Hints for Mini-Project #1 version 2.0

Prof. Brian L. Evans
EE 313 Linear Systems and Signals, The University of Texas at Austin, Fall 2018

Organization and Content of the Report

When writing the mini-project report, please consider the audience of your report to be the other students in our EE 313 course.

Please organize your writeup for the mini-project into the following sections:

1. Introduction -- explain in your own words and and with appropriate references the larger topic of synthesizing music. Build on your experience on the sound quality of notes generated as single-frequency tones using sinusoidal signals. Explain example audio effects that you have already seen using amplitude modulation and frequency modulation (e.g. chirp signals). You can also use ideas from the Introduction section in the lab assignment. Probably half of a page for this section.
2. Overview -- explain in your own words and with appropriate references the general idea of frequency modulation sound synthesis, including mathematical models. You can also use ideas from the Overview section in the lab assignment. Not more than a page for this section.
3. Warm-Up -- complete Section 3 Warm-Up in the lab assignment and explain your work in your own words and with appropriate references using sentences, equations, plots, Matlab code, etc.
4. FM Synthesis of Instrument Sounds -- complete Section 4 of the same name from te lab assignment. Explain in your own words and with appropriate references using sentences, equations, plots, Matlab code, etc.
5. Conclusion -- draw conclusions from your work and explanations in the earlier sections. Probably half of a page for this section.

Please read the guidelines for mini-project writeups on the homework page at

http://users.ece.utexas.edu/~bevans/courses/signals/homework/index.html

This mini-project report is something that you show at interviews as an example of your work.

Problem 3.2

When the problem asks to draw a spectrogram by hand, you may use the spectogram function in Matlab instead. However, please compute the formula for the instantaneous frequency to allow you to check against what the spectrogram is showing.

For problem 3.2 on mini-project #1, the signal duration is 1.35s, which corresponds to 10800 samples (i.e. 8000 samples/s times 1.35 s = 10800 samples).

Using a large block size for the spectrogram, such as 8000, would only allow a couple of blocks to be used for the spectrogram.

You can increase the duration of the signal according to the block size chosen.

Additional comments are given below for parts (a) and (c).

Problem 3.2(a)

Here is my code for problem 3.2(a) using the default signal duration of 1.35s. I used a block size of 256 for the spectrogram. A block size of 128 and 512 also work. For block sizes larger than 512, I would need to increase the signal duration by the same factor. More on that approach in the hint for Problem 3.2(c).

```
% Mini-Project #1
% Problem 3.2(a)
% N = block size for spectrogram.
% Frequency resolution is fs / N.
% Using a larger block size means
% that we'll need more samples for
% the signal (i.e., increase tmax).
fs = 8000;
Ts = 1/fs;
tmax = 1.35;
t = 0 : Ts : tmax;
f0 = 900;
fm = 3;
B = 200;
x = cos(2*pi*f0*t + B*sin(2*pi*fm*t));
N = 256;
spectrogram(x, hamming(N), N/2, N, fs, 'yaxis');
```

You can also use the approach described below for Problem 3.2(c) with $f_m = 3$ Hz.

Problem 3.2(c)

For mini-project #1 problem 3.2(c), we can see the harmonic structure in the signal by generating a longer duration signal and using a larger block size for the spectrogram.
I'm appending my code for problem 3.2(c) to plot the corresponding spectrogram.
If you zoom into the spectrogram, either manually by clicking the magnifying glass icon and the sweeping a rectangular region to expand or by using
ylim([0.8 1])
You'll see harmonics shown by bright yellow lines at $f_0 = 900$ Hz, $f_0 + f_m = 930$ Hz, $f_0 - f_m = 870$ Hz, etc.
Here, $f_m = 30$ Hz.
That is, you're seeing harmonics of 30 Hz plus an offset of 900 Hz.
You can try out other values of f_0 and f_m.

```
% Mini-Project #1
% Problem 3.2(c)
% N = block size for spectrogram.
% Frequency resolution is fs / N.
% Using a larger block size means
% that we'll need more samples for
% the signal (i.e., increase tmax).
% Ninc the factor to increase both
% tmax and N for the analysis.
```

```
fs = 8000;
Ts = 1/fs;
Ninc = 32;
tmax = 1.35*Ninc;
t = 0 : Ts : tmax;
f0 = 900;
fm = 30;
B = 20;
x = cos(2*pi*f0*t + B*sin(2*pi*fm*t));
N = 256*Ninc;
spectrogram(x, hamming(N), N/8, N, fs, 'yaxis');
```

Problem 3.2(e)

The analysis is very similar to Problem 3.2(c) except that $f_m = 300$ Hz.
If you zoom into the spectrogram, either manually by clicking the magnifying glass icon and the sweeping a rectangular region to expand or by using

```
ylim( [0.5 1.3] )
```

You'll see harmonics shown by bright yellow lines at $f_0 = 900$ Hz, $f_0 + f_m = 1200$ Hz, $f_0 - f_m = 600$ Hz, etc. That is, you're seeing harmonics of 300 Hz plus an offset of 900 Hz.

Problem 3.2 **Mathematical Analysis**

We can use the results from homework problems 2.2 and 2.4(d) to provide a mathematical analysis to unlock the harmonic structure in mini-project #1 problem 3.2.

In miniproject #1 problem 3.2, the signal has the form

$x(t) = \cos(2\,\pi\, f_0\, t + B \sin(2\,\pi\, f_m\, t))$

We'll use the following trigonometric identity

$\cos(\alpha+\beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)$

to give

$x(t) = \cos(2\,\pi\, f_0\, t + B \sin(2\,\pi\, f_m\, t)) = \cos(2\,\pi\, f_0\, t)\cos(B \sin(2\,\pi\, f_m\, t)) - \sin(2\,\pi\, f_0\, t)\sin(B \sin(2\,\pi\, f_m\, t))$

The term $\cos(2\,\pi\, f_0\, t)\cos(B \sin(2\,\pi\, f_m\, t))$ is (sinusoidal) amplitude modulation that was analyzed in homework problem 2.2. Multiplication by $\cos(2\,\pi\, f_0\, t)$ will cause a shift in the frequency components of $\cos(B \sin(2\,\pi\, f_m\, t))$ by $+f_0$ and $-f_0$. The frequency components in $\cos(B \sin(2\,\pi\, f_m\, t))$ were explored in homework problem 2.4(d). The solution for homework problem 2.4(d) used the Taylor series expansion of $\cos(\theta)$ which is

$$\cos(\theta) = 1 - \frac{1}{2!}\theta^2 + \frac{1}{4!}\theta^4 - \cdots$$

to obtain

$$\cos(B \sin(2\pi f_m t)) = 1 - \frac{1}{2!}B^2\sin^2(2\pi f_m t) + \frac{1}{4!}B^4\sin^4(2\pi f_m t) - \cdots$$

where $\sin^2(2\pi f_m t) = \frac{1}{2} - \frac{1}{2}\cos(2\pi(2f_m)t)$ which has frequency components of $-2f_m$, 0, and $2f_m$. We can view $\sin^4(2\pi f_m t) = \sin^2(2\pi f_m t)\sin^2(2\pi f_m t)$ which gives frequency components of $-4f_m$, $-2f_m$, 0, $2f_m$, and $4f_m$. If we include all of the higher-order terms, then we'll get all of the even harmonics of f_m.

We can perform a similar analysis of the term $\sin(2\pi f_0 t)\sin(B\sin(2\pi f_m t))$. This term is also a form of (sinusoidal) amplitude modulation that uses $\sin(2\pi f_0 t)$ instead of $\cos(2\pi f_0 t)$ that was analyzed in homework problem 2.2. A similar effect happens. Multiplying by $\cos(2\pi f_0 t)$ will cause a shift in the frequency components of $\sin(B\sin(2\pi f_m t))$ by $+f_0$ and $-f_0$.

We can expand $\sin(B\sin(2\pi f_m t))$ using the Taylor series

$$\sin(\theta) = \theta - \frac{1}{3!}\theta^3 + \frac{1}{5!}\theta^5 - \cdots$$

to obtain

$$\sin(B\sin(2\pi f_m t)) = B\sin(2\pi f_m t) - \frac{1}{3!}B^3\sin^3(2\pi f_m t) + \frac{1}{5!}B^5\sin^5(2\pi f_m t) - \cdots$$

If we keep all of terms in the series, then we'll get all of the odd harmonics of f_m.

In conclusion,

$x(t) = \cos(2\pi f_0 t + B\sin(2\pi f_m t)) = \cos(2\pi f_0 t)\cos(B\sin(2\pi f_m t)) - \sin(2\pi f_0 t)\sin(B\sin(2\pi f_m t))$

has frequency components of ..., $f_0 - 2f_m$, $f_0 - f_m$, f_0, $f_0 + f_m$, $f_0 + 2f_m$, ... That's what we see in the spectrograms in mini-project #1 problem 3.2.

Problem 4.1

When Matlab evaluates a function that hasn't already been defined such as `bell`, it will search for a file with the name of the function followed by .m such as `bell.m` on the Matlab search path. You may add folders/directories to the Matlab search path by using the addpath command, or by using the "Set Path" option in the Home tab in the Matlab user interface. If the file defining a function has changed, Matlab will reload the file.

Problem 4.3

Please work all six examples, but in your report, only include the write-ups for two of the six cases—choose one of the first four and one of the last two.