

Lab S-7: Spectrograms of AM and FM Signals

Pre-Lab: Read the Pre-Lab and do all the exercises in the Pre-Lab section *prior to attending lab*.

Verification: The Exercise section of each lab should be completed **during your assigned Lab time** and the steps marked *Instructor Verification* signed off **during the lab time**. When you have completed a step that requires verification, demonstrate the result to your instructor and answer any questions about it. Turn in the completed verification sheet before you leave the lab.

Lab Homework Questions: The Lab-Homework Sheet has a few lab related questions that can be answered at your own pace. The completed Lab-HW sheet is due at the beginning of the next lab.

1 Pre-Lab

1.1 Objective

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 companion website that provide examples of these signals for many different conditions, e.g., *FM Synthesis*. The resulting signal can be analyzed to show its time-frequency behavior by using the *spectrogram*.

This lab studies signal synthesis for AM and FM signals, and their time-frequency content as shown in a spectrogram. 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:

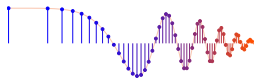
1. Synthesize a beat-note signal with a MATLAB M-file, and display its spectrogram.
2. Study the frequency resolution of the spectrogram for two closely spaced sinusoids.
3. *Spectrogram:* Make empirical observations of the spectrogram as the section length is changed.
4. Synthesize a linear-FM chirp 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.

1.2 Overview

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

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

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, i.e., frequency-modulated sinusoids.



1.3 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 as a summation of terms with complex amplitudes via:

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

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.

1.3.1 Beat Control GUI

To assist you in your experiments with beat notes and AM signals, the MATLAB GUI tool called **beatcon** has been created. The M-file `beatcon.m` is part of the *DSP-First* (or *SP-First*) Toolbox. 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. It 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 to hear the *beat frequency* sound.

1.4 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 define 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.

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

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

where $\psi(t)$ is the angle function.¹ The time derivative of the angle function $\psi(t)$ in (3) gives a frequency that we call the *instantaneous (radian) frequency*:

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

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

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

1.5 Chirp, or Linearly Swept Frequency

A linear-FM *chirp* signal is a sinusoid whose instantaneous frequency changes linearly from a starting value to an ending one.² 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 $\psi(t)$ in (3) as

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

¹Look for the demo *FM Synthesis* on the companion website.

²Look for the demo *Spectrograms & Sounds: Wideband FM* on the companion website.



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

$$f_i(t) = 2\mu t + f_0 \quad (\text{hertz}) \quad (5)$$

The slope of $f_i(t)$ is equal to 2μ and its $t = 0$ intercept is f_0 . The frequency variation in (5) 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 (5) 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 (5) will be

$$\text{SLOPE} = 2\mu = \frac{f_2 - f_1}{t_2 - t_1} \quad (6)$$

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

1.6 MATLAB Synthesis of Chirp Signals

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
```

- 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.
- 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.
- 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 call 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.

1.7 Spectrogram of an FM Signal: Sinusoidal Modulation

Define an FM signal whose instantaneous frequency is sinusoidal, i.e.,

$$\omega_i(t) = 2\pi f_c + 2\pi\alpha \cos(2\pi\beta t + \gamma) \quad \text{rad/s} \quad (7)$$

where f_c is the center frequency, and the parameters α , β and γ control the sinusoidal frequency modulation.

- (a) Determine the mathematical formula for an FM signal that has the instantaneous frequency in (7).
- (b) Write a MATLAB function (or script) to create sinusoidal FM signals of the form defined in (7). Modify the code in Sect. 1.6 to use the parameters in (7). If you choose to make a function, the MATLAB function should use the following template:
- ```
makeSinusFMvals(alpha, beta, gamma, fc, fSamp, tStart, tStop).
```
- (c) Create a sinusoidal-FM signal with  $f_c = 100$  Hz,  $\alpha = 50$ ,  $\beta = 1.5$ , and  $\gamma = \pi/3$ . Make the signal duration equal to 3.04 secs, starting at  $t = 0$ . Use a sampling rate of 1000 samples/s. The signal amplitude should be one.
- (d) Create a spectrogram of this chirp signal, and use it to verify that you have the correct instantaneous frequency predicted by (7). The section length should be *short enough* to track the changing instantaneous frequency.

## 1.8 Review Topic: Spectrograms

The main issue in this lab will be the dependence of the spectrogram on the choice of *section length*. 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.

For more information refer to the write-up in a previous lab.



## 2 Lab Exercise

For the lab exercise, you will synthesize some AM and FM signals, and then verify that these signals have the correct frequency content by using the spectrogram. The objective is to learn enough to be able to discuss 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).

### 2.1 Beat Notes and Frequency Resolution

In the section on beat notes in Chapter 3 of the text, we discussed signals formed as the product of two sinusoidal signals of slightly different frequencies; i.e.,

$$x(t) = B \cos(2\pi f_{\Delta} t + \varphi_{\Delta}) \cos(2\pi f_c t + \varphi_c) \quad (8)$$

where  $f_c$  is the (high) center frequency, and  $f_{\Delta}$  is the (low) frequency that modulates the envelope of the signal. An equivalent representation for the beat signal is obtained by rewriting the product as a sum:

$$x(t) = A_1 \cos(2\pi f_1 t + \varphi_1) + A_2 \cos(2\pi f_2 t + \varphi_2) \quad (9)$$

It is relatively easy to derive the relationship between the frequencies  $\{f_1, f_2\}$  and  $\{f_c, f_{\Delta}\}$ .

#### 2.1.1 MATLAB Code for Beat Signals

A beat signal is defined by five parameters  $\{B, f_c, f_{\Delta}, \varphi_c, \varphi_{\Delta}\}$  along with the start and end times and the sampling rate ( $f_s$ ), as shown in the following template:

```
Amp = 10; %-- B in equation above
fc = 1024; %-- center frequency
phic = 2*pi*rand; %-- phase of 2nd sinusoid (random)
fDelta = 4; %-- modulating frequency
phiDelta = 2*pi*rand; %-- phase of 1st sinusoid (random)
tStart = 0; %-- starting time (secs)
tStop = 5; %-- ending time (secs)
fSamp = 8000;
%
tt = tStart:(1/fSamp):tStop; %-- vector of times
xx = Amp*cos(2*pi*fc*tt+phic).*cos(2*pi*fDelta*tt+phiDelta)
```

#### 2.1.2 Beat Note Spectrograms

Beat notes have a simple time-frequency characteristic in a spectrogram. Even though a beat note signal, when defined as a product in (8), may be viewed as a single frequency signal whose amplitude varies with time, *the spectrum requires an additive combination* as in (9) which turns out to be the sum of two sinusoids with different constant frequencies.

Beat notes 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.<sup>3</sup> 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

<sup>3</sup>The section length is often called the window length; the two terms are used interchangeably in DSP.



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 as distinct lines in the spectrogram if  $|f_1 - f_2| \approx C/T_{\text{SECT}}$  where  $C$  is a proportionality constant and  $T_{\text{SECT}}$  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 Section 2.1.1 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/s. The frequency  $f_\Delta$  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<sup>4</sup>:

```
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.<sup>5</sup> Then reduce the value of  $L_{\text{SECT}}$  little by little to get the smallest  $L_{\text{SECT}}$  that will work. Getting a value of  $L_{\text{SECT}}$  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  $L_{\text{SECT}}$  and determine whether the frequencies present in the spectrogram are correct. In addition, convert  $L_{\text{SECT}}$  to the section duration in seconds,  $T_{\text{SECT}}$ .

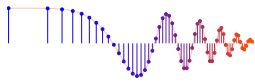
**Instructor Verification** (separate page)

### 2.1.3 Inverse Relationship: Section Length vs. Frequency Resolution

The shortest section length when you are able to discern the two spectrum lines was determined in the previous section.

<sup>4</sup>Use `plotspec` instead of `specgram` in order to get a linear amplitude scale rather than logarithmic.

<sup>5</sup>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.



- (a) The shortest (*window*) *section length* of  $L_{\text{SECT}}$  samples has been converted into a (window) section duration in seconds (via the sampling rate). Compare the inverse of this (window) section duration to the frequency separation of the spectrum lines.

$$|f_1 - f_2| \leftarrow ? \rightarrow \frac{1}{T_{\text{SECT}}}$$

If we believe that the inverse relationship between (window) section duration and frequency separation is true, then we can calculate a constant  $C$  such that

$$|f_1 - f_2| = \frac{C}{T_{\text{SECT}}} \quad (10)$$

From your first experiment, determine  $C$ .

- (b) Now change  $f_{\Delta}$  to 16 Hz and repeat the resolution experiment in Sect. 2.1.2(d). That is, find a section length that will resolve the two frequency components which are now farther apart. Use the value of  $C$  and (10) to predict the section length that you will need. Verify that this section length will work correctly.

*Note:* The relationship is approximate, so the derived section length is not guaranteed to work. If that happens, a small increase in  $L_{\text{SECT}}$  should make it work.

**Instructor Verification** (separate page)

## 2.2 Spectrogram for a Chirp with Negative Instantaneous Frequency

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.

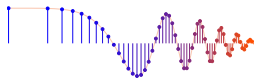
- (a) What happens when we make a signal that “chirps down” and the *instantaneous frequency goes negative*? Generate a chirp signal that starts at 2000 Hz when  $t = 0$  s, and chirps down to  $-1000$  Hz, at  $t = 1.5$  s. Use  $f_s = 8000$  Hz. Determine the parameters needed in (4).
- (b) Generate the chirp signal in MATLAB and make a spectrogram with  $L_{\text{SECT}} = 200$  to verify that you have the correct starting and ending frequencies. For  $L_{\text{SECT}} = 200$ , determine the section duration  $T_{\text{SECT}}$  in secs.
- (c) Will you hear negative frequency? Use a spectrogram that contains negative frequencies to explain your answer. If possible listen to the signal and describe what you will hear; then explain in terms of the two-sided spectrogram.

**Instructor Verification** (separate page)

### 2.2.1 Section Length in Chirp Spectrogram

When we have a signal whose frequency is not constant, we can study how the temporal features of the spectrogram depend on the section length  $L_{\text{SECT}}$ . As we saw in the previous section, the section length has to be short to capture the temporal changes in the signal. However, let’s see what happens with a long section length.

- (a) Generate the same signal as in Sect. 2.2, but make the spectrogram with  $L_{\text{SECT}} = 1600$ .
- (b) When  $L_{\text{SECT}} = 1600$ , determine the section duration  $T_{\text{SECT}}$  in secs.



- (c) The spectrogram uses 50% overlapping and skipping (see Sect. 1.8). Based on  $T_{\text{SECT}}$ , determine the time locations where the spectrum is being computed. Relate these time locations to what you see in the spectrogram.

**Instructor Verification** (separate page)

- (d) *Optional:* Use the slope of the instantaneous frequency to determine how much the frequency changes during one section. Calculate the frequency change from  $T_{\text{SECT}}$  and  $\mu$ . In this spectrogram, the changing frequency appears as rectangular bars that have a measurable width and height. The width should be  $0.5T_{\text{SECT}}$  in secs. Compare the calculated frequency change to the vertical height of the bars—are they different or equal?

### 2.3 Lab-HW: Matching Unknown Spectrograms

Now you are given a spectrogram in Fig. 1, and you must synthesize a signal that will match that time-frequency plot. Explain and discuss your work.

1. Define a time signal  $x(t)$  whose spectrogram will match the given spectrogram. This signal definition should be a simple mathematical formula.  
Note: you might have to iterate with the following two steps to get a good approximation.
2. Generate samples of the signal over the appropriate time interval using  $f_s = 4000$  Hz.
3. Choose the section length in `plotspec` carefully so that your spectrogram is an excellent match.

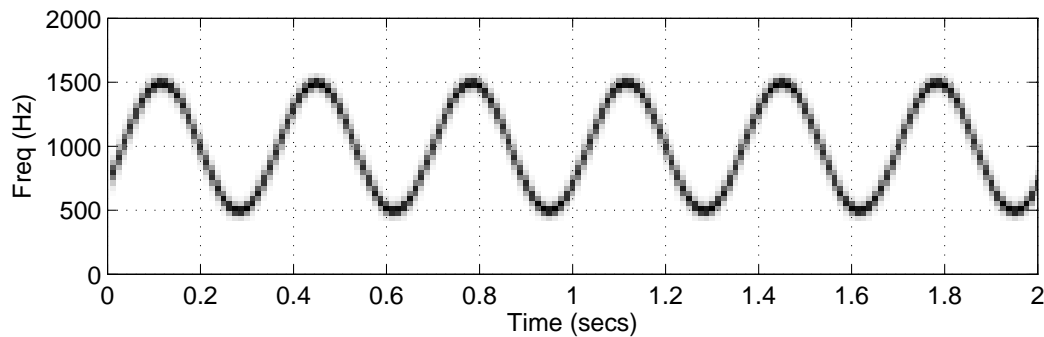
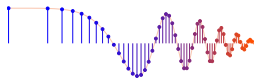


Figure 1: Spectrogram of unknown signal with  $f_s = 4000$  Hz. Section length to be determined. Horizontal axis is time in seconds.





**Lab: Spectrograms of AM and FM Signals**  
**INSTRUCTOR VERIFICATION SHEET**

Turn this page in to your lab grading TA before the end of your scheduled Lab time.

Name: \_\_\_\_\_ LoginUserName: \_\_\_\_\_ Date: \_\_\_\_\_

Part 2.1.2 Record the value of the section length  $L_{SECT}$  (in samples) and  $T_{SECT}$  (in secs) when you can discern two separate spectral lines for the beat note signal, using  $f = 4$  Hz.

$L_{SECT} =$

$T_{SECT} =$

Verified: \_\_\_\_\_ Date/Time: \_\_\_\_\_

Part 2.1.3 Calculate the proportionality constant  $C$  for the inverse relationship:  $|f_1 - f_2| \approx C/T_{SECT}$ . Then determine a new section length  $L_{SECT}$  for  $f = 16$  Hz. Synthesize the signal and make its spectrogram with the new section length  $L_{SECT}$ . Then verify that the two spectrum lines are resolved.

$C =$

$L_{SECT} =$

Verified: \_\_\_\_\_ Date/Time: \_\_\_\_\_

Part 2.2 Write MATLAB code for synthesizing a linear-FM chirp whose instantaneous frequency goes negative. Also, display the *two-sided* spectrogram that includes the negative frequency region, as well as the one-sided spectrogram that has positive frequency components only. Determine the section duration  $T_{SECT}$  in secs.

$T_{SECT} =$

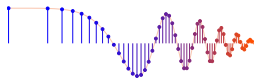
Verified: \_\_\_\_\_ Date/Time: \_\_\_\_\_

Part 2.2.1 Spectrogram of Chirp with long-duration section,  $L_{SECT} = 1600$ . Explain features in the spectrogram, e.g., location of sections along the time axis and duration of sections.

$T_{SECT} =$

Section times (centers) =

Verified: \_\_\_\_\_ Date/Time: \_\_\_\_\_



---

## Lab: Spectrograms of AM and FM Signals

### LAB HOMEWORK QUESTION

Turn this page in to your lab grading TA at the very beginning of your next scheduled Lab time.

Name: \_\_\_\_\_ LoginUserName: \_\_\_\_\_ Date: \_\_\_\_\_

#### Part 2.3

1. Match the unknown spectrogram. Give the mathematical formula for the signal
2. Include the MATLAB code for generating the signal.
3. Choose a section length to get the desired spectrogram. Include a plot of the spectrogram when you hand in your Lab-HW.