C.7 Spectrum

The purpose of this lab is to learn to examine the frequency domain content of signals. Two methods will be used. The first method will be to plot the discrete Fourier series coefficients of finite signals. The second will be to plot the Fourier series coefficients of finite segments of time-varying signals, creating what is known as a **spectrogram**.

C.7.1 Background

A finite discrete-time signal with p samples has a discrete-time Fourier series expansion

$$x(n) = A_0 + \sum_{k=1}^{(p-1)/2} A_k \cos(k\omega_0 n + \phi_k) \tag{C.6}$$

for p odd and

$$x(n) = A_0 + \sum_{k=1}^{p/2} A_k \cos(k\omega_0 n + \phi_k) \tag{C.7}$$

for p even, where $\omega_0 = 2\pi/p$.

A finite signal can be considered to be one cycle of a periodic signal with fundamental frequency ω_0, in units of radians per sample, or $1/p$ in Hertz. In this lab, we will assume p is always even, and we will plot the magnitude of each of the frequency components, $|A_0|, \cdots, |A_{p/2}|$ for each of several signals, in order to gain intuition about the meaning of these coefficients.

Notice that each $|A_k|$ gives the amplitude of the sinusoidal component of the signal at frequency $k\omega_0 = k2\pi/p$, which has units of radians per sample. In order to interpret these coefficients, you will probably want to convert these units to Hertz. If the sampling frequency is f_s samples per second, then you can do the conversion as follows (see box on page203):

$$\frac{(k2\pi/p)[\text{radians/sample}]f_s[\text{samples/second}]}{2\pi[\text{radians/cycle}]} = kf_s/p[\text{cycles/second}]$$

Thus, each $|A_k|$ gives the amplitude of the sinusoidal component of the signal at frequency kf_s/p Hz.

Note that Matlab does not have any built-in function that directly computes the discrete Fourier series coefficients. However, it does have a realization of the fast Fourier transform, a function called `fft`, which can be used to construct the Fourier series coefficients. In particular, `fourierSeries` is a function that returns the DFS coefficients[2]:

```
function [magnitude, phase] = fourierSeries(x)
% FOURIERSERIES - Return the magnitude and phase of each
% sinusoidal component in the Fourier series expansion for
```

[2] This function can be found at http://www.eecs.berkeley.edu/ eal/eecs20/matlab/fourierSeries.m.

```
% the argument, which is interpreted as one cycle of a
% periodic signal.  The argument is assumed to have an
% even number p of samples. The first returned value is an
% array containing the amplitudes of the sinusoidal
% components in the Fourier series expansion, with
% frequencies 0, 1/p, 2/p, ... 1/2. The second returned
% value is an array of phases for the sinusoidal
% components. Both returned values are arrays with length
% (p/2)+1.
p = length(x);
f = fft(x)/p;
magnitude(1) = abs(f(1));
upper = p/2;
magnitude(2:upper) = 2*abs(f(2:upper));
magnitude(upper+1) = abs(f(upper+1));
phase(1) = angle(f(1));
phase(2:upper) = angle(f(2:upper));
phase(upper+1) = angle(f(upper+1));
```

In particular, if you have an array `x` with even length,

```
[A, phi] = fourierSeries(x);
```

returns the DFS coefficients in a pair of vectors.

To plot the magnitudes of the Fourier series coefficients vs. frequency, you can simply say

```
p = length(x);
frequencies = [0:fs/p:fs/2];
plot(frequencies, A);
xlabel('frequency in Hertz');
ylabel('amplitude');
```

where `fs` has been set to the sampling frequency (in samples per second). The line

```
frequencies = [0:fs/p:fs/2];
```

bears further examination. It produces a vector with the same length as `A`, namely $1 + p/2$, where p is the length of the `x` vector. The elements of the vector are the frequencies in Hertz of each Fourier series component.

C.7.2 In-lab section

1. To get started, consider the signal generated by

```
t = [0:1/8000:1-1/8000];
x = sin(2*pi*800*t);
```

This is 8000 samples of an 800 Hz sinusoid sampled at 8 kHz. Listen to it. Use the `fourierSeries` function as described above to plot the magnitude of its discrete Fourier series coefficients. Explain the plot.

Consider the continuous-time sinusoid

$$x(t) = \sin(2\pi 800 t).$$

The `x` vector calculated above is 8000 samples of this sinusoid taken at a sample rate of 8 kHz. Notice that the frequency of the sinusoid is the derivative of the argument to the sine function,

$$\omega = \frac{d}{dt} 2\pi 800 t = 2\pi 800$$

in units of radians per second. This fact will be useful below when looking at more interesting signals.

2. With `t` as above, consider the more interesting waveform generated by

```
y = sin(2*pi*800*(t.*t));
```

This is called a **chirp**. Listen to it. Plot its Fourier series coefficients using the `fourierSeries` function as described above.

This chirp is 8 kHz samples of the continuous-time waveform

$$y(t) = \sin(2\pi 800 t^2).$$

The **instantaneous frequency** of this waveform is defined to be the derivative of the argument to the sine function,

$$\omega(t) = \frac{d}{dt} 2\pi 800 t^2 = 4\pi 800 t.$$

For the given values `t` used to compute samples `y`, what is the range of instantaneous frequencies? Explain how this corresponds with the plot of the Fourier series coefficients, and how it corresponds with what you hear.

3. The Fourier series coefficients computed in part 2 describe the range of instantaneous frequencies of the chirp pretty well, but they do not describe the dynamics very well. Plot the Fourier series coefficients for the waveform given by

```
z = y(8000:-1:1);
```

Listen to this sound. Does it sound the same as `y`? Does its Fourier series plot look the same? Why?

4. The chirp signal has a dynamically varying frequency-domain structure. More precisely, there are certain properties of the signal that change slowly enough that our ears detect them as a change in the frequency structure of the signal rather than as part of the frequency structure (the timbre or tonal content). Recall that our ears do not hear sounds below about 30 Hz. Instead, the human brain hears changes below 30 Hz as variations in the nature of the sound rather than as frequency domain content. The Fourier series methods used above fail to reflect this psychoacoustic phenomenon.

   A simple fix is the **short-time Fourier series**. The chirp signals above have 8000 samples, lasting one second. But since we don't hear variations below 30 Hz as frequency content, it probably makes sense to reanalyze the chirp signal for frequency content 30 times in the one second. This can be done using the following function:[3]

```
function waterfallSpectrogram(s, fs, sizeofspectra, numofspectra)

% WATERFALLSPECTROGRAM - Display a 3-D plot of a spectrogram
% of the signal s.
%
% Arguments:
%   s - The signal.
%   fs - The sampling frequency (in samples per second).
%   sizeofspectra - The number of samples to use to calculate each
%          spectrum.
%   numofspectra - The number of spectra to calculate.

frequencies = [0:fs/sizeofspectra:fs/2];
offset = floor((length(s)-sizeofspectra)/numofspectra);
for i=0:(numofspectra-1)
    start = i*offset;
    [A, phi] = fourierSeries(s((1+start):(start+sizeofspectra)));
    magnitude(:,(i+1)) = A';
end
waterfall(frequencies, 0:(numofspectra-1), magnitude');
xlabel('frequency');
ylabel('time');
zlabel('magnitude');
```

   To invoke this function on the chirp, do

```
t = [0:1/8000:1-1/8000];
y = sin(2*pi*800*(t.*t));
waterfallSpectrogram(y, 8000, 400, 30);
```

   which yields the plot shown in figure C.6. That plot shows 30 distinct sets of Fourier series coefficients, each calculated using 400 of the 8000 available samples. Explain how this plot describes the sound you hear. Create a similar plot for the reverse chirp, signal `z` given in part 3.

[3] This code can be found at http://www.eecs.berkeley.edu/ eal/eecs20/matlab/waterfallSpectrogram.m.

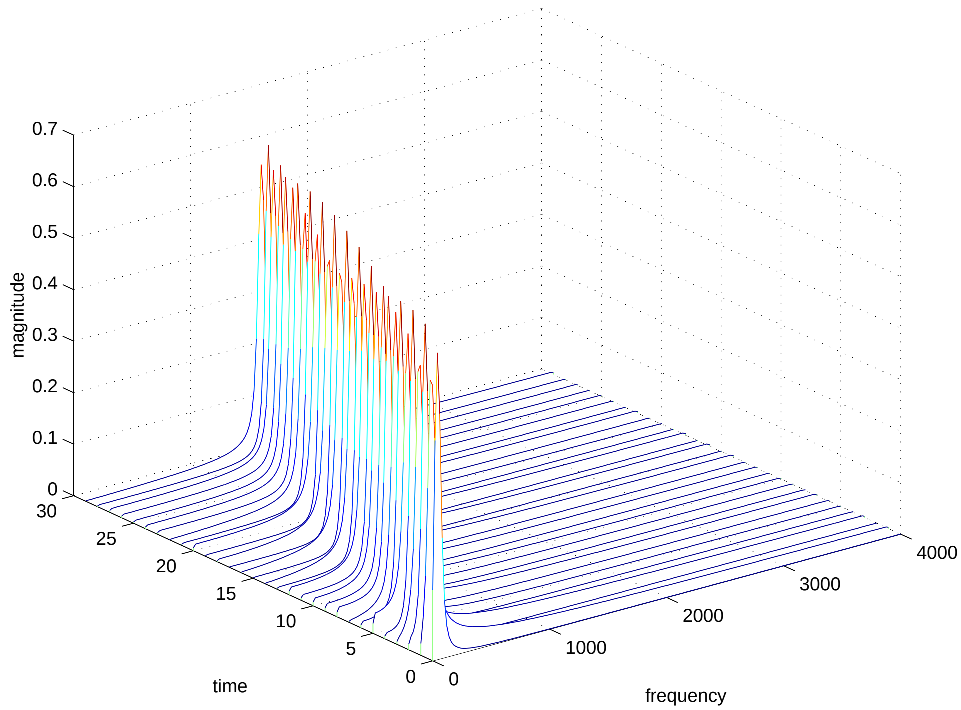

Figure C.6: Time varying discrete Fourier series analysis of a chirp.

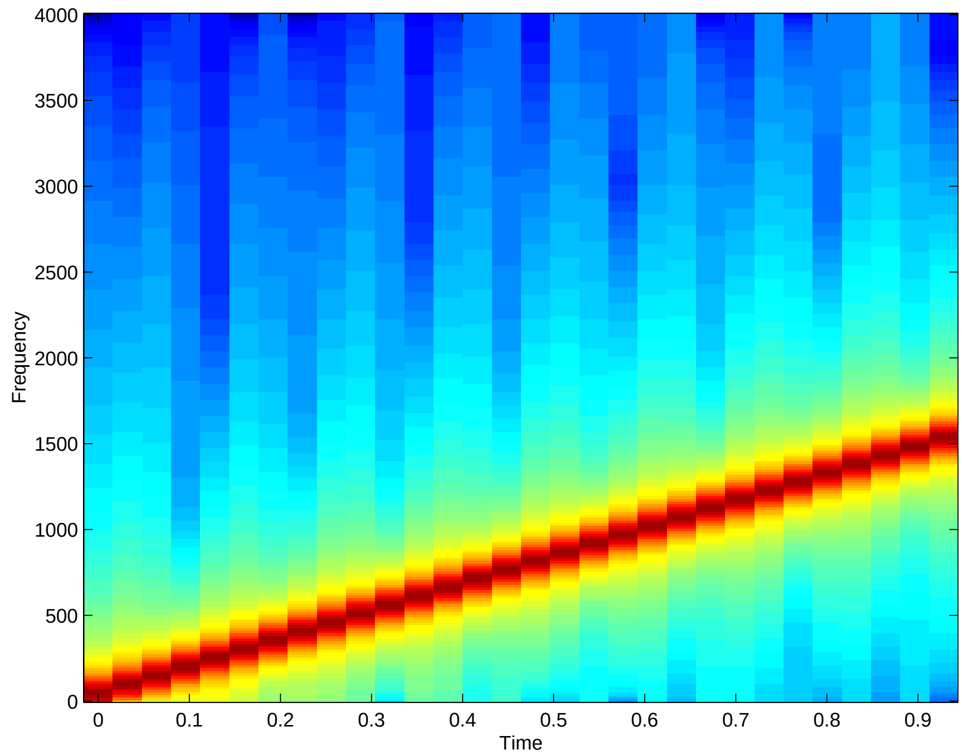

Figure C.7: Spectrogram of the chirp signal.

5. Figure C.6 is reasonably easy to interpret because of the relatively simple structure of the chirp signal. More interesting signals, however, become very hard to view this way. An alternative visualization of the frequency content of such signals is the **spectrogram**. A spectrogram is a plot like that in figure C.6, but looking straight down from above the plot. The height of each point is depicted by a color (or intensity, in a gray-scale image) rather than by height. You can generate a spectrogram of the chirp as follows:

```
specgram(y,512,8000);
```

This results in the image shown in figure C.7. There, the default colormap is used, which is `jet`. A rendition of this colormap is given in figure C.3. You could experiment with different colormaps for rendering this spectrogram by using the `colormap` command. A particularly useful one is `hot`, obtained by the command

```
colormap(hot);
```

Create a similar image for the reverse chirp, `z`, of part 3.

6. A number of audio files are available at

```
http://www.eecs.berkeley.edu/~eal/eecs20/sounds
```

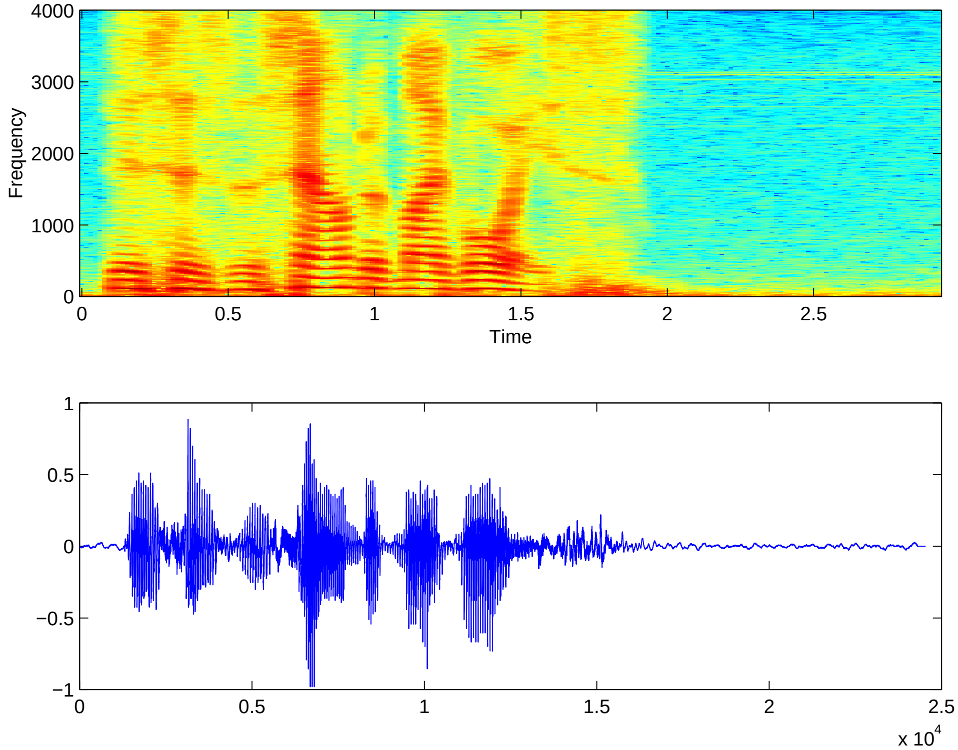

Figure C.8: Spectrogram and plot of a voice segment (one of the authors saying "this is the sound of my voice."

In Netscape, you can save these to your local computer disk by placing the mouse on the file name, clicking with the right mouse button, and selecting "Save Link As." For example, if you save `voice.au` to your current working directory, then in Matlab you can do

```
y = auread('voice.au');
soundsc(y)
subplot(2,1,1); specgram(y,1024,8000,[],900)
subplot(2,1,2); plot(y)
```

to get the result shown in figure C.8. Use this technique to get similar results for other sound files in the same directory. Interpret the results.

C.7.3 Independent section

1. For the chirp signal as above,

```
y = sin(2*pi*800*(t.*t));
```

generate the discrete Fourier series coefficients using `fourierSeries` as explained in section C.7.1. The, write a Matlab function that uses (C.7) to reconstruct the original signal from the coefficients. Your Matlab function should begin as follows:

```
function x = reconstruct(magnitude, phase)
% RECONSTRUCT - Given a vector of magnitudes and a vector
% of phases, construct a signal that has these magnitudes
% and phases as its discrete Fourier series coefficients.
% The arguments are assumed to have odd length, p/2 + 1,
% and the returned vector will have length p.
```

Note that this function will require a large number of computations. If your computer is not up to the task, the construct the Fourier series coefficients for the first 1000 samples instead of all 8000, and reconstruct the original from those coefficients. To check that the reconstruction works, subtract your reconstructed signal from `y` and examine the difference. The difference will not be perfectly zero, but it should be very small compared to the original signal. Plot the difference signal.

2. In the remainder of this lab, we will study **beat signals**, which are combinations of sinusoidal signals with closely spaced frequencies. First, we need to develop some background.

   Use Euler's relation to show that

   $$2\cos(\omega_c t)\cos(\omega_\Delta t) = \cos((\omega_c + \omega_\Delta)t) + \cos((\omega_c - \omega_\Delta)t).$$

   for any ω_c, ω_Δ, and t in *Reals*. **Hint**: See box on page 243.

   A consequence of this identity is that if two sinusoidal signals with different frequencies, ω_c and ω_Δ, are multiplied together, the result is the same as if two sinusoids with two other frequencies, $\omega_c + \omega_\Delta$ and $\omega_c - \omega_\Delta$, are added together.

3. Construct the sum of two cosine waves with frequencies of 790 and 810 Hz. Assume the sample rate is 8 kHz, and construct a vector in Matlab with 8000 samples. Listen to it. Describe what you hear. Plot the first 800 samples (1/10 second). Explain how the plot illustrates what you hear. Explain how the identity in part 2 explains the plot.

4. What is the period of the waveform in part 3? What is the fundamental frequency for its Fourier series expansion? Plot its discrete Fourier series coefficients (the magnitude only) using `fourierSeries`. Plot its spectrogram using `specgram`. Choose the parameters of `specgram` so that the warble is clearly visible. Which of these two plots best reflects perception?

Instructor Verification Sheet for C.7

Name: ______________________________ Date: ______________________________

1. Plot of the DFS coefficients of the sinusoid, with explanation.

   Instructor verification: ______________________________

2. Plot of the DFS, plus range of instantaneous frequencies, plus correspondence with the sound.

   Instructor verification: ______________________________

3. Plot of the DFS is the same, yet the sound is different. Explanation.

   Instructor verification: ______________________________

4. Explain how figure C.6 describes the sound you hear. Plot the reverse chirp.

   Instructor verification: ______________________________

5. Create and interpret a spectrogram for one other sound file, at least.

   Instructor verification: ______________________________