UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
Department of Electrical and Computer Engineering

ECE 498MH Signal and Image Analysis

Homework 2

Fall 2013

Assigned: Friday, September 13, 2013 — Due: Friday, September 20, 2013

Reading: Signal Processing First (SPF) pp. 36-57

Problem 2.1

Consider the following signal:

$$x[n] = 2 + 2\cos\frac{\pi n}{4} + \sin\frac{\pi n}{2} + \frac{1}{2}\cos\frac{3\pi n}{4}$$

(a) Find its Fourier series coefficients, X_k.

(b) Sketch the power spectrum, $|X_k|^2$.

(c) Compute the total power of the signal.

Problem 2.2

Consider the signal

$$x(t) = |\cos(2\pi t)|$$

(a) Sketch $x(t)$.

(b) What is its period, T_0? What is its fundamental frequency, Ω_0?

(c) Find the Fourier series coefficients.

- Hint #1: notice that $|\cos(2\pi t)|$ is sometimes equal to $\cos(2\pi t)$, and sometimes equal to $-\cos(2\pi t)$, so if you choose the right period of time over which to integrate, you might be able to get rid of the absolute value signs.
- Hint #2: use the relationship $\cos(2\pi t) = \frac{1}{2}(e^{j2\pi t} + e^{-j2\pi t})$ so that you can integrate exponentials instead of integrating cosines.
- Hint #3: $\int (e^{at} + e^{bt})dt = \frac{1}{a}e^{at} + \frac{1}{b}e^{bt}$.

Matlab Exercises

Problem 2.3

(a) Say "ah," or any other sustained vowel of your choice. Record your voice using the matlab command `wavrecord`. Use a sampling frequency of at least 8000 samples/second, but no more than 16000 samples/second; write down the sampling frequency you use.

(b) Use `plot` to plot the waveform, and `zoom` to zoom in. Notice how the mouth and throat (the vocal tract) rings like a bell each time the vocal folds close; the period between vocal fold closures is called the *pitch period*, and the frequencies at which the ringing occurs called *formant frequencies*. Choose ten consecutive pitch periods, starting with the zero-crossing just before a peak. Excise this waveform snippet using a command like `x=y(n_start:n_end)`, where `n_start` and `n_end` are the starting and ending sample numbers of your ten-pitch-period snippet.

(c) Use `t=[0:(N-1)]/Fs;` to compute the time, in seconds, of each sample in your waveform snippet, where `Fs` is the sampling frequency, and `N=length(x);`. Use `plot(t,x);` to show the whole waveform, with a time axis labeled in seconds. Print a copy of this plot as a PNG (`print -dpng`), and include it in your lab report. Answer the question: what is the fundamental period of your waveform, `T0`, in seconds? It should be roughly on the order of $0.005 \leq T_0 \leq 0.01$ seconds (a pitch frequency of $100 \leq F_0 \leq 200$Hz).

(d) Use `X=fft(x);` to compute the Discrete Fourier Transform of your signal. The DFT is actually just a scaled version of the Fourier series; the `fft` operation finds the Fourier series coefficients. Use `MagX=abs(X);` to compute its magnitude, and `PhaX=unwrap(angle(X));` to compute its phase. Use `subplot` to create two subplots, one above the other. Plot the magnitude in the top plot, and the phase in the bottom plot, as a function of the Fourier coefficient number `k=[0:(length(X)-1)];`. Print a copy of this plot, and include it in your lab report. Be sure that the first sample in your plot starts at Fourier coefficient number 0, not Fourier coefficient number 1!

(e) Use `zoom` to zoom in on your magnitude plot. Notice, first of all, that the number of frequency samples is equal to the number of time-domain samples. Notice, second of all, that the upper frequencies (above $N/2$) have the same magnitude as the lower frequencies, but the opposite phase; this is because matlab is actually using the upper half of the vector `X` to store the negative-frequency Fourier series coefficients. Finally, notice that only one sample out of every ten has large magnitude; this is because you used `fft` to compute the Fourier series based on *ten* periods, insted of just one period. Fix this problem: find the true Fourier series coefficients of your vowel sound by extracting every tenth sample from the `MagX` and `PhaX` vectors, using commands something like `M=MagX(1:10:length(MagX));` and `Phi=PhaX(1:10:length(PhaX))`. Use the `stem` command to plot just the first six coefficients of `M` (`stem([0:5],M(1:6));`), and the first six coefficients of `Phi`. Print a copy of this plot, and turn it in.

(f) Create a figure with two subplots. In the top plot, plot `x` as a function of `t`. In the second plot, plot `y` as a function of `t`, where `y` is the one-cosine approximation of `x`:

$$y(t) = M_1 \cos\left(\frac{2\pi t}{T_0} + \Phi_1\right)$$

Be sure to realize that, because matlab counts vector elements starting with 1 instead of 0, the first Fourier series coefficient is given by `M1=M(2)` and `Phi1=Phi(2)`.

(g) Repeat the previous step, but this time, use a two-cosine approximation:

$$y(t) = M_1 \cos\left(\frac{2\pi t}{T_0} + \Phi_1\right) + M_2 \cos\left(\frac{4\pi t}{T_0} + \Phi_2\right)$$

(h) Repeat the previous step, but this time, use a five-cosine approximation:

$$y(t) = \sum_{k=1}^{5} M_k \cos\left(\frac{2\pi k t}{T_0} + \Phi_k\right)$$

(i) Repeat the previous step, but this time, use a twenty-cosine approximation:

$$y(t) = \sum_{k=1}^{20} M_k \cos\left(\frac{2\pi k t}{T_0} + \Phi_k\right)$$

(j) Use the `soundsc` command (with the correct sampling frequency!) to play back your waveform snippet. Play back, also, the one-cosine, two-cosine, five-cosine, and ten-cosine approximations. Comment: how much of the auditory vowel quality is retained by each approximation?