section before your assigned lab time.

ITS: When you come to the lab, you **must** answer the online ITS questions. You can use MATLAB or any notes you might have but you cannot discuss the exercises with any other students.

Verification: The Warm-up section of each lab must be completed **during your assigned Lab time** and the steps marked *Instructor Verification* must also be signed off **during the lab time**. When you have completed a step that requires verification, simply raise your hand and demonstrate the step to the TA or instructor. After completing the warm-up section, turn in the verification sheet to your TA *before leaving the lab.*

It is only necessary to turn in Section 4 as the lab report for this lab. More information on the lab report format can be found on **`t-square`** under the **INFO** link. Please ***label*** the axes of your plots and include a title and Figure number for every plot. In order to reduce *orphan plots*, include each plot as a figure *embedded* within your report. This can be done easily with MATLAB's `notebook` capability. For more information on how to include figures and plots from MATLAB in your report file, consult the **INFO** link on **`t-square`**, or ask your TA for details.

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 and you are allowed to consult old lab reports, but you cannot give or receive written material or electronic files. Your submitted work should be original and it should be your own work.

Due Date: The report will be **due during the period 15–21 Sept. at the start of your lab.**

---

1 Introduction

The objective of this lab is to introduce more complicated signals that are related to the basic single-frequency 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.

FM Synthesis

2 Pre-Lab

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

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

In this lab, we will extend our treatment of sinusoidal waveforms to more complicated signals composed of sums of sinusoidal signals with different frequencies, and also sinusoids with changing frequency, e.g., *frequency-modulated* (FM) signals.

2.1 Amplitude Modulation

If we add several sinusoids, each with a different frequency (f_k), we cannot use the phasor addition theorem, but we can still express the result with complex amplitudes as:

$$x(t) = \sum_{k=1}^{N} A_k \cos(2\pi f_k t + \varphi_k) = \Re\left\{ \sum_{k=1}^{N} \left(A_k e^{j\varphi_k} \right) e^{j\,2\pi f_k t} \right\} \tag{2}$$

where $A_k e^{j\varphi_k}$ is the complex amplitude of the k^{th} complex exponential term. The choice of f_k will determine the nature of the signal—for amplitude modulation or beat signals we pick two or three frequencies that are very close together, see Chapter 3.

2.1.1 Beat Control GUI

beatcon.m

To facilitate your experiments with beat notes and AM signals, the MATLAB GUI tool called **`beatcon`** has been created. This *user interface controller* will exhibit the basic signal shapes for beat signals and play the signals. A small control panel will appear on the screen with *buttons* and *sliders* that vary the different parameters for the beat signals. In addition, the GUI can also call a user-written function named `beat.m`. Experiment with the **`beatcon`** control panel and use it to produce a beat signal with two frequency components: one at 690 Hz and the other at 700 Hz. Use a longer duration than the default when you want to listen to the *beat frequency* sound.

2.2 Frequency Modulated Signals

In this lab, we will examine signals whose frequency content varies as a function of time. Recall that in a constant-frequency sinusoid (1) the argument of the cosine is $(2\pi f_0 t + \varphi)$ which is also the exponent of the complex exponential. We will refer to the argument of the cosine as the ***angle function***. In (1), the *angle function* changes *linearly* versus time, and its time derivative, $2\pi f_0$, equals the constant frequency of the cosine.

FM Synthesis

A generalization is available if we adopt the following notation for the class of signals with time-varying angle functions:

$$x(t) = A\cos(\psi(t)) = \Re\{Ae^{j\,\psi(t)}\} \tag{3}$$

The time derivative of the angle function $\psi(t)$ in (3) gives a frequency that we call the *instantaneous (radian) frequency:*

$$\omega_i(t) = \frac{d}{dt}\psi(t) \qquad \text{(rad/sec)}$$

If we prefer units in hertz, then we divide by 2π to define the *instantaneous (cyclic) frequency:*

$$f_i(t) = \frac{1}{2\pi}\frac{d}{dt}\psi(t) \qquad \text{(Hz)} \tag{4}$$

2.3 Chirp, or Linearly Swept Frequency

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, i.e., by defining $\psi(t)$ in (3) as

$$\psi(t) = 2\pi\mu t^2 + 2\pi f_0 t + \varphi \tag{5}$$

The derivative of $\psi(t)$ yields an instantaneous (cyclic) frequency (4) that changes *linearly* versus time.

$$f_i(t) = 2\mu t + f_0$$

The slope of $f_i(t)$ is equal to 2μ and its intercept is equal to f_0. If the signal starts at time $t = 0\,\text{s}$, then f_0 is also the starting frequency. The frequency variation produced by the quadrative angle function is called *linear frequency modulation,* abbreviated LFM. Finally, since the linear variation of the frequency can produce an audible sound similar to a siren or a chirp, linear-FM signals are also called *chirps.*

2.4 MATLAB Synthesis of Chirp Signals

The following MATLAB code will synthesize a linear-FM (chirp) signal:

```
fsamp = 8000;
delta_t = 1/fsamp;  %- time increment between samples
dur = 1.1;
tt = 0 : delta_t : dur;
f1 = 400;
psi = 2*pi*(100 + f1*tt + 500*tt.*tt);
xx = real( 7.7*exp(j*psi) );
soundsc( xx, fsamp );
```

(a) Determine the total duration of the synthesized signal in seconds, and also the length of the `tt` vector.

(b) 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 get the samples by evaluating the LFM angle formula (5) at $t = t_n$, so the MATLAB code is equivalent to 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. From the MATLAB code above, identify the values of A, μ, f_0, and φ.

(c) 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. Make a sketch by hand of the instantaneous frequency versus time.

(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 were created (`fsamp` in the code above). For more information do `help sound` and `help soundsc` in MATLAB.

2.5 Spectrogram: Three M-files

It is often useful to think of a signal in terms of its spectrum. A signal's spectrum is a representation of the frequencies present in the signal. For a constant frequency sinusoid as in (1) 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 on a two-dimensional plot versus frequency and time.

Sounds & Spectro-grams

When unsure about a command, use `help`.

There are a few important things to know about using MATLAB to create spectrograms:

1. Newer versions MATLAB (starting in 2008) have a function called `spectrogram` that will compute the spectrogram.[1] Type `help spectrogram` to learn more about this function and its arguments.

2. A common call to the new `spectrogram` function is `spectrogram(xx,Lsect,[ ],[ ],fsamp,'yaxis')` where `fsamp` is the sampling rate for the signal `xx`, and the second argument is the section length[2] (or window length) given as the number of samples (an integer). The string `'yaxis'` make the plot with the frequency axis on the y-axis; the default is to put the frequency axis on the x-axis. The empty matrices for the third and fourth arguments are used to take the default values for those. The third argument is the section overlap whose default value is half of `Lsect`; the fourth argument is the FFT length which defaults to the section length.

   The full calling sequence is `spectrogram(xx, Lsect, Noverlap, NFFT, fs, axisstring)`.

   Changing the second argument, the *window length*, will give different looking spectrograms. For example, the spectrogram might need a longer window length, e.g., 1024 or 2048, in order to "see" separate spectrum lines that are closely spaced in frequency.

3. Prior to version 7, MATLAB's spectrogram function was named `specgram`. The argument list for `specgram` is different from that of `spectrogram`.

4. A common call to the `specgram` function is `specgram(xx,N,fs)`. In this form, the second argument is the *window length*. The full calling sequence is `specgram(xx,NFFT,fs,Lsect,Noverlap)`.

5. If you are working at home, you might not have either of the spectrogram functions because they are part of the *Signal Processing Toolbox*. In that case, use the function `plotspec(xx,fs,Lsect)` which is part of the *SP-First Toolbox* which can be downloaded from the link on **`t-square`**:
   `http://users.ece.gatech.edu/mcclella/SPFirst/Updates/SPFirstMATLAB.html`

   • Note: The argument list for `plotspec` has a different ordering from `spectrogram` and `specgram`, because the *section length* (or, window length) is an optional third argument in `plotspec` (its default value is 256).The overlap in `plotspec` is alwys 50%. In addition, `plotspec` does not use color for the spectrogram; instead, darker shades of gray indicate larger values with black being the largest. The final difference is the amplitude format; `plotspec` displays a gray level that is proportional to the amplitude, whereas, `spectrogram` and `specgram` use a decibel scale so that the displayed color is proportional to $10\log_{10}|B(t,f)|$ where $|B|$ is the amplitude of the spectrogram.

6. Spectrograms are numerical calculations and provide only an estimate of the time-varying frequency content of a signal. There are theoretical limits on how well they can actually represent the frequency content of a signal. Another lab on the *SP-First* CD-ROM treats this issue with a project that uses the spectrogram to extract the frequencies of piano notes.

7. **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 `plotspec`. However, when used with `spectrogram`, or `specgram`, it produces an image that does not have the negative frequency region in the expected location.

[1] Prior to verion 7, MATLAB's spectrogram function was named `specgram`.

[2] Usually the window 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 signal length is a power of 2.

Make some Test Spectrograms In order to see a typical spectrogram, run the following code:

```
fs=8000; xx = cos(2000*pi*(0:1/fs:0.5)); spectrogram(xx,512,[ ],[ ],fs,'yaxis'); colorbar
```

or, `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.

Spectrogram with Negative Frequencies To create a spectrogram that also shows negative frequencies, try the following

```
fs=8000; xx = cos(2000*pi*(0:1/fs:0.5)); plotspec(xx+j*1e-9,fs,512); colorbar
```

Explain why the spectrogram now contains two spectral lines.

Spectrogram of a Chirp Next, show the spectrogram of the chirp produced in Section 3.3. Describe the important features that you see in the spectrogram of the chirp and explain why they are correct.

3 Warm-up

The instructor verification sheet may be found at the end of this lab. The "Beat Control GUI" is part of the *SP First* toolbox, and it should be on the MATLAB path already installed on the computers in the ECE Labs.

3.1 Intelligent Tutoring System (ITS)

During this lab you should run ITS and answer the questions (see the instructions below). In general, ITS will be used throughout the semester to measure progress on the basic concepts covered in ECE-2025.

1. The ITS is only accessible from the ECE network (e.g., in the Klaus classrooms, or VanLeer)
2. The link to ITS is: `https://www-dev3.ece.gatech.edu/ece2025/ITS/php/`. To open ITS, login with your AD username and password (same as for T-square)

3.2 Beat Control GUI

Use the **`beatcon`** control panel to produce a beat signal with two frequency components: one at 700 Hz and the other at 710 Hz. Use a longer duration than the default to hear the "beating" sound. Demonstrate the plot and sound to your TA. The signal is periodic—determine its fundamental period.

Instructor Verification (separate page)

3.3 Function for a Chirp

Use the code provided in the pre-Lab section as a starting point in order to write a MATLAB function that will synthesize a "chirp" signal according to the following template. This will require that you relate the chirp parameters μ, f_0, and φ to the starting and ending frequencies. Fill in code where you see `???`.

```
function   [xx,tt] = myLFMchirp( f1, f2, dur, fsamp )
%MYLFMCHIRP      generate a linear-FM chirp signal
%
%  usage:   xx = myLFMchirp( f1, f2, dur, fsamp )
%
%      f1 = starting frequency
%      f2 = ending frequency
%     dur = total time duration, assume the start time is t=0
%   fsamp = sampling frequency  (OPTIONAL: default is 8000)
%
%      xx = (vector of) samples of the chirp signal
%      tt = vector of time instants for t=0 to t=dur
%
if( nargin < 4 )   %-- Allow optional input argument
   fsamp = 8000;
end
tt = ???
%%%
%%%--  Calculate slope and intercept of linear frequency change
%%%
psi = 2*pi*( ????*tt + ?????*tt.*tt);
xx = real( exp(j*psi) );
```

As a test case, generate a chirp sound whose frequency starts at 2000 Hz and ends at -500 Hz; its duration should be 0.75 sec and the sampling rate should be $f_s = 8000$ samples/sec. Give the exact calling sequence for `myLFMchirp.m` in order to produce the test case.

Listen to the chirp using the `soundsc` function. What does it mean to have a negative frequency, and how does it sound? Use the spectrogram in the next section to help explain what you hear.

Instructor Verification (separate page)

3.3.1 Spectrogram of the Chirp

Next, show the spectrogram of the LFM chirp produced in the previous part, Section 3.3. Make sure that you produce a spectrogram image that includes negative frequencies. Describe the important features that you see in the spectrogram of the chirp and explain why they are correct.

Instructor Verification (separate page)

4 Lab Project: Chirps and Beats

For the lab exercise and lab report, you will synthesize some AM and FM signals. In order to verify that these signals have the correct frequency content, you will use the spectrogram. Your lab report should discuss the connection between the *time-domain* definition of the signal and its *frequency-domain* content.

4.1 Spectrum Values of Periodic Signals

Periodic time signals will always have a harmonic line spectrum structure. This property underlies the Fourier Series method, where we can write the periodic signal as

$$x(t) = \sum_k a_k e^{jk\omega_0 t} \tag{6}$$

where ω_0 is the fundamental frequency in rad/s. If a mathematical formula is available for the time signal $x(t)$, then it might be possible to derive an exact mathematical expression for the Fourier Series coefficients, $\{a_k\}$, but sometimes this isn't possible.

In cases where the Fourier Series integral cannot be reduced to a simple expression, it might still be possible to calculate the $\{a_k\}$ coefficients numerically. If we know that $a_k = 0$ for $|k| > k_{\max}$, then the following idea will work if the sampling rate `fsamp` is greater than $2k_{\max}\omega_0/2\pi$.

> The spectrogram functions in MATLAB can be used to extract the exact spectrum values of a periodic signal. One way to accomplish this is to use the *SP-First* function called `spectgr`, which can give outputs instead of making a plot. If the signal `xx` has a period of `LL` samples when sampled at the rate `fsamp`, then
>
> ```
> [BB,FF,TT] = spectgr(xx+j*1e-13, LL, fsamp, ones(LL,1)/LL, 0);     (7)
> ```
>
> will give the exact numerical values of the spectrum in the 2D array `BB`. All the columns of `BB` will be identical because, for each section, the signal is being analyzed over one period. The vector `FF` is a list of the frequencies where the spectrum has been computed; `TT` is a list of the times along the t-axis of the spectrogram.

4.1.1 Periodic Beat Signals

Beat notes provide an interesting way to investigate the time-frequency characteristics of spectrograms. Although some of the mathematical details of spectrograms are beyond the reach of this course, it is not difficult to appreciate the following concept: there is a fundamental trade-off between knowing which frequencies are present in a signal (or its spectrum) and knowing how quickly those frequencies change versus time. As mentioned previously in Section 2.5, a spectrogram estimates the frequency content over short sections of the signal. If we make the section length very short we can track rapid changes in the time signal, e.g., places where the frequency changes. However, very short sections do not provide enough data to do accurate frequency measurement. On the other hand, long sections allow the spectrogram to perform excellent frequency measurements, but make it impossible to track sudden frequency changes. The long section length is valuable especially when we want to "resolve" two closely spaced frequency components. This trade-off between the section length (in time) and frequency resolution is equivalent to Heisenburg's Uncertainty Principle in physics. More discussion of the spectrogram is available in the last chapter of *SP-First*.

A beat note signal may be viewed *either* as the product of two sinusoids which leads to the interpretation that it is a high frequency frequency signal whose amplitude varies with time, *or* as the sum of two signals with different constant frequencies. Both views will be needed to explain the effect of window length when studying the spectrogram of a beat signal.

(a) First of all, test the exact spectrum computation idea in (7) on a single sinusoid. Generate a sinusoid $x(t) = 13\cos(2\pi(40)t - 0.2\pi)$ at a sampling rate of `fsamp = 1000` samples/s. The period of $x(t)$ is 0.025 s which is 25 samples. Verify that the `spectgr` function as in (7) will obtain the correct values of the spectrum. In addition, explain why almost all of the spectrum values are zero.

(b) Create a beat signal defined via: $b(t) = 13\cos(2\pi(16)t)\cos(2\pi(800)t + 1.3\pi)$, with a duration of 1.1 secs. Use a sampling rate of $f_s = 8000$ samples/sec to produce the signal in MATLAB. Plot a few periods of $b(t)$.

(c) Derive (mathematically) the spectrum of the signal defined in the previous part, part (b). Make a sketch (by hand) of the spectrum with the correct frequencies and complex amplitudes.

(d) Determine the fundamental period of $b(t)$ in secs. When $f_s = 8000$ samples/s, determine the number of samples in one fundamental period.

(e) Use `spectgr` as in (7) to get the complex amplitudes in the spectrum from a numerical calculation. Determine the section length that will give the correct complex amplitudes; call it L_b.

(f) Use `plotspec` with a section length that is $4L_b$ to produce a spectrogram image. Repeat with a section length of $\frac{1}{4}L_b$.

(g) Compare the two results in the previous part, and explain why one works and one fails to see the correct spectrum.

4.1.2 Periodic FM Signals

It is realtively easy to define a signal with sinusoidal frequency modulation to be periodic. For example, $x(t) = A\cos(2\pi f_c t + \alpha\cos(2\pi f_m t + \varphi))$ is periodic for appropriate choices of the frequencies f_c and f_m.

(a) It is easy to create a periodic FM signal; for example,

$$s(t) = 13\cos(2000\pi t + 100\sin(8\pi t)) \tag{8}$$

Determine the fundamental period of this signal, i.e., the shortest period.

(b) For the signal $s(t)$ in the previous part, the instantaneous frequency provides a good explanation for what you hear when playing $s(t)$ with `soundsc` in MATLAB. Derive the mathematical expression for the instantaneous frequency. In addition, make a spectrogram for $s(t)$ by using `fsamp = 8000` samples/s, and show that the spectrogram matches the formula for the instantaneous frequency.

(c) A similar periodic FM signal is given by

$$c(t) = 13\cos(2000\pi t + 1.5\cos(100\pi t)) \tag{9}$$

Determine the fundamental period of this signal.

(d) In this case, a sampled version of $c(t)$ will sound much different from the time-frequency profile predicted by the instantaneous frequency. Use `fsamp = 8000` samples/s. Therefore, it is better to represent $c(t)$ by its Fourier Series. To begin, make a spectrogram plot with a section length that is equal to ten periods. Use the spectrogram image to determine which harmonics are likely to be present in the Fourier Series synthesis summation:

$$c(t) = \sum_k a_k e^{jk\omega_0 t}$$

It is sufficient to list only those indices k that are nonnegative.
Note: with `plotspec` it is not possible to see spectrum lines below about 0.01, so some of the nonzero lines might be missing (but you can find them in the next part).

(e) Find the values of the Fourier Series coefficients using the numerical method in (7), and list them in a table that gives the magnitude and phase of each one. Amplitudes below 10^{-4} can be ignored.

(f) If we assume that human hearing is performing a spectrogram as its front-end processing for perception, then it might be possible to explain why these two FM signals sound different. Can you provide an explanation using the context what what you've done in this lab?

Lab #3
ECE-2025
Fall-2009
INSTRUCTOR VERIFICATION SHEET

Turn this page in to your TA before the end of your lab period.

Name: _____ Date of Lab: _____

Answered ITS questions:

Verified:_____ Date/Time:_____

Part 3.2 Demonstrate usage of the Beat Control GUI.

Verified:_____ Date/Time:_____

Part 3.3 Demonstrate the `myLFMchirp.m` function. In the space below write the MATLAB code to show how you would call the function with a correct set of arguments.

Verified:_____ Date/Time:_____

Part 3.3.1 For the ***two-sided*** spectrogram of the chirp, describe the important features that you see in the spectrogram and explain why they are correct.

Verified:_____ Date/Time:_____