GEORGIA INSTITUTE OF TECHNOLOGY

SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

EE 2200 Fall 1998

Lab #4: AM and FM Sinusoidal Signals

Date: week of 19 Oct 1998

Lab Quiz: we will have an on-line quiz given in lab during the week of 26-Oct.
It will cover MATLAB issues.

This is *the official* Lab#4 description; it is nearly *identical* to Lab-A in Appendix C.4 of the text.

⟹ One of the warm-up exercises can be done prior to lab!

The Warm-up section of each lab must be completed in Lab and the steps marked *Instructor Verification* must also be signed off **during the lab time**.

The lab report for this lab will be informal: discuss your results from section 4. Staple the **Instructor Verification** sheet to the end of your lab report.

The report will due during the week of **26-Oct** 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 sinusoid. These signals which implement frequency modulation (FM) and amplitude modulation (AM) are widely used in communication systems such as radio and television, but they also 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.

CD-ROM

FM Synthesis

2 Overview

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

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

In this lab, we will continue to investigate sinusoidal waveforms, but for more complicated signals composed of sums of sinusoidal signals, or sinusoids with changing frequency.

2.1 Amplitude Modulation

If we add several sinusoids, each with a different frequency (f_k) we can express the result as:

$$x(t) = \sum_{k=1}^{N} A_k \cos(2\pi f_k t + \phi_k) = \Re e\left\{\sum_{k=1}^{N} Z_k\, e^{j2\pi f_k t}\right\} \tag{2}$$

where $Z_k = A_k e^{j\phi_k}$ is the complex exponential amplitude. The choice of f_k will determine the nature of the signal—for amplitude modulation we pick two or three frequencies very close together, see Chapter 3.

2.2 Frequency Modulated Signals

We will also look at signals in which the frequency varies as a function of time. In the constant-frequency sinusoid (1) the argument of the cosine is also the exponent of the complex exponential, so the phase of this signal is the exponent $(2\pi f_0 t + \phi)$. This phase function changes *linearly* versus time, and its time derivative is $2\pi f_0$ which equals the constant frequency of the cosine.

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

CD-ROM *FM Synthesis*

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

The time derivative of the phase from (3) gives a frequency

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

but we prefer units of hertz, so we divide by 2π to define the *instantaneous 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

CD-ROM *Spectrograms & Sounds: Wide-band FM*

A *chirp* signal is a sinusoid whose frequency changes linearly from some low value to a high one. The formula for such a signal can be defined by creating a complex exponential signal with quadratic phase by defining $\psi(t)$ in (3) as

$$\psi(t) = 2\pi\mu t^2 + 2\pi f_0 t + \phi$$

The derivative of $\psi(t)$ yields an instantaneous 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 $t = 0$, then f_0 is also the starting frequency. The frequency variation produced by the time-varying phase is called *frequency modulation*, and this class of signals is called FM signals. Finally, since the linear variation of the frequency can produce an audible sound similar to a siren or a chirp, the linear-FM signals are also called “chirps.”

2.4 Advanced Topic: Spectrograms

It is often useful to think of signals in terms of their spectra. 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 $2\pi f_0$, the other at $-2\pi f_0$. For more complicated signals the spectra may be very interesting and, in the case of FM, the spectrum is considered to be 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 found 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.

CD-ROM *Sounds & Spectrograms*

There are a few important things to know about spectrograms:

1. In MATLAB the function `specgram` will compute the spectrogram, as already explained in Lab 3. Type `help specgram` to learn more about this function and its arguments.

2. Spectrograms are numerically calculated and only provide 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. Lab 11 will treat this problem when we use the spectrogram to extract the frequencies of piano notes.

3 Warm-up

The instructor verification sheet may be found at the end of this lab.

3.1 Beat Control GUI

CD-ROM
beatcon.m

To assist you in your experiments with beat notes, the tool called `beatcon` has been created. This *user interface controller* will exhibit the basic signal shapes for beats 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. It can also call a user-written function called `beat.m`. Experiment with the `beatcon` control panel and use it to produce a beat signal with two frequency components at 850 Hz and 870 Hz. Demonstrate the plot and sound to your TA.

Instructor Verification (separate page)

3.2 MATLAB Synthesis of Chirp Signals

(a) The following MATLAB code will synthesize a chirp:

```
fsamp = 11025;
dt = 1/fsamp;
dur = 1.8;
tt = 0 : dt : dur;
psi = 2*pi*(100 + 200*tt + 500*tt.*tt);
xx = real( 7.7*exp(j*psi) );
soundsc( xx, fsamp );
```

Determine the range of frequencies (in hertz) that will be synthesized by this MATLAB script. Make a sketch by hand of the instantaneous frequency versus time. What are the minimum and maximum frequencies that will be heard?

Instructor Verification (separate page)

(b) Listen to the signal to verify that it has the expected frequency content (use `soundsc()`). Also, compute the spectrogram of your chirp using the MATLAB function: `specgram(xx,[],fsamp)`.

Instructor Verification (separate page)

4 Lab A: Chirps and Beats

4.1 Function for a Chirp

Use the code provided in the warm-up to help you write a MATLAB function that will synthesize a "chirp" signal according to the following comments:

```
function   xx = mychirp( f1, f2, dur, fsamp )
%MYCHIRP      generate a linear-FM chirp signal
%  usage:   xx = mychirp( f1, f2, dur, fsamp )
%       f1 = starting frequency
%       f2 = ending frequency
%      dur = total time duration
%    fsamp = sampling frequency  (OPTIONAL: default is 11025)
%
if( nargin < 4 )   %-- Allow optional input argument
   fsamp = 11025;
end
```

As a test case, generate a chirp sound to match the frequency range of the chirp in the warm-up. Listen to the chirp using the `sound` function. Include a listing of the `mychirp.m` function that you wrote.

When unsure about a command, use `help`.

4.2 Synthesize a Chirp

Use the `mychirp` function to synthesize a "chirp" signal for your lab report. Use the following parameters:

1. A total time duration of 3 secs. with a D/A conversion rate of $f_s = 11025$ Hz.
2. The instantaneous frequency starts at 5,000 Hz and ends at 300 Hz.

Listen to the signal. What comments can you make regarding the sound of the chirp (e.g., is it linear)? Does it chirp down, or chirp up? Create a spectrogram of this chirp signal, and use it to verify that you have the correct instantaneous frequencies.

4.3 Another Chirp

Synthesize a second "chirp" signal (for your lab report) with the following parameters:

1. A total time duration of 3 secs. with a D/A conversion rate of $f_s = 11025$ Hz.
2. The instantaneous frequency starts at 3,000 Hz and ends at $-2,000$ Hz (negative frequency).

Listen to the signal. Does it chirp down, or chirp up, or both? Create a spectrogram of this second chirp signal. Use the theory of the spectrum (with its positive and negative frequency components) to help explain what you hear and what you see in the spectrogram.

4.4 Beat Notes

In the section on beat notes in Chapter 3 of the text, we analyzed the situation in which we had two sinusoidal signals of slightly different frequencies; i.e.,

$$x(t) = A\cos(2\pi(f_c - f_\Delta)t) + B\cos(2\pi(f_c + f_\Delta)t) \tag{5}$$

In this part, we will compute samples of such a signal and listen to the result.

(a) Write an M-file called `beat.m` that implements (5) and has the following as its first lines:

```
function        [xx, tt] = beat(A, B, fc, delf, fsamp, dur)
%BEAT    compute samples of the sum of two cosine waves
%  usage:
%       [xx, tt] = beat(A, B, f, delf, fsamp, dur)
%
%              A = amplitude of lower frequency cosine
%              B = amplitude of higher frequency cosine
%             fc = center frequency
%           delf = frequency difference
%          fsamp = sampling rate
%            dur = total time duration in seconds
%             xx = output vector of samples
%--OPTIONAL Output:
%             tt = time vector corresponding to xx
```

Hand in a copy of your M-file. You might want to call the `sumcos` written in Lab 2 to do the calculation. The function could also generate its own time vector. You may elect to not implement the second output vector `tt`, but it is quite convenient for plotting.

CD-ROM
`beatcon.m`

(b) To assist you in your experiments with beat notes a tool called `beatcon` has been created. This *user interface controller* actually calls your function `beat.m`, if you check the box [Use External beat()] in the lower left-hand corner of the GUI. Therefore, before you invoke `beatcon` you should be sure your M-file is free of errors.

Test the M-file written in part (a) via `beatcon` by using the values `A=10`, `B=10`, `fc=1000`, `delf=10`, `fsamp=11025`, and `dur=1` secs. Plot the first 0.2 seconds of the resulting signal. Describe the waveform and explain its properties. Hand in a copy of your plot with measurements of the period of the "envelope" and period of the high frequency signal underneath the envelope.

(c) For this part, set `delf` to 10 Hz. Send the resulting signal to the D-to-A converter and listen to the sound (there is a *button* on `beatcon` that will do this for you automatically). Explain the nature of the sound based on the waveform plotted in part (b) and on the theory developed in Chapter 3.

(d) (Optional) Experiment with different values of the frequency difference f_Δ.

4.5 More on Spectrograms

Beat notes provide an interesting way to investigate the time-frequency characteristics of spectrograms. Although some of the mathematical details are beyond the reach of this course, it is not difficult to understand the following issue: there is a fundamental trade-off between knowing which frequencies are present in a signal (or its spectrum) and knowing how those frequencies vary with time. As mentioned previously in Section 2.4, a spectrogram estimates the frequency content over short sections of the signal. Long sections give excellent frequency resolution, but fail to track frequency changes well. Shorter sections have poor frequency resolution, but good tracking. 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 will be undertaken in Chapter 9 and Lab 11, if you want to read ahead.

A beat note signal may be viewed as a single frequency signal whose amplitude varies with time, *or* as two signals with different constant frequencies. Both views will be useful in evaluating the effect of window length when finding the spectrogram of a beat signal.

(a) Create and plot a beat signal with

  (i) $f_\Delta = 32$ Hz
  (ii) $T_{\text{dur}} = 0.26$ sec
  (iii) $f_s = 11025$ Hz
  (iv) $f_0 = 2000$ Hz

(b) Find the spectrogram using a window length of 2048 using the commands:

```
specgram(x,2048,fsamp); colormap(1-gray(256)).
```

Comment on what you see.

(c) Find the spectrogram using a window length of 16 using the commands:

```
specgram(x,16,fsamp); colormap(1-gray(256)).
```

Comment on what you see, and compare to the previous spectrogram.

Lab #4
EE-2200
Fall-1998
Instructor VERIFICATION Sheet

Staple this page to the end of your Lab Report.

Name: ____ Date of Lab: __________

Part 3.1 Demonstrate Beat control GUI:

Verified:____________________ Date/Time:__________

Part 3.2(a) Explain chirp signal with hand-drawn sketch of the instantaneous frequency:

Verified:____________________ Date/Time:__________

Part 3.2(b) Cross-check the instantaneous frequency with a spectrogram plot:

Verified:____________________ Date/Time:__________