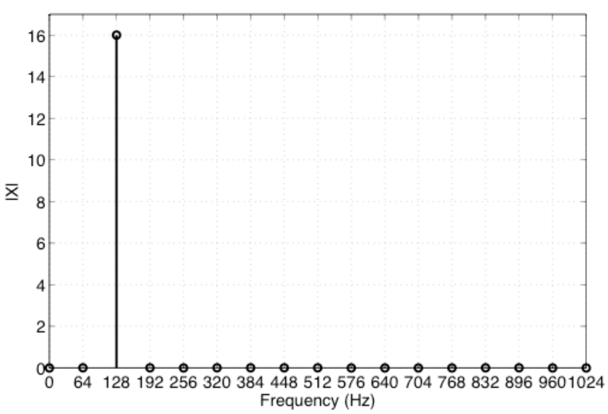
Lab 3

Windowing

0. Preface

In this lab you will explore the danger that faces us all: living in finite time and space. You will finally answer the question posed at the end of the first section of Lab 2. "What question?" you may ask. Let me briefly recap that exciting episode.

You were just a young thing then—pink as a newborn and only doing what you were told. That Tuesday was like any other during the semester. With MATLAB running, you calculate the magnitude spectrum of 32 samples of a 128 Hz sinusoid at a sampling rate of 2,048 Hz. Up pops (Figure 1, left) a familiar Kronecker delta function—well, as familiar as a Kronecker delta function can be in your day-to-day life—centered right on 128 Hz. You feel comforted by the consistency in the theory. You expected what you see.



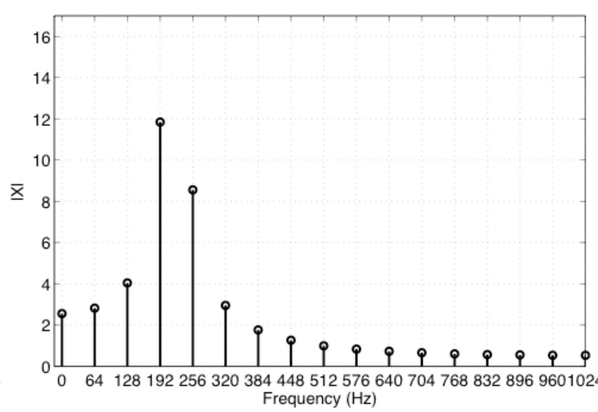


Figure 1: (Left) The magnitude spectrum of a 128 Hz sinusoid sampled at 2048 Hz over $0 \le n \le 31$. (Right) The magnitude spectrum of a 220 Hz sinusoid sampled at 2048 Hz over $0 \le n \le 31$

A dog barks. In the distance. At least you think it was a dog.

The next question asks you to evaluate the spectrum of 32 samples of a 220 Hz sinusoid sampled at the same rate: 2,048 Hz. “Well that will just be another Kronecker delta function, but centered at 220 Hz.” After making the necessary adjustment to the code you have the surprise ***of your life*** (Figure 1, right): there is energy in every fcuking frequency. Sweat beads on your upper lip, like you just ate something you shouldn’t have but it’s too late. All seems wrong with the world.

Instead of taking only 32 samples of this 220 Hz sinusoid, you take 512 samples and then evaluate its magnitude spectrum. This time you see (Figure 2) what you expected in the first place: a single, tall and triumphant Kronecker delta function centered at 220 Hz. The sweat dissipates. Tears form.

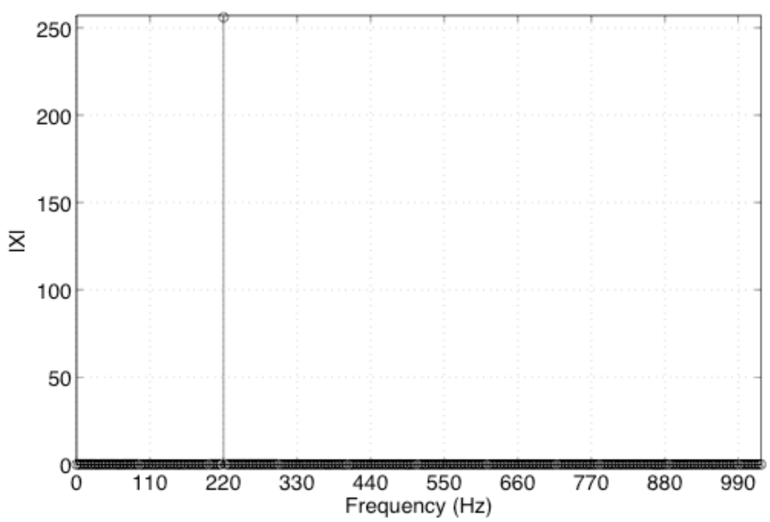


Figure 2: The magnitude spectrum of a 220 Hz sinusoid sampled at 2048 Hz over $0 \le n \le 511$*. Now it is properly resolved. Peace has returned to Mordor.*

Distant dog barks again. Scene. Roll credits.

The DFT of an N-length signal uses only N basis functions (phasors) having normalized frequencies of $f_k = \frac{k}{N},\ 0 \le k \le N-1$. The DFT of any N-length sampled sinusoid with a frequency that is not one of these N frequencies will suffer from *spectral leakage* because such a signal resembles something in every N-length DFT basis function. Another way to think about this is the effect of cropping a periodic sequence to a length that is not an integer number of periods. When you evaluate the discrete Fourier series of this signal (which is nothing more than the DFT), the periodized sequence now contains a discontinuity (Figure 3). The spectral leakage is caused by this discontinuity.

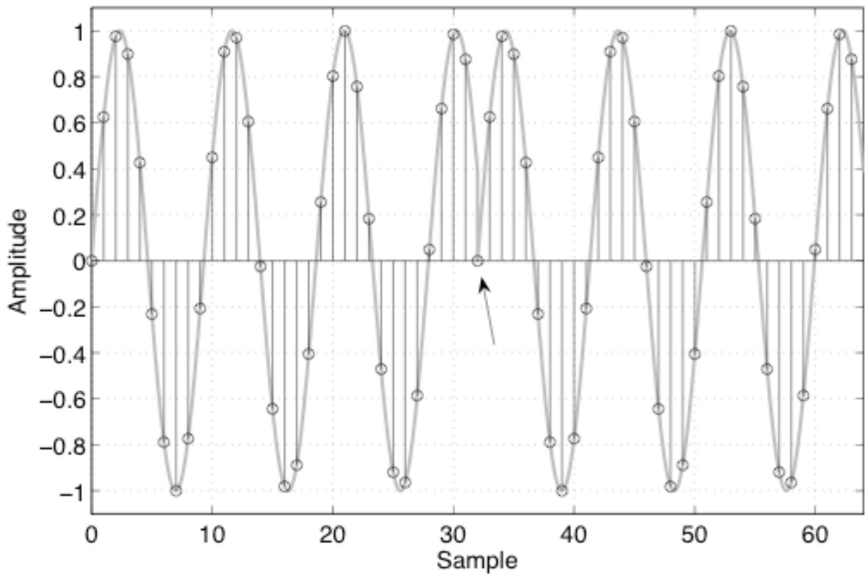


Figure 3: The periodized waveform (gray) resulting from 32 samples of a 220 Hz sinusoid sampled at 2048 Hz. Arrow points to the discontinuity in the waveform.

We solved the embarrassing leakage problem, shamefully shown in Figure 2, when we used 512 samples of the 220 Hz sinusoid. That way, the periodized sequence has no discontinuities, and the expected components are found by the DFT.

This synthetic signal is not very realistic however, since most, if not all, real-world signals vary in time (Figure 4) and are certainly not infinite. Thus one can never find a value N for which a real-world sampled signal is periodic—in the strict sense of the word. “N’kay,” you think, and take a deep breath. “Let’s just increase N to infinity and then we won’t have to worry about periodicity or spectral leakage because the basis over which we decompose the signal will consist of every frequency between D.C. and the Nyquist frequency.” Did I not just mention that all real-world signals are time-varying? Even though we could obtain a precise measurement of the magnitudes and phases of every component present in that signal, we would have no idea over what times those components were active.

Spectral leakage is something that will always exist. This lab will teach you how to “deal with it” so you can “move on”.

Figure 4: The celebrated composition “Flight of the Bumblebee” by Nikolai Rimsky-Korsakov is a celebrated example of a time-varying signal. It becomes even more so when real bees are involved.

1. Interpolating the DFT

Though you will never be able to rid yourself of spectral leakage, one way to help interpret “soggy spectra” is to just increase your signal length artificially by “zeropadding”. Let’s see what happens then.

1.1 Using 32 samples of a sinusoid with frequency 220 Hz, sampled at a constant rate of 2,048 Hz, zeropad it to a total length of 1024 samples and calculate the DFT of the result. Plot this using a line, and overlay this with a stem plot of the DFT of the original 32 samples. Include your plot and code. See Figure 5 for guidance.

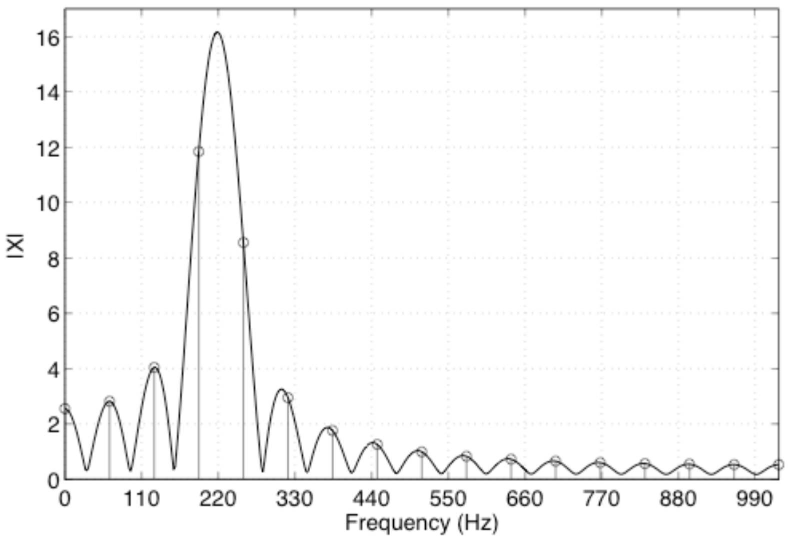


Figure 5: The DFT of the length-32 sinusoid with frequency 220 Hz (stem), and with zeropadding to length-1024 (line).

Now, as if by magic, we see a peak smack-dab on 220 Hz with the correct magnitude. By zeropadding the length-32 sequence we are in effect sampling the discrete-time Fourier transform (DTFT) at a finer resolution than 1/32. This does *not* mean we have cheated the system and solved our frequency resolution problems.

1.2 Using your code from 1.1, reduce the frequency of the sinusoid to 128 Hz and run it again. Include just the plot and comment on what has happened.

By your results in 1.2 you see that zeropadding is not always a good thing. What once was a single real component at 128 Hz becomes several components, for instance, one at 128 Hz and one at around 32 Hz. Every time you zeropad a signal you are inadvertently windowing the original signal (in an abstract sense, an infinite signal) by a rectangular window.

Since a multiplication in the time-domain is a convolution in the frequency domain, the spectrum of an infinite-length 220 Hz sinusoid, which has delta functions at ±220 Hz, is being convolved with the spectrum of the rectangular window (which has the shape of a sinc function). The convolution of the sinc and the impulse at the negative frequency is superposed with that of the positive frequency. This effect is most visible in Figure 5 at the low frequencies where the first "lobe" to the left of the largest one ("main lobe") is larger than any of the lobes to the right.

Windowing is also occurring in 1.2, resulting in the spectrum you see. This time the convolution of a 32-sample sinc with the impulse at 128 Hz is such that zeros are created in every frequency bin except one. After zeropadding this signal however, spurious frequencies are introduced because the original signal is now windowed in such a way that a discontinuity is introduced into the periodized signal.

Spectral leakage may not be a big problem when it is known that only one real sinusoid is present. As seen in Figure 5, by zeropadding the signal we are able to resolve the exact frequency and amplitude present. But what if a signal consists of two or more frequencies that are close together and have very different amplitudes?

1.3 Using the same code in 1.1, perform a DFT of a length-32 signal that is a sum of the following three sinusoids:

$$x[n] = \sin(2\pi 128n/2048) + 0.2\sin(2\pi 220n/2048) + 0.01\cos(2\pi 525n/2048)$$

Also find the DFT of the signal zeropadded to a length 2048. Overlay the DFT spectrum of length-32 using `stem`, with the zero-padded DFT spectrum using `plot`, an example of which is seen in Figure 5. Include just your plot.

1.4 Without the knowledge that there are multiple pure tones in this sequence, can you pick out the right frequency components in the magnitude spectrum?

Do not be mistaken—as many are—that adding more zeros to your input will give you finer frequency resolution. In effect, it makes your spectrum converge to the Fourier transform, or DTFT, of the sampled and thus periodized signal. Zeropadding does not give you more resolution in the sense that you can differentiate between closely spaced components. Fortunately there are ways to remedy the effects of spectral leakage. And it goes something like this.

2. The Frequency Content of Windows

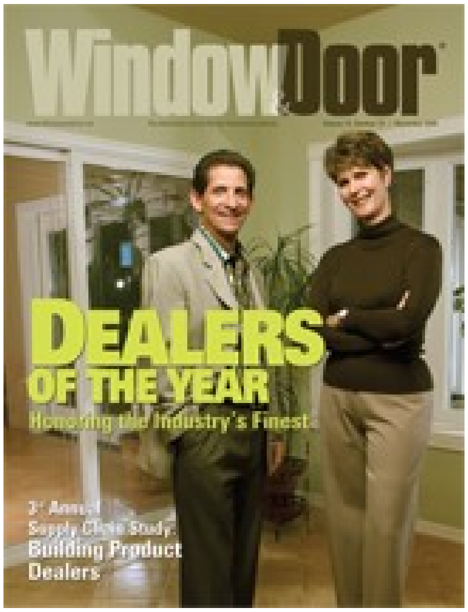


Figure 6: Window design is such an art that dealers have their own trade shows and magazines!

Windowing is necessary in our world because we not only like to know *what* happens, but *when* it happens. We also like to stay warm in the winter. Since multiplication in the time-domain is convolution in the frequency domain, one obviously wants a window that has an impulse-like spectrum. The only window that has that property is an infinite rectangular window, which is not realistic, and in fact leads to complete uncertainty in how a signal varies with time. With careful window design, however, one can fashion a window that will relieve the problems above, and at the same time impress Prof Martha Stewart.

There are literally hundreds of windows one can use to help analyze or process data. Some window designs become popular and then pass out of fashion (Figure 6). We will focus on four that have stood the test of time.

A length-N rectangular window is defined as:

$$w[n] = \begin{cases} 1, & 0 \le n \le N-1 \\ 0, & else \end{cases}. \tag{1}$$

A length-N triangular window, where N is necessarily odd, is defined as:

$$w[n] = \begin{cases} \left(\frac{N-1}{2} - \left| n - \frac{N-1}{2} \right| \right) \frac{2}{N-1}, & 0 \le n \le N-1 \\ 0, & else \end{cases}. \tag{2}$$

A length-N sine window is defined as:

$$w[n] = \begin{cases} \sin(\pi n/(N-1)), & 0 \le n \le N-1 \\ 0, & else \end{cases}. \tag{3}$$

A length-N Hann window is defined as:

$$w[n] = \begin{cases} \frac{1}{2}(1-\cos(2\pi n/(N-1))), & 0 \le n \le N-1 \\ 0, & else \end{cases}. \quad (4)$$

You will now analyze the frequency content of these four common windows. After that you will see how these windows perform in resolving the frequency content of the signal you synthesised in 1.3.

2.1 Program the four windows above. Using N=31 for the triangular window, and N=32 for the others, plot each window on the same graph (using `plot` instead of `stem`) and label each using `gtext`. Include this plot in your report.

2.2 Now evaluate the DFT of each window at 1,024 points around the unit circle (`fft(w,1024)`). Plot the normalized magnitude spectrum (not dB) as a function of normalized frequency of each window on four different plots; but before you do this you will want to use `fftshift` to move the zero frequency to the center. The normalized frequency axis will go from $-0.5 \le f \le 0.5$. Your figures should look like those in Figure 7. Include your figure in your report.

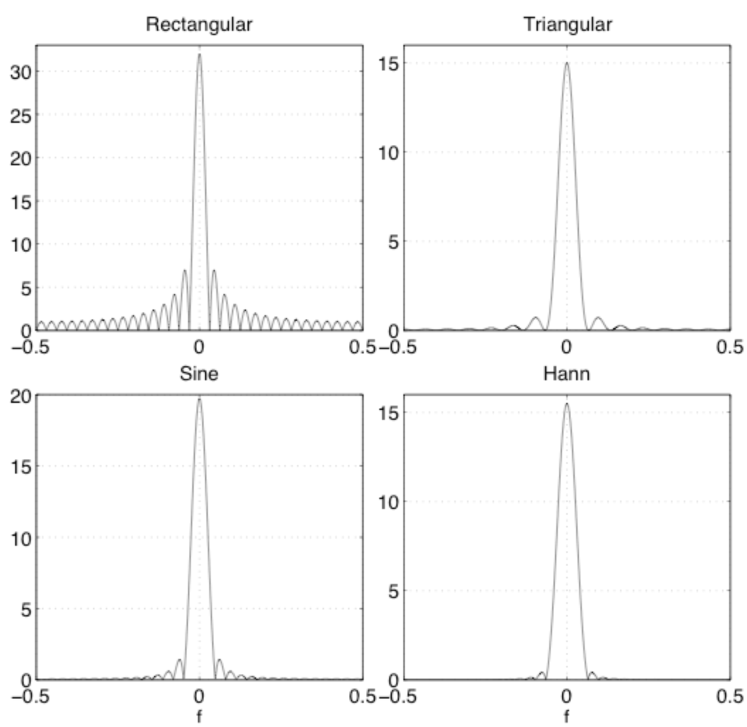


Figure 7: The frequency content of the four window functions given by (1-4)

One thing that is immediately apparent is that the windows all have different spectra. The main-lobe is thinnest for the rectangular window, and widest for the Triangular and Hann. The sidelobes are at different levels for each one too.

The subtle differences between the windows can be made much more clear using a logarithmic magnitude scale. The magnitude spectrum can be transformed into a normalized decibel spectrum using the following equation:

$$|X[k]|_{dB} = 20\log_{10}\left(\frac{|X[k]|}{\max(|X[k]|)}\right). \tag{5}$$

A spectrum that is normalized in this way has a maximum value of zero, and everything else is negative—so don't be alarmed.

2.3 For the spectra found in 2.2, plot the normalized dB spectrum of the rectangular and triangular window functions (1-2), on the *same* graph. Limit the magnitude axis to [-80:5] dB, and the frequency axis to [-0.2:0.2]. Include your plot with a legend, as well as your code. Your figure should look similar to Figure 8.

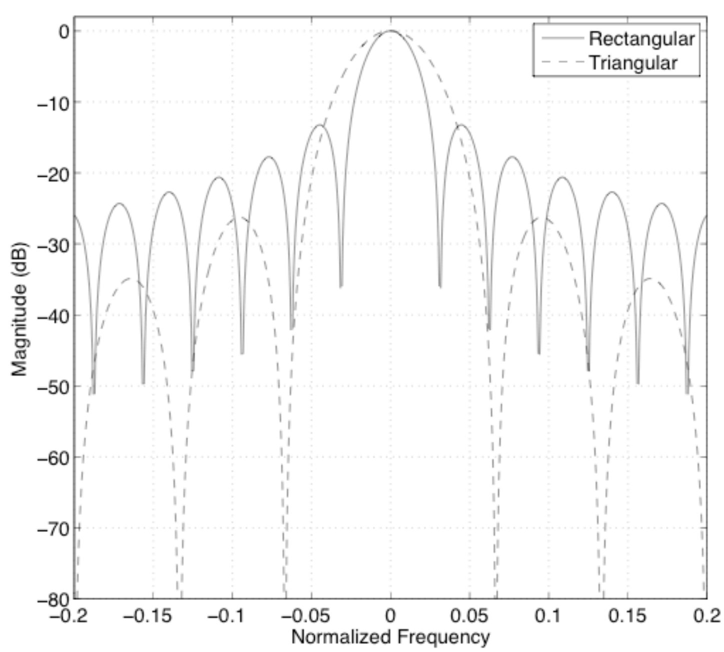


Figure 8: The frequency content of the rectangular and triangular windows

2.4 Find the width of the main-lobe for these two windows. What is the width of the main-lobe for the triangular window in terms of that for the rectangular window? Do you have any intuition why this is the case?

2.5 At what magnitude dB is the height of the first side-lobe for the rectangular and triangular windows?

2.6 Repeat 2.3 for the Sine and Hann window functions (3-4). Include your plot with a legend. No need to include your code.

2.7 What is the width of the main-lobe for the Sine and Hann window?

2.8 At what magnitude (dB) is the height of the first side-lobe for the Sine and Hann window?

From the above we can clearly see that of these four windows, the rectangular window has the thinnest main-lobe but the highest side-lobes. It also has the "slowest" decay of the side-lobes. The triangular window has a steeper decay of the side-lobes, but the main-lobe is twice as wide as that of the rectangular window. The Hann window has a similar main-lobe width, but has the lowest magnitude side-lobes of all these four windows.

3. Application of these windows to signal analysis

Now let's apply these windows to the problems observed above in 1.4.

3.1 Window the 32-sample sequence created in 1.3 with the rectangular, triangular, Sine, and Hann windows. Plot each spectrum (four plots) using stem (not in dB). Plot the normalized magnitude of each one (`abs(X)./max(abs(X))`). This time do not use `fftshift`. Use the proper frequency indicies, i.e., `f = n*Fs/N`. Plot only the frequencies between 0 and the Nyquist frequency. Include your code and plot. Your plot should look like Figure 9: The normalized magnitude DFT of the length-32 windowed sum of three sinusoids with frequencies 128, 220, and 525 Hz.

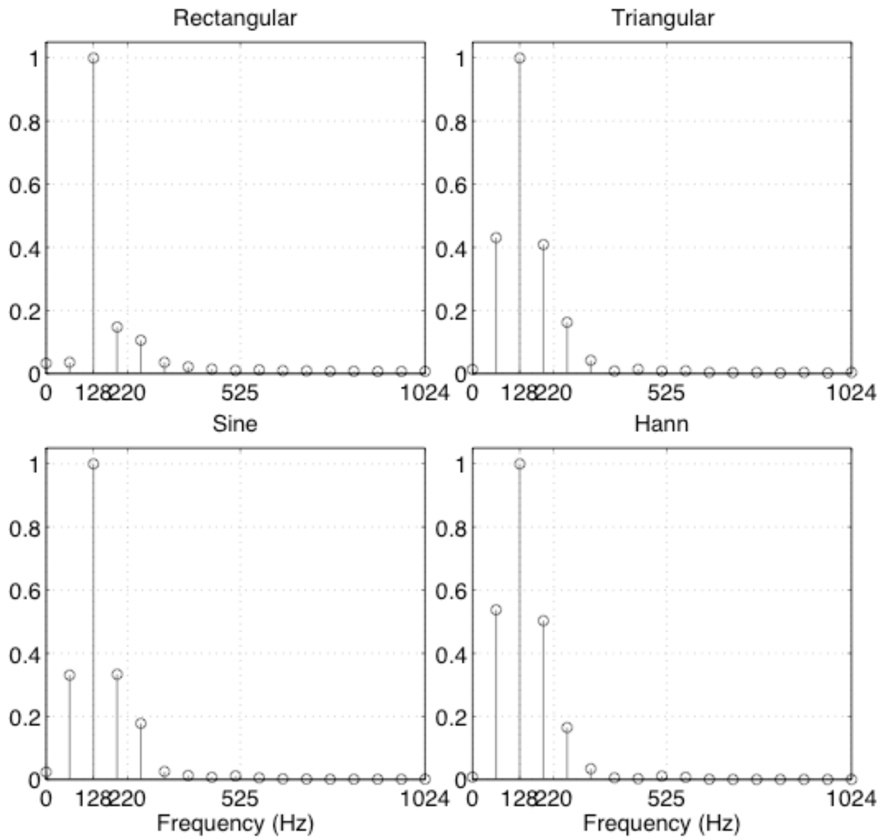


Figure 9: The normalized magnitude DFT of the length-32 windowed sum of three sinusoids with frequencies 128, 220, and 525 Hz

Obviously we still can't resolve the component at frequency 220 Hz since that frequency has fallen in a crack. The Hann and Sine windows do show energy present at 525 Hz that does not appear to be due to the windowing. Having only 32 samples to begin with, this is the best we can do with one window. Let's go one-step further and zeropad each sequence to length 2,048—effectively windowing the windowed sequence by a rectangular window.

3.2 For each of the windowed signals in 3.1, zeropad each sequence to length 2,048, and evaluate the DFT (you can do both at the same time by `fft(x,2048)`). Plot each magnitude spectrum (four plots) using plot and a normalized dB magnitude scale. Plot only the frequencies between 0 and the Nyquist frequency. Include your code and plot.

3.3 Answer all of the following questions:

- Knowing how the main-lobes depend on the type of window, and how the side-lobes decay, for which of these windows can you say with certainty that this signal has three components?
- For which windows can you say with accuracy the frequency of the middle amplitude sinusoid?
- Why are the Sine and Hann windows good at revealing the lowest amplitude component, but not the middle amplitude component?
- Why should the lowest amplitude component have a normalized dB level of –40 dB?