CSE 3313 – Homework #2

Assigned:	Wednesday, September 4, 2019
Due:	Wednesday, September 11, 2019 at the end of class

Note the following about the homework:

1. You must show your work to receive credit.
2. For the hand-written problems, submit a hard-copy.
3. For the problems requiring Python, upload your source code to Blackboard. See instructions at the end of the document.

Assignment: Work these by hand

1. (30 points) Determine the amplitude, radian frequency, and phase for the sinusoid shown in Figure 1. When performing your calculations, be sure to include units. **Your final answer should be in the form**

$$x(t) = Acos(\omega_0 t - \phi)$$

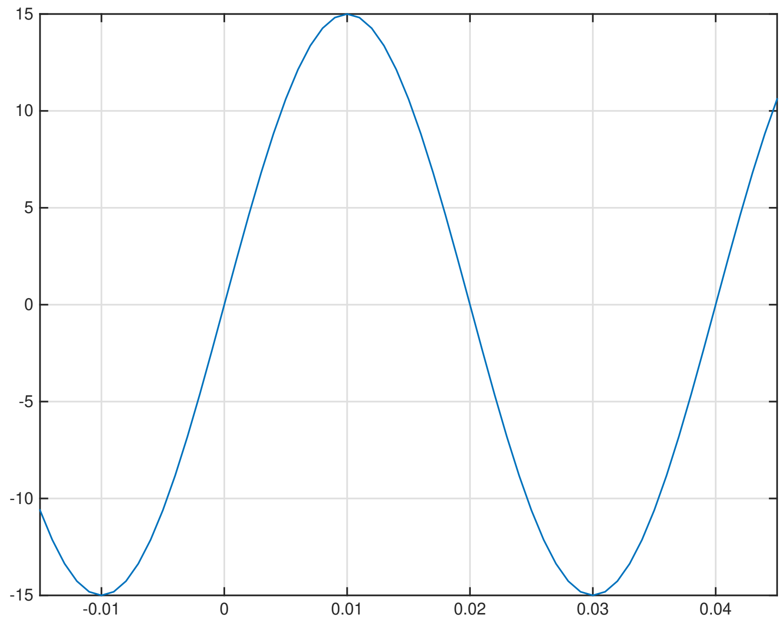

Figure 1: sinusoid

2. (10 points) The website http://www.zytrax.com/tech/audio/audio.html provides the frequency ranges for a number of common sounds in the section titled *Audio Frequencies*. For each of the sounds below, use the Shannon Sampling Theorem to determine the lower bound on the sample frequency required to accurately reconstruct the sound. Base your decisions on the frequencies included in the problem.

(a) a piece of music in which a piano (28-4186 Hz), violin (196-3136 Hz), and acoustic guitar (82-1397 Hz) are being played at the same time

(b) human hearing (20-20000 Hz)

Assume the the signal we are sampling is band-limited such that the highest frequency present is the maximum frequency given in the problem. Round your answer up to the nearest integer if necessary.

3. (10 points) We are filming a spinning wheel, which has a dot on its edge. If the wheel turns at a rate of 25,000 revolutions per hour, what is the minimum frame rate (in frames/second) for the camera if we wish to accurately determine the path of the dot while avoiding aliasing? Choose the smallest integer that satisfies this. Keep in mind that one revolution of the wheel is essentially a cycle.

   See the example at
   `http://www.ignaciomellado.es/blog/Measuring-heart-rate-with-a-smartphone-camera`.

Applications

4. **Application Area: Music** (50 points)
   Purpose: See how sinusoids of specific frequencies lead to specific sounds.

   On the course website are the files `DSPFirst-lab3.pdf` and `twinkle.pdf`. The former has the background material that you need for this assignment and the latter has the music to use. Both are from [MSY98].

   (a) The document `twinkle.pdf` has 8 lines of notes from *Twinkle, Twinkle Little Star*. You will write a Python program to produce an audio file of the 24 notes from lines 1 and 3.

   (b) The first note will be C_5 (key number 52) on the keyboard shown in Figure C.2 on page 436. The other notes will be relative to this note.

   (c) For each note of the song, you will need to determine its approximate frequency, which will be an offset of the key A_4 (key number 49). The formula for this becomes

   $$f = 440 \times 2^{(keyNumber-49)/12}$$

   You will then use this frequency to produce the **first 0.5 second of values of a cosine wave**. When creating your "continuous" signal, use a step rate of 1/8000 in your time values and an amplitude of 1.

   (d) Each note will represent 0.5 second of a sound, regardless of what the sheet music actually says. That is, don't differentiate between quarter notes, half notes, and so forth.

   (e) You will ultimately produce a single sequence of values for the 24 notes by concatenating the values of each of the cosine waves.

   (f) You will create a `.wav` file called `twinkle.wav` using your sequence of values and a sample rate of $f_s = 8000$. Use the PySoundFile library `https://pysoundfile.readthedocs.io/`. Note that this is not part of the Anaconda distribution, so you will need to install it.

   (g) Your program will be called `hw02.py` (write it EXACTLY like that, including the zero).

General requirements about the Python problems:

a) As a comment in your source code, include your name.

b) The Python program should do the work. Don't perform the calculations and then hard-code the values in the code or look at the data and hard-code to this data unless instructed to do so.

c) The program should not prompt the user for values, read from files unless instructed to do so, or print things not specified to be printed in the requirements.

To submit the Python portion, do the following:

a) Create a directory using your last name, the last 4 digits of your student ID, and the specific homework, with a hyphen between your ID and the homework number. For example, if John Smith has a student ID of 1000123456 and is submitting hw02, his directory would be named **smith3456-hw02**. Use all lowercase and zero-pad the homework number to make it two digits.

   If you have a hyphenated last name or a two-part last name (e.g., Price-Jones or Price Jones), let's discuss what you should do.

b) Place your .py files in this directory.

c) Zip the directory, not just the files within the directory. You must use the zip format and the name of the file (using the example above) will be `smith3456-hw02.zip`.

d) Upload the zip'd file to Canvas.

References

[MSY98] James H. McClellan, Ronald W. Schafer, and Mark A. Yoder. *DSP First: A Multimedia Approach*. Prentice Hall, Upper Saddle River, New Jersey, 1998. This is the book that I used the first time I taught Signal Processing and therefore much of what I know came from it. The authors believe this could be the first course taken by electrical engineering students. I don't agree, but they do assume less at the beginning than traditional texts which means they explain more than traditional texts. Like all academic textbooks for DSP, it's heavy on the math.