Tuesday Warm Up, Unit 1: Filter Design

Prof. Jordan C. Hanson

April 8, 2025

1 Memory Bank

1. **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 LP and HP Recursive Filters

1. **Low-pass recursive filter**. (a) Suppose our sampling rate f_{s} is 10 kHz, and the low-pass cutoff frequency f_{C} is 3 kHz. (b) Find $x = \exp(-2\pi(f_{\mathrm{C}}/f_{\mathrm{s}}))$. (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 filter in an `octave` script.

2. **Low-pass recursive filter**. (a) Repeat the previous exercise, but for a high-pass recursive filter, using $a_0 = (1+x)/2$, $a_1 = -(1+x)/2$, and $b_1 = x$. Use $f_{\mathrm{C}} = 2$ kHz. Implement this filter in an `octave` script.

3 Isolating AM Signals

1. Suppose we are designing a DSP system to isolate the audio data in an AM signal. (a) Create an `octave` script that mixes (multiplies) a 1.4 MHz carrier with a 2.5 kHz audio tone and adds some gaussian white noise. (b) What frequencies should be present in the spectrum, other than noise? (c) Now *unmix* the signals using an LO (local oscillator) with a frequency equal to the carrier, and use the recursive LP and HP filters from the first two exercises to isolate the signal.

4 Phase Response

1. (a) Create a square pulse in `octave`, and run it through the recursive LP filter. (b) Plot the *phase* of the output versus frequency. (c) Do the same for the LP filter. (d) Are the results *linear* or non-linear?

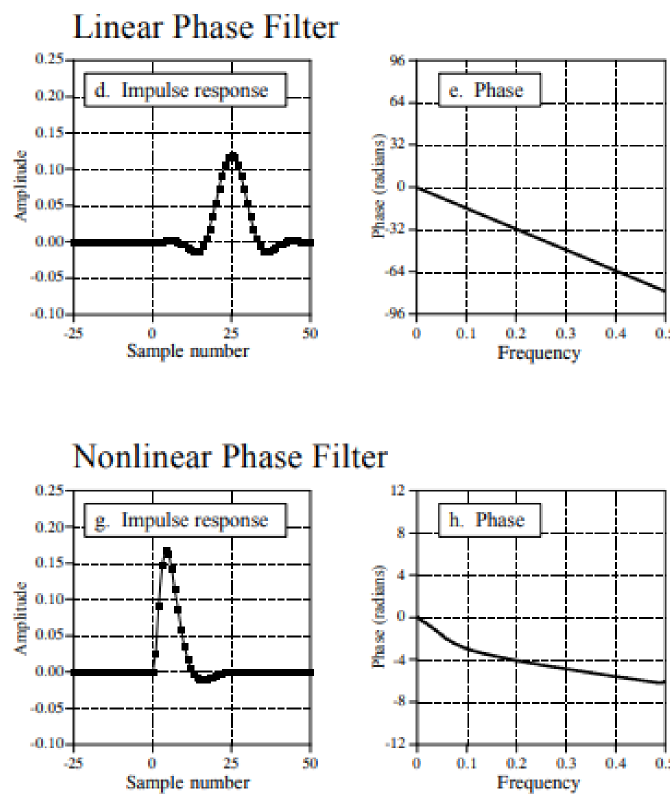

Figure 1: (Top) A linear phase filter has left-right symmetry. (Bottom) A non-linear phase filter does not have left-right symmetry.

2. (a) Use the `flipud` or the `fliplr` functions in `octave` to reverse the LP and HP outputs. (b) Finally, run the *flipped* outputs through the LP and HP filters once more. (c) Plot the phase versus frequency of each output. (d) Are the results linear?