GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

EE 2025 Spring 2004
Lab #3: AM and FM Sinusoidal Signals

Date: 27-Jan – 2-Feb 2004

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.** You must complete the online Pre-lab exercise on Web-CT at the **beginning** of your scheduled lab session. You can use MATLAB and also consult your lab report or any notes you might have, but you cannot discuss the exercises with any other students. You will have approximately 20 minutes at the beginning of your lab session to complete the online Pre-Post-Lab exercise. The Pre-Post-Lab exercise for this lab includes some questions about concepts from the previous Lab report as well as questions on the Pre-Lab section of this lab.

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**. One of the laboratory instructors must verify the appropriate steps by signing on the **Instructor Verification** line. 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.

The exercises in Section 4 should written up in this week's lab report. More information on the lab report format can be found on Web-CT under the "Information" link. You should **label** the axes of your plots and include a title and Figure number for every plot. Every plot should be referenced by Figure number in your text discussion. In order to make it easy to find all the plots, include each plot *inlined* within your report. This can be done easily with MATLAB's `notebook` capability.

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 the submitted work should be original and it should be your own work.

The report will be **due during the period 3-Feb to 9-Feb 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 Pre-Lab

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

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

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.

2.1 Amplitude Modulation

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

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

where $A_k e^{j\phi_k}$ 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.

2.2 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 $(2\pi f_0 t + \phi)$ which is also the exponent of the complex exponential. We will refer to the argument of the cosine as the ***angle function***. In equation (1), the *angle function* changes *linearly* versus time, and its time derivative is $2\pi f_0$ which equals the constant frequency of the cosine.

CD-ROM *FM Synthesis*

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

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

The time derivative of the angle function $\psi(t)$ in (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 linear-FM *chirp* signal is a sinusoid whose 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 quadratic angle function 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 time $t = 0$ secs., then f_0 is also the starting frequency. The frequency variation produced by the time-varying angle function 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 MATLAB Synthesis of Chirp Signals

The following MATLAB code will synthesize a chirp:

```
fsamp = 8000;
dt = 1/fsamp;
dur = 1.1;
tt = 0 : dt : 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 do the following:

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

In the MATLAB code above, identify the values of A, μ, f_0, and ϕ. Write an expression that defines all the values of t_n.

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

(d) Listen to the signal to determine whether the signal's frequency content is increasing or decreasing (use `soundsc()`). Notice that `soundsc()` needs to know two things: the vector containing the signal sample, and the rate at which the signal samples were created.
For more information do `help sound` and `help soundsc`.

2.5 Concept Maps

The human brain processes information that it receives from various senses and tries to organize it and make meaning out of it. In the case of vision, the brain's task is simplified if information is presented in a graphical format rather than in a textual format. Concept maps are a method of visualizing the relationships among bits of information. In a concept map, information is presented in a very structured graphical format that is easy for the brain to interpret as useful information.

2.5.1 What are Concept Maps?

Concepts can be defined as a perceived regularity in events or objects, or records of events or objects, designated by a label. Propositions are statements about some object or event in the universe, either naturally occurring or constructed. Propositions contain two or more concepts connected with other words to form a meaningful statement.

Concept maps consist of various concepts that are linked together to generate propositions. A proposition is usually a semantic unit (a unit of meaning). A simple concept map is shown in Fig. 1. The concepts, "Mammals", "Dogs", "Cats", "Mice" and "Milk" are enclosed in circles or boxes. The linking words "Could be", "chase", "eat" and "drink" connect concepts together to create propositions. Thus Fig. 1 displays the following propositions:

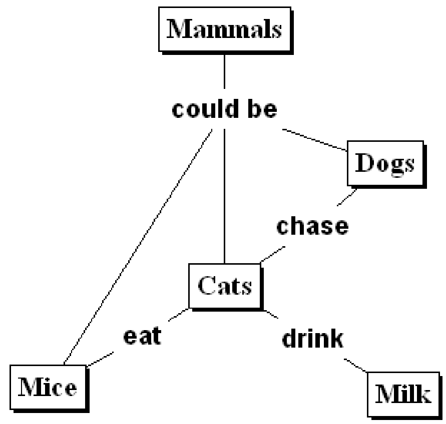

Figure 1: Sample Concept Map

- Mammals could be dogs.
- Mammals could be cats.
- Mammals could be mice.
- Dogs chase cats.
- Cats eat mice.
- Cats drink milk.

2.5.2 Creating Concept Maps

The following guidelines should be followed while generating concept maps.

- Concept maps should be created in a hierarchical fashion with the more general concepts at the top and the more specific concepts near the bottom.
- Cross-links can be used when concepts have multiple links.
- Linking words used should be precise.
- Concept maps should be created such that propositions can be understood independent of one another. An example of a bad map is shown in Fig. 2(a). In Fig. 2(a), the propositions "Wood for fire" and

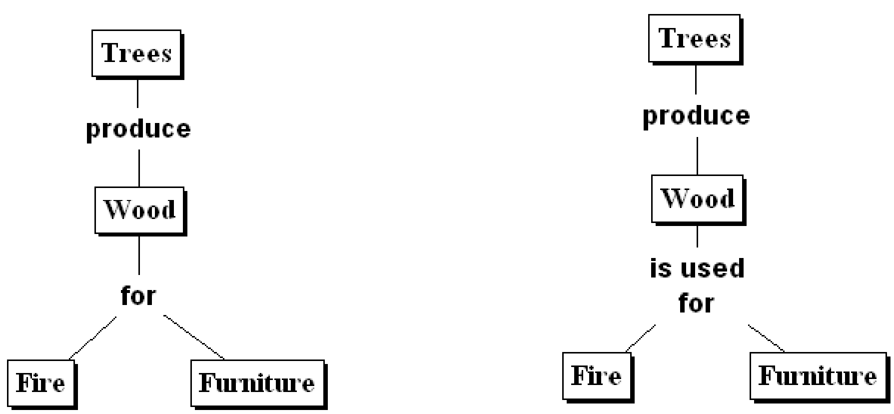

Figure 2: (a) Propositions that are not independent of one another. (b) Independent Propositions.

"Wood for furniture" make no sense by themselves. This map has been designed to be read as "Trees produce wood for fire" and "Trees produce wood for furniture". A better map is shown in Fig. 2(b).

- Avoid using sentences in boxes to represent concepts. Instead, break the sentence down to generate a new subsection in the map. The words that sit in a box should represent only one concept.
- Avoid string maps. These represent poor map construction skills and poor grasp of the material since there are no cross links. A sample string map is shown in Fig. 3.

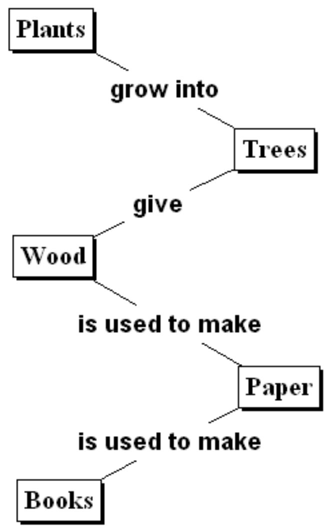

Figure 3: A String Map shows poor grasp of the concepts.

2.6 Concept Navigation Tool (CNT)

Concept maps in ECE-2025 can be created using the ***Concept Navigation Tool (CNT)*** Software.[1] This software allows the user to create concept maps with ease and when a concept map is complete, it can be exported to XML. This saved XML code can be opened in ***CNT*** for the purpose of editing or viewing the map. A screenshot of the ***Concept Navigation Tool*** is shown in Fig. 4.

When running the software, one can see three tabs at the top left. The `Workspace` is where concept maps can be created and modified. The `Preview` window displays the current map in a printer friendly format. `Options` allows one to set various options such as fonts, colors and meta information. At the top right of the workspace sits a toolbar. Various buttons in the toolbar allow one to open maps, create new maps, save maps, zoom in or out of maps, move maps and generate XML code. Using these functions one can easily create and modify concept maps. The options menu allows the user to change the appearance of the concept map as well as enter important meta information. ***CNT*** will not let you save your concept maps unless all the meta information has been entered. Nodes and links can be modified by changing their colors and their skins. Currently two skins are available; `Default` and `Clarity,` shown in Fig. 5. Both skins have their advantages and disadvantages. The Clarity skin shows just the concept name and thus the information carried by the node has a strong impact on the user. However, in order to modify any attributes of the node or link it to other nodes, the user needs to navigate through the menu by clicking the button to the top left of the node. The Default skin has shortcuts in the body of the node for modifying the concept name and linking the node to other nodes. Thus the Default skin should make it faster to construct the map.

[1]Authored by Chris Scheibe of Georgia Tech, using Macromedia Flash. Available from the link on the Web-CT lab page.

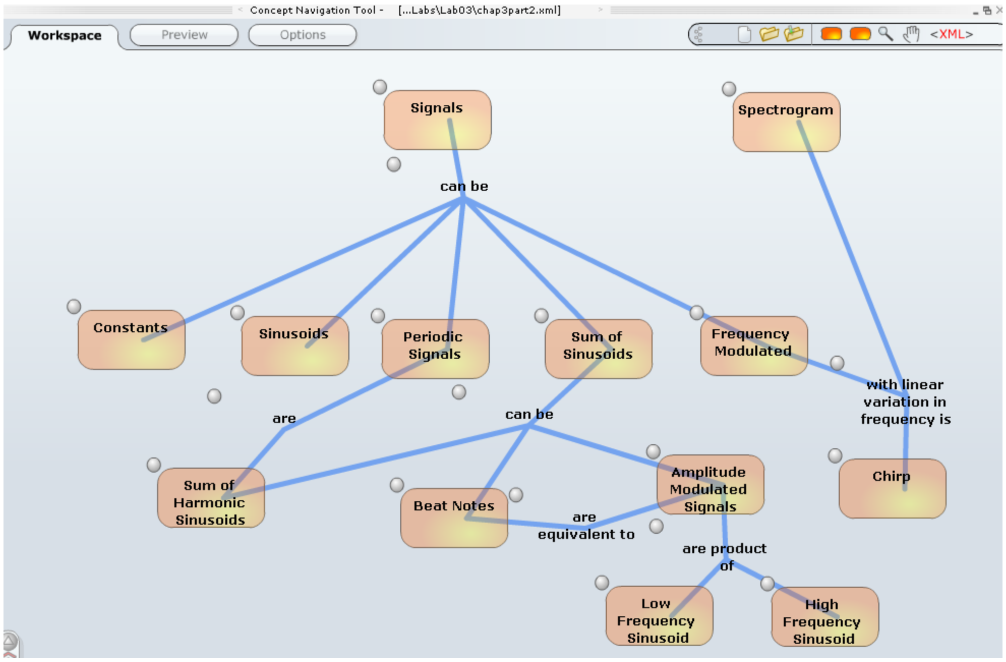

Figure 4: The ***Concept Navigation Tool (CNT)*** Software. Also, the map to be created in the warm-up.

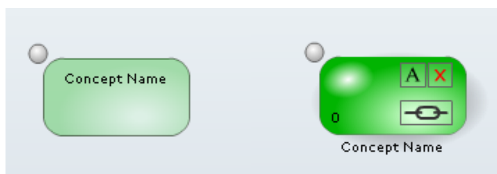

Figure 5: Node Skins: Clarity (Left) and Default (Right).

2.6.1 Adding Resources to Concepts

With ***CNT*** it is also possible to add various resources (such as pictures, movies, .pdf files, other concept maps, etc.) to each concept. The addition of resources to concept maps could become very tedious. The maps that we are creating are based on concepts in *SP First*. Thus, concepts throughout the book are interrelated and related resources appear in various locations throughout the book. It would be humanly impossible to be able to read through the whole book and add relevant resources to each concept. The ***Concept Navigation Tool*** solves this problem of adding resources. Each concept in a map has two fields associated with it. The first field holds the concept name that is displayed on the map in the viewer. The second field contains a list of keywords. Once a concept is created and keywords are entered, ***CNT*** searches through a database of resources (such as homework solutions, test solutions, lecture videos, etc.) and those resources that have matching keywords are automatically added to the concept. This eliminates the tedious process of manually adding resources. All the map creator has to do is include the relevant keywords in each node of the map, and then the resources will be automatically added.

3 Warm-up

3.1 Building a Concept Map

In this exercise, you will have to create the concept map in Fig. 4 using the ***CNT*** software. To run the software navigate to the directory where you downloaded the ***CNT*** application and "double-click" on `cnt.exe` The next few steps will guide you in creating the required Concept Map.

(a) First, we will set up some Options that control the way the Concept Map looks. To do this,

  (i) Click on the `Options` tab

  (ii) Click on the button that says `Nodes` to set the node attributes. In particular, choose the `Clarity` skin for Nodes.

  (iii) Click the button that says `Links` and similarly choose the `Clarity` skin for Links.

(b) Next step is to place the individual Concept (Nodes) and create appropriate links between them as shown in Fig. 4. To do this,

  (i) Click on the `Workspace` tab. To allow you to create a concept map, a toolbar is provided at the top right. The toolbar is shown in Fig. 6.

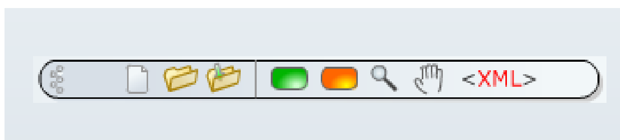

Figure 6: Toolbar.

  (ii) Placing the pointer over the little green box in the toolbar, you will see a bubble that reads `New Node...`

  (iii) Placing the pointer over the orange box will read `New Link...`

  (iv) Placing the pointer over the Lens will read `Zoom` and this can be used to Zoom in or out on the Concept Map once you have populated it with more concepts. These buttons will be used to add concept nodes and links to concept maps.

(c) To place a Node, click on the green box in the toolbar and drag the pointer to a point in the workspace. You should see a green block move with the pointer. If the node is the first concept node that you are going to add, then place it at the top center of the screen. Any node can be moved around by clicking on it and dragging it, holding the mouse button and dragging it to the desired location.

(d) The Round Grey button to the left of the Node (or Link) is the Menu button. Click on it to see available options. Use this to Delete or Move a Node and to add text as shown in the next step. Menu options are shown in Fig. 7.

(e) This first node will correspond to the *Signals* concept shown in Fig. 4. In order to add text to the node, click the menu button to the top left of the node. Clicking it again will close the menu. Choose `Keywords` from the menu. An `Input` window opens up. Under `Concept Name,` type in `Signals.` This is the text that will be displayed on the screen. The text that you enter under `Keywords` will be used to search through a database for relevant resources. This feature is not applicable for the current lab. You may leave this field as it is. Click `OK.` You should see the concept name *Signals* appear in the node.

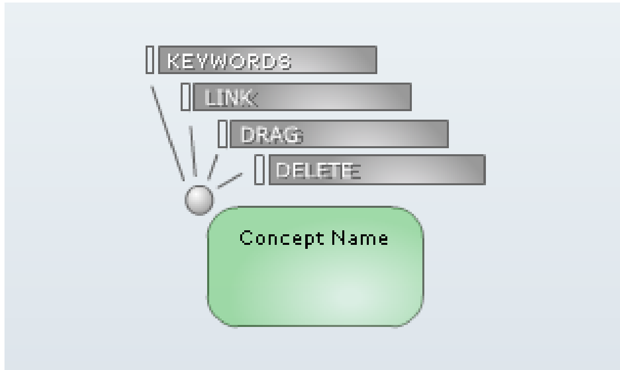

Figure 7: Menu.

(f) Similarly, add the concept *Sum of Sinusoids* and place it in the workspace corresponding to a location as shown in Fig. 4.

(g) Now you need to add a link between these concepts. In the toolbar, click on the orange box to add a new link. Place this link in between the two nodes.

(h) In order to add text, click the grey menu button to the top left of the link to open the menu and click `Association.` In the `Input` window that opens up, type `can be` under `Associations,` and click `OK.` Now you have two concepts and a linking term ready.

(i) You now need to physically connect these together using linking lines. In order to connect the concept *Signals* with the link `can be,` open the menu for the *Signals* node and click on `Link.` All candidates for linking will light up. Click on the link `can be.` You will see a line connecting the two together.

(j) In order to connect the link `can be` to the node *Sum of Sinusoids*, open the menu for the `can be` link and click `Link.` Once again, all candidates for linking will light up. Click on the *Sum of Sinusoids* node and you will see a line connecting the two.

(k) Once you have done this, you will have an incomplete Concept Map similar to Fig. 8.

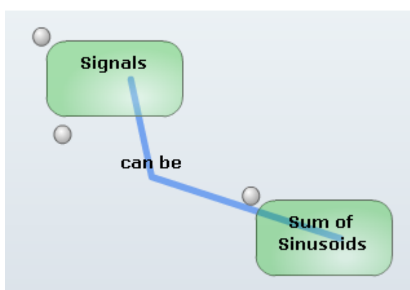

Figure 8: Incomplete Concept Map.

(l) Note: if you connect a line to the incorrect object, then the line can be removed by reconnecting the same two objects.

(m) Go back to the options menu and set the skin to Default for nodes and links. Click back on the Workspace. You will observe that the skins have changed. The default skin is more user friendly since all the menu options are displayed in the body of the nodes and links. However, using the Default skin would lead to the map being cluttered and the information in the map has a low impact. You may choose to continue working with whichever skin you prefer.

Now that you have created a small part of the concept map, populate the remaining portion of the concept map to create the entire map in Fig. 4. Use the zoom and scroll controls if the map gets larger than the screen display.

(n) When you are done, you need to save the concept map that you have created. The ***Concept Navigation Tool*** will not allow you to save unless all the meta information has been filled in.

   (i) Click on `Options` and click on the `Global` button.

   (ii) Under `Name`, enter your name. Under `Lecture`, enter `Concept Map Lab 1`. Under `Class`, enter `ECE2025`. Finally enter today's date.

   (iii) Now click on the `Workspace` tab and click on the `Save File...` button in the `Toolbar` (Fig. 6) to save the Concept Map that you have just created.

   (iv) Click `Preview` to display the map in a printer friendly format.

Show your map on the screen to the TA. **Instructor Verification** (separate page)

3.2 Function for a Chirp

Use the code provided in the warm-up as a starting point in order to write a MATLAB function that will synthesize a "chirp" signal according to the following template. Fill in code where you see `????`.

```
function   [xx,tt] = 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 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 = ????
psi = 2*pi*( f1*tt + ?????*tt.*tt);
xx = real( exp(j*psi) );
```

As a test case, generate a chirp sound whose frequency starts at 3200 Hz and ends at 800 Hz; its duration should be 1.25 sec and the sampling rate should be $f_s = 11025$ samples/sec. Listen to the chirp using the `soundsc` function. Give the exact calling sequence for `mychirp.m` in order to produce the test case.

Instructor Verification (separate page)

3.3 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 $\omega = 2\pi f_0$, the other at $\omega = -2\pi f_0$. For more complicated signals the spectra may be very interesting, as in the case of FM, where 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

When unsure about a command, use `help`.

There are a few important things to know about spectrograms:

1. In MATLAB the function `specgram` will compute the spectrogram. Type `help specgram` to learn more about this function and its arguments.

2. Spectrograms are numerical calculations 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. Another lab on the CD-ROM that accompanies the text treats this problem by using the spectrogram to extract the frequencies of piano notes.

3. A common call to the function is `specgram(xx,1024,fs).` The second argument[2] is the *window length* which could be varied to get different looking spectrograms. The spectrogram is able to "see" the separate spectrum lines with a longer window length, e.g., 1024 or 2048.[3]

4. If you are working at home, you might not have the `specgram()` function because it is part of the *Signal Processing Toolbox*. In that case, use the function `plotspec(xx,fs)` which is part of the *SP-First Toolbox* which can be downloaded from WebCT.

   • Note: The argument list for `plotspec()` has a different order from `specgram`, because `plotspec()` uses an optional third argument for the *window length* (default value is 256). In addition, `plotspec()` does not use color for the spectrogram; instead, darker shades of gray indicate larger values with black being the largest.

5. **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 complex-valued.
   Warning: This trick works nicely with the *SP-First* function called `plotspec`. However, when used with `specgram` it produces an image that does not have the negative frequency region in the proper location.

In order to see what the spectrogram produces, run the following code:

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

or, if you are using `plotspec(xx,fs)`:

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

[2]If the second argument is made equal to the "empty matrix" then its default value of 256 is used.

[3]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.

Notice that the spectrogram image contains one horizontal line at the correct frequency of the sinusoid. For 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
```

Instructor Verification (separate page)

4 Lab: Chirps and Beats

For the lab exercise and lab report, you will synthesize some AM and FM signals. In order to verify that they 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 Beat Note 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 appreciate 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 3.3, 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 frequency. However, shorter sections lack the ability to do accurate frequency measurement because the amount of input data is limited. On the other hand, long sections can give excellent frequency measurements, but fail to track sudden frequency changes well. For example, if a signal is the sum of two sinusoids whose frequencies are nearly the same, a long section length is needed to "resolve" the two sinusoidal 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 will be undertaken in the last chapter of *SP-First*.

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 defined via: $x(t) = \cos(1790\pi t)\sin(30\pi t)$, with a duration of 1.234 secs. Use a sampling rate of $f_s = 8000$ samples/sec to produce the signal in MATLAB. Use `xt` as the name of the MATLAB vector for the signal.

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

(c) The signal defined in part (a) is periodic; determine its period from the mathematical theory of harmonic spectra.

(d) Plot the spectrogram of $x(t)$ using a window length of 1024 using the commands[4]:
```
plotspec(xt,fs,1024); grid on
```
Comment on what you see. Are the correct frequencies present in the spectrogram? If necessary, use the zoom tool (in the MATLAB figure window) to examine the important region of the spectrogram.

(e) Plot the spectrogram of $x(t)$ using a window length of 128 using the commands:
```
plotspec(xt,fs,128); grid on.
```

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

Comment on the dark and light bands that you see, and determine what property of $x(t)$ is causing them. In addition, compare to the previous spectrogram.

(f) Which spectrogram is correct? In other words, describe what the true spectrogram should look like.

4.2 Spectrogram of a Chirp

Use the `mychirp` function (written during the Warm-up) to synthesize a "chirp" signal for your lab report. Use the following parameters:

1. A total time duration of 2.718 secs. with a sampling rate of $f_s = 11025$ Hz.
2. The instantaneous frequency starts at 5400 Hz and ends at 600 Hz.

Listen to the signal. What comments can you make regarding the sound of the chirp (e.g., is the frequency movement 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. In addition, give the mathematical formula for the chirp and its instantaneous frequency.

4.3 A Chirp Puzzle

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

1. A total time duration of 2.718 secs. with a sampling rate of $f_s = 11025$ Hz.
2. The instantaneous frequency starts at -1500 Hz (negative frequency), and ends at $+4000$ Hz.

Listen to the signal. Does it chirp down, or chirp up, or both? Give the mathematical formula for the chirp and its instantaneous frequency. In addition, create a spectrogram of this second chirp signal. Use the theory of the spectrum (with its positive and negative frequency components) to help explain the connection between what you hear and what you see in the spectrogram. Observe the changing instantaneous frequency in the spectrogram which implies that the frequency components in the spectrum are moving.
Hint: In order to create a **spectrogram with negative frequencies**, see the instructions in Section 3.3.

4.4 Interpreting a Concept Map

Based on the concept map that you created during the warm-up, answer the following questions:

(a) How are periodic signals characterized?

(b) Are beat signals different from amplitude-modulated (AM) signals?

(c) How are chirps characterized?

Include your answers to this section in your lab report.

Lab #3
ECE-2025
Spring-2004
INSTRUCTOR VERIFICATION SHEET

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

Name: ____ Date of Lab: __________

Part 3.1 Show your completed concept map to your TA.

Verified:____________________ Date/Time:__________

Part 3.2 Demonstrate the `mychirp.m` function. In the space below write how you would call the function with a correct set of arguments.

Verified:____________________ Date/Time:__________

Part 3.3 Demonstrate a spectrogram of a sinusoid.

Verified:____________________ Date/Time:__________