BIOL 708L/NACS 728B Lab #4
Fall 2011

Fourier Transforms in MATLAB

The main goal of this lab is to make some signals in Matlab and see what their Fourier Transform looks like. Take your time on these and make sure you understand what's going on with the signals at each step.

For all problems here, use $N = 16$ points, starting from 0 (we need the points to be aligned properly with $n = 0$ when taking the Fourier Transform), and define n to be the ***column*** vector, N points long, starting from zero. **In all graphs in the time domain, make sure you explicitly use *n* for the variable along the horizontal axis.** Make another array of points corresponding to the Fourier frequency. Call it *fk* (not to be confused with "*f*" used below). *fk*, which takes values from 0 to (just less than) 1, should be N points long, starting at 0, in steps of $1/N$. **In all plots in the frequency domain, make sure you explicitly use *fk* along the graph's horizontal axis.**

1) Set of cosine signals of varying frequencies

a) We'll use 9 frequencies of the form k/N where k is an integer. Create a row vector of these frequencies, i.e., only the first 9 frequencies of *fk* but in ***row*** form, and call the vector *f*.

b) Create an array of 9 signals (N rows by 9 columns), where each column is given by $\cos(2\pi fn)$ for the appropriate individual frequency *f*. You can do this in a single step if you're careful. Call the array `cossigs`.

c) In a single figure with 9 subplots, using `bar()`, plot each of the signals, in order of frequency. Using `axis()`, set the axis limits to range from -1 to +1 in height and from `n(1)-.5` to `n(end)+.5` in width (the extra '.5's are so that the leftmost and rightmost bars don't get cut off—feel free to try it without them to see what happens). Make sure your horizontal axis is labeled by n and not some array that Matlab guessed.

d) Why did we only go use 9 frequencies and not $N - 1$ or N? Equivalently, why don't we need to have *f* go as far as *fk*?

2) *Examining the Fourier Transform*

a) In a new figure, using 9 subplots, in each using `bar()`, graph the magnitude of the Fourier transform (use the Matlab function `fft()` to compute the Fourier transform) of each signal in `cossigs`. Using `axis()`, set the axis limits to range from 0 to N in height and from `fk(1)-.5/N` to `fk(end)+.5/N` in width (the extra $.5/N$s are so that the leftmost and rightmost bars don't get cut off—feel free to try it without them to see what happens). Make sure your horizontal axis is labeled by *fk* and not some array that Matlab guessed.

b) What should the amplitude behave like? Is it non-zero in the correct points? What should the value of non-zero points be?

3) Applying the two point moving average filter

a) Create the impulse response of the two-point moving average convolutive filter using:

`h2pt = [1/2 1/2].'` and apply it to the cosine signals from (1) using: `filtcossigs = filter(h2pt,1,cossigs)`. Graph them the same way you did in (1), using the exact same axis limits. How do they compare with the results in (1) in amplitude? In particular, how do they compare with the results in (1) in amplitude for every point after the first points? Can you reason why they should do this? Do you know why the first points behave "oddly" compared to all the others? Hint: it's an artifact.

b) Repeat (2a) and (2b) but for the Fourier transform of `filtcossigs`.

4) *Impulse*

a) Create an N point version of the delta (impulse) function, called *del0*, as a column vector. You can do it simply by typing the correct numbers of ones and zeros in the right place, or more cleverly by making an array the same size as `n` using `zeros()` and then setting the appropriate values to one. [The most elegant (i.e. the most useful for complicated signals) is to use the `find()` function to determine which points to change from zero, e.g. `find(0==n)` .] Graph it in a new figure window using `bar()`. Make sure your horizontal axis is labeled by n. Using `axis()`, set the axis limits to range from -1 to +1 in height and from `n(1)-.5` to `n(end)+.5` in width.

b) In a new figure window, graph the magnitude of the FFT against *fk*. Using `axis()`, set the axis limits to range from 0 to N in height and from `fk(1)-.5/N` to `fk(end)+.5/N` in width. How does this compare to *any* of the Fourier transformed cosine signals? Can you interpret the difference?

5) *Filtered Impulse*. Filter the delta (impulse) using `h2pt`. Plot the filtered impulse in both the time and frequency domains. Interpret these plots compared to the plots of question 4, given the properties of the filter in the time domain and the observed Fourier Transform properties of filtered sinusoids.