

# Tuesday Warm Up, Unit 1: Filter Design

Prof. Jordan C. Hanson

April 1, 2025

## 1 Memory Bank

1. **Note:** in the following formulas,  $f_C$  is taken to be a fraction of the sampling frequency, so it has no units.
2. **The sinc function** is the inverse DFT of the ideal low-pass filter in the frequency-domain:

$$h[i] = \sin(2\pi f_C i) / (i\pi) \quad (1)$$

For  $i \rightarrow 0$ ,  $h[i] \rightarrow 2$ . The sinc function is usually defined over  $M$  samples, shifted right by  $M/2$  samples.

3. **The Blackman window** is a window for a filter kernel like the sinc function, to remove unwanted effects due to the finite length of  $h[i]$ .

$$w[i] = 0.42 - 0.5 \cos(2\pi i/M) + 0.08 \cos(4\pi i/M) \quad (2)$$

4. **The windowed-sinc filter** is the product of the sinc kernel and the Blackman window:

$$h[i] = K \left( \frac{\sin(2\pi f_C (i - M/2))}{i - M/2} \right) w[i] \quad (3)$$

The factor  $K$  must be set so that the sum of all the samples must be 1. For  $i = M/2$ ,  $h[i] = 2\pi f_C K$ .

5. **The convolution theorem** states that the Fourier transform of the convolution of two functions is the same as the product of the Fourier transforms of the two functions.

## 2 Windowed-Sinc Filters

1. Write octave functions that (a) implement the sinc function, and (b) implement the Blackman window.
2. Write an octave function that implements the product of the sinc and Blackman window, following Eq. 3.
3. Using the octave function `sum`, compute the sum of the samples in your filter kernel, given  $M$  and  $f_C$ . Set  $K$  equal to the reciprocal of the result, so your windowed-sinc function is *normalized*. That is, the sum of the samples in the kernel is set to 1.
4. Generate a vector of white noise using `randn` in the usual way, and convolve it with a kernel output from your windowed-sinc function.

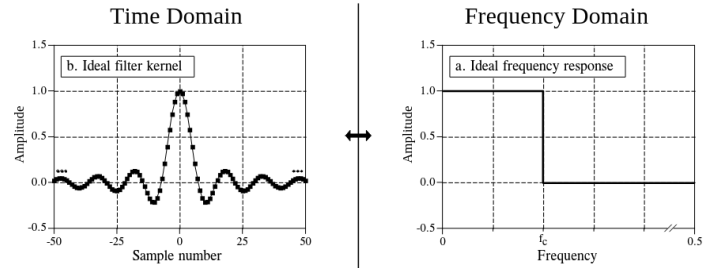


Figure 1: The ideal low-pass filter (right) is unattainable, due to the finite nature of the digital sinc function (left).

5. (a) Measure the *stop-band attenuation* of your filter. That is, what is the ratio of attenuated to unattenuated amplitude in the power spectrum? (b) Measure the *roll-off* of your filter. That is, in how many Hz does the frequency response go from maximum to minimum?

## 3 FFT Convolution

1. Suppose we are working with a moving-average filter, to remove high-frequency noise. Our incoming signal is a square wave. In order to predict the signal output, we must calculate the convolution of a square pulse with a square wave. Think conceptually about how convolution works. Show graphically that the convolution of two square pulses should be a triangle wave.
2. One of the fastest ways to convolve two functions is to compute the product of their DFTs, and then compute the inverse DFT of the result. This algorithm relies on the *convolution theorem*. Write a short octave script that computes the `fft()` of two square pulses, multiplies them, and then computes the `ifft()` (inverse DFT) of the result. Graph the result to show that it is indeed a triangle wave.