Thursday Warm Up, Unit 1: Filter Design

Prof. Jordan C. Hanson

April 3, 2025

1 Memory Bank

1. **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. **Recursive filter formula**. Start with convolution, and let $h[i] = a_i$. The result is

$$y[n] = \sum_{i=0}^{N} a_i x[n-i] \tag{1}$$

Now add *feedback from prior output samples* to compute the *next* output sample, using coefficients labeled b_i. Note that $b_0 = 0$, as this corresponds to $y[n]$.

$$y[n] = \sum_{i=0}^{N} a_i x[n-i] + \sum_{i=1}^{N} b_i y[n-i] \tag{2}$$

2 FFT Convolution

1. Write a short `octave` script that convoles a *gaussian pulse* with a square pulse. A gaussian pulse is a sine wave with a gaussian envelope. This calculation arises in certain branches of quantum mechanics. Sketch the output below:

3 LP and HP Recursive Filters

1. Suppose we want to implement a **low-pass** recursive filter (Fig. 1). (a) If our sampling rate f_s is 10 kHz, and we need a cutoff frequency f_C of 1 kHz, what is the ratio of these? This is the unitless cutoff frequency. (b) Find $x = \exp(-2\pi f_C)$, using the unitless cutoff frequency. (c) Calculate $a_0 = 1 - x$, and $b_1 = x$, with all other a_i and b_i set to zero. (d) Simplify Eq. 2 to account for the constraints on a_i and b_i. (e) Implement this in an `octave` script, and find the step response.

2. Repeat the previous exercise, but for a **high-pass** recursive filter (Fig. 2), using $a_0 = (1+x)/2$, $a_1 = -(1+x)/2$, and $b_1 = x$. Use the same unitless cutoff frequency.

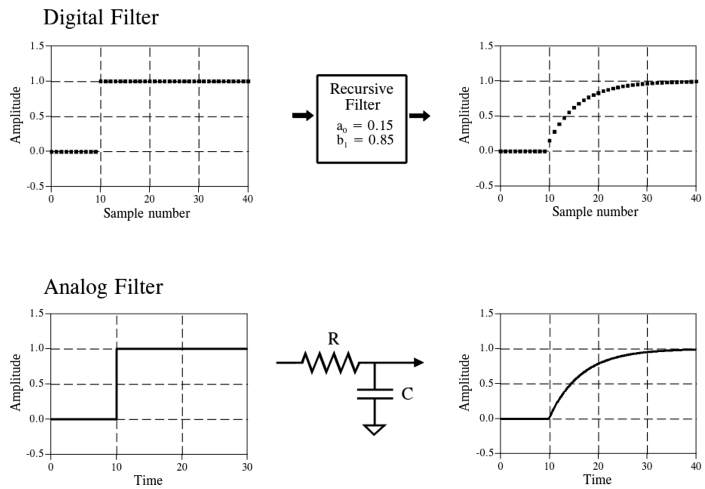

Figure 1: A low-pass RC-like recursive filter.

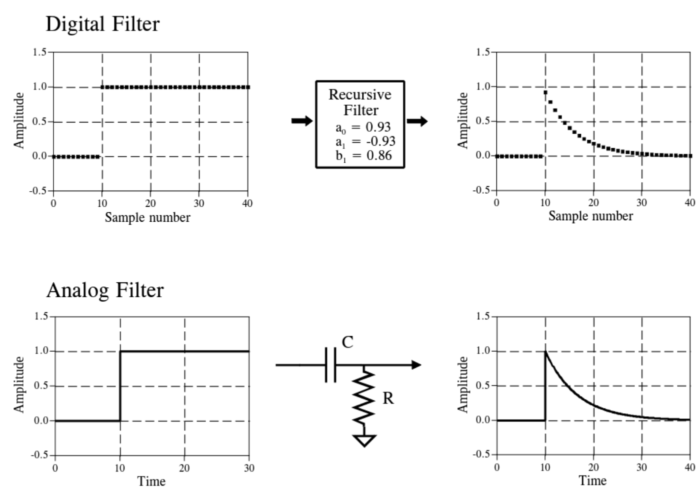

Figure 2: A high-pass RC-like recursive filter.