EE-3221 LABORATORY

Week 7

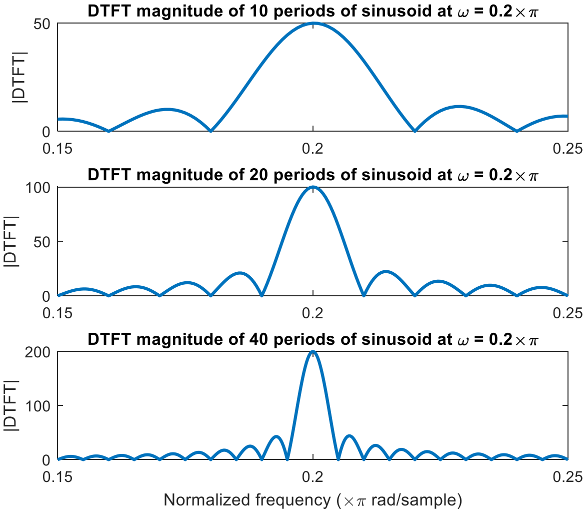
DFT (Discrete Fourier Transform) Windowing

Goal – Compare windows for DFT spectral analysis

Materials - Laptop computer with MATLAB

Background: Frequency Resolution and the DTFT (Discrete-Time Fourier Transform)

The figure at right illustrates how the DTFT of a sinusoid changes as more samples are collected. The frequency resolution improves as we observe the sinusoid longer. The resolution is 2π/N, where N is the number of samples collected; N is also the length of the *DFT* (the *DTFT* is plotted vs. continuous ω). The main lobe's width is twice the resolution, while the side lobes have width equal to the resolution. Note that the DTFT is approaching the ideal of an impulse in the frequency domain. Both a wide main lobe and relatively powerful side lobes can interfere with detecting nearby tones. This interference is addressed by windowing.



Overview: DFT Analysis and Windowing

The DFT assumes that a whole number of periods is input. If, say, 1.25 periods are presented, the algorithm analyzes the waveform that progresses for 1.25 of the original period and then jumps to the beginning of the real period. This can result in an incorrect spectrum given the underlying signal. Often instruments are operated in a manner where the exact period is not known. In addition to generating the "wrong" spectrum, adjacent spectral components are often obscured by the "side bands." "Windowing," or multiplying the time samples by another time function (thus convolving the 2 spectra), helps to minimize the effects at the edges of the function. However, the spectrum is further corrupted. Often this is desirable, however, because it can serve to separate two close-by spectral components so they can be observed.

Window design is its own field of study with many articles written about it. Key tradeoffs between various window types, along with their typical units, are:

- Height of the main lobe (relative to rectangular window (*i.e.*, just limiting the signal to N samples), linear units). The major lobe is the frequency image, perhaps spanning several frequency samples, caused by a single, pure frequency in the input.
- Highest sidelobe in dB (-13 to -57 for some common windows; a parameter in some windows, such as the Chebyshev). A sidelobe is separated from the main lobe by a null.
- Bandwidth. One common definition is the width at which the magnitude response is 3 dB down from the maximum. This is at a minimum for the rectangular window, but the rectangular window has very poor roll-off and side lobe height.
- Roll-off: how quickly the sidebands decay in dB/octave.

Your task

Consider the signal given by:

```
n=0:N-1;
c1 = 1.00*cos(0.48*pi*n);
c2 = 0.01*cos(0.52*pi*n);
x = c1 + c2;
w_pi = n/N*2; % digital frequency with pi divided out
```

N is not given. You will experiment with different values of *N*.

Component c2 has a voltage that is only 1% of c1's, thus its power is 0.01% of c2's, or -40 dB relative to c2. This is a difficult detection problem (and we haven't even added noise!): the components are close in frequency and vary greatly in power, making it likely that c2 will get buried in one of c1's lobes.

Consult MATLAB's Signal Processing Toolbox documentation to learn about the available window types. Type **doc** to open the documentation tool, then browse through the contents to Signal Processing Toolbox | Spectral Analysis | Windows | Windows. Read this article. You may want to read some of the linked pages and experiment with the GUI tools described, but they are not necessary for this lab.

You will use 4 window types. You may add a couple more if desired.

1. Rectangular (unwindowed)
2. Hamming, periodic (note that you should specify 'periodic' for any window that supports it; **doc hamming** to learn why)
3. Your choice #1: you many use any of the ones available in MATLAB. There are no selection criteria. Some other common windows you might want to consider are the Blackman and Hann windows. The Chebyshev window is an interesting choice.
4. Your choice #2.

Part 1: Compare spectra for all your windows using 3 ***favorable*** values of N: 50, 100, and 150.

1. *Include your magnitude spectra for your values of N.*
2. *What makes these values of N "favorable"?*
3. *What window or windows perform well in this situation? Justify your response with reference to features in your spectra.*
4. *What is the minimum favorable N that you're comfortable using for this task. Why?*

Organize your plots for easy comparison and label them accordingly (xlabel, ylabel, title; include both quantities and units). Put spectra in ***subplots*** to avoid an excessive profusion of figures. Use a dB scale and radians/sample (× π) as your units. The following snippet will be helpful.

```
subplot(4,1,2)
plot(w_pi, 20*log10(abs(fft(x.*hamming(N, 'periodic')'))), 'b-o')
title(sprintf('Hamming, periodic, N=%g', N))
```

Part 2: Compare spectra for all your windows using 4 ***unfavorable*** values of N: 23, 61, 138, 277.

5. *Include your magnitude spectra for your values of N.*
6. *Explain why these numbers are unfavorable. Recall that the DFT samples the DTFT at all integer multiples of 2π/N.*
7. *Discuss the pros and cons of using a larger N in this context.*
8. *What window or windows perform well in this situation? Justify your response.*
9. *What is the minimum unfavorable N that you're comfortable using for this task. Why?*
10. *Include all of your MATLAB code as an appendix. Use a monospaced font like Consolas and use a style with no extra inter-line spacing.*