**Tune-Up Tuesday for October 5, 2021**

% Create a MATLAB simulation for fall 2021 [midterm problem 1.2](http://users.ece.utexas.edu/~bevans/courses/signals/lectures/midterm1/Appendix%20K%20K-66%20Midterm%201%20Fall%202021.pdf)

% The problem involves analysis of the output of a squaring system *y*(*t*) = *x*2(*t*).

% Let input *x*(*t*) = cos(2 p *f*1 *t*) + cos(2 p *f*2 *t*) where *f*1 = 110 Hz and *f*2 = 220 Hz.

% (a) Using a sampling rate of 8000 Hz and time from 0 to 3 seconds, write

% the code to generate a sampled version of the signal *x*(*t*).

fs = 8000;

Ts = 1/ fs;

tmax = 3;

t = 0 : Ts : tmax;

f1 = 110;

f2 = 220;

x = cos(2\*pi\*f1\*t) + cos(2\*pi\*f2\*t);

% (b) Play sampled version of *x*(*t*).  Describe what you hear.

*% Answer: The signal x(t) is composed of A note in the second octave (110 Hz)*

*% and A note in the third octave (220 Hz).*

*%* ***On a laptop****. On many laptop speakers, the 110 Hz tone may not be audible*

*% due to limitations in playing back low audible frequencies. On a laptop, the*

*% playback sounded like a single note with hum in the background.*

*%* ***Audio system with a sub-woofer****. A sub-woofer plays low audible frequencies*

*% down to 20 Hz. The sub-woofer is often a separate large speaker (due to the*

*% longer acoustic wavelengths l for low frequencies, i.e. l = c / f) in an audio*

*% system. On an audio system with a sub-woofer, the playback sounded like a*

*% beat frequency with hum in the background. Both notes were audible.*

soundsc(x, fs);

pause(tmax+1);

% (c) Plot the spectrum of the sampled version of *x*(*t*). *Principal frequencies*

*% are 110 Hz and 220 Hz.*

% **Using the spectrogram**. *See the second page for the plot.*

figure;

spectrogram(x, 512, 256, 512, fs, 'yaxis');

% **Using the fast Fourier transform approach from mini-project #1**.

fourierSeriesCoeffs = fft(x);

N = length(x);

freqResolution = fs / N;

ff = (-fs/2) : freqResolution : (fs/2)-freqResolution;

figure;

plot(ff, abs(fftshift(fourierSeriesCoeffs)));

xlabel('f');

xlim( [-1000, 1000] );

ylim( [-10, 15000] );

% (d) Using a sampling rate of 8000 Hz and time from 0 to 3 seconds,

% write the code to generate the sampled version of the signal *y*(*t*).

y = x.^2;

% (e) Play sampled version of *y*(*t*). Describe what you hear.

*% Answer: The signal y(t) is composed of ‘A’ note in the second octave (110 Hz)*

*% ‘A’ note in the third octave (220 Hz), ‘E’ note in the fourth octave (330 Hz),*

*% and ‘A’ note in the fourth octave (440 Hz). When played back, the signal*

*% y(t) has a higher pitch than x(t) but it was difficult to distinguish more than*

*% two notes. The 0 Hz term is not audible. Please see the answer in part (b).*

soundsc(y, fs);

pause(tmax+1);

% (f) Plot the spectrum of the sampled version of x(t).  *Principal frequencies*

*% are 0, 110, 220, 330, and 440 Hz. Principal frequencies at 0, 330, 440 Hz aren’t*

*% in x(t) and are called intermodulation distortion caused by the squaring system.*

% **Using the spectrogram**. *See below for the plot*.

figure;

spectrogram(y, 512, 256, 512, fs, 'yaxis');

% **Using the fast Fourier transform approach from mini-project #1**. *See below*.

fourierSeriesCoeffs = fft(y);

N = length(y);

freqResolution = fs / N;

ff = (-fs/2) : freqResolution : (fs/2)-freqResolution;

figure;

plot(ff, abs(fftshift(fourierSeriesCoeffs)));

xlabel('f');

xlim( [-1000, 1000] );

Spectrogram for sampled *x*(*t*)

Spectrogram for sampled *y*(*t*)



Spectrum for sampled *y*(*t*)

Spectrum for sampled *x*(*t*)