Filter Design Cheatsheet

February 15, 2015

1 FIR Filter Design

This cheatsheet is compiled from book Digital Signal Processing with MATLAB by Ingle & Proakis. Page references are given as #PageNo

Problem Statement Design a lowpass filter that has a passband $[0, \omega_p]$ with tolerance δ_1 (or R_p in dB) and a stopband $[\omega_s, \pi]$ with tolerance δ_2 (or A_s in dB) — see Figure 1.1.3b,c.
In FIR we focus on linear-phase filters:

$$H(e^{j\omega}) = H_r(\omega)e^{j(\beta-\alpha\omega)} \tag{1}$$

where α is *constant group delay* and H_r is the amplitude. The impulse response of linear-phase filters is symmetric or antisymmetric about *alpha*, therefore $\alpha = \frac{M-1}{2}$ and $h(n) = h(M-1-n)$ where M is the filter length. A central goal in FIR filter design is to keep M minimal. Based on whether the filter is symmetric or anti-symmetric, and whether M is odd or even, there are four types of filters well-accepted in the literature. Each of these filters has a different usage.:

	M	symm/anti-symm	β	usage
Type I	odd	symm	0	any
Type I	even	anti-symm	0	lowpass only (not highpass or bandstop (see #233))
Type I	odd	symm	$\pi/2$	digital Hilbert transformers and differentiators (to take derivative)
Type I	even	anti-symm	$\pi/2$	digital Hilbert transformers and differentiators (to take derivative)

Table 1: Filter types

1.1 Design approaches

1. Windowing an ideal LP filter
2. Designing directly in Frequency Domain
3. Optimal equiripple design

The first two are intuitive but not optimal. The third has a more sophisticated theory but is optimal. Optimality here means minimal M for given design specs (*i.e.* desired $\omega_p, \omega_s, R_p, A_s$ values).

1.1.1 Windowing an ideal lowpass filter

An ideal lowpass (LP) filter (see Fig. 1.1.3a) must be infinite. To make it FIR we must truncate the ideal filter, which will introduce ripples around cutoff frequency (see Fig. 1.1.3b) due to Gibbs phenomenon (see #245 onwards). The truncation is done by windowing, *i.e.* multiplication (or conv in freq domain) with a windowing function. The goal is to find the windowing function that meets the design specs with minimal M. The simplest and worse-performing windowing fn is boxcar. More sophisticate ones are triangular, Hanning, Hamming and Blackman. Hamming is typically used. See #251 for comparison among windows.

1.1.2 Designing directly in Frequency Domain

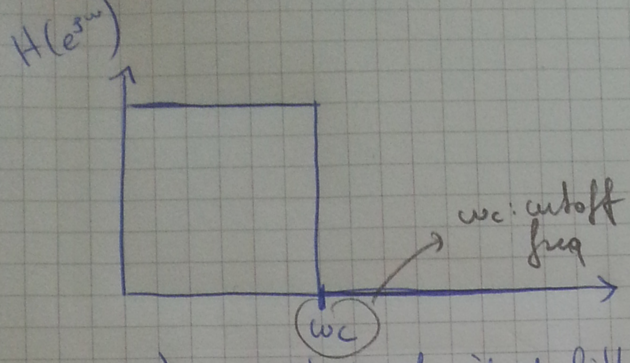
We manually design a sequence in frequency domain such as $H(k) = \{1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0\}$ (number of 1's is proportional to ω_c). This would be the freq response of the ideal filter. To minimize ripples, we manually introduce a slope such as $H(k) = \{1, 1, 1, T_1, T_2, 0, 0, 0, 0\}$. We decide the number of T_i values, then the goal is to find the optimal T_i values. See #264 onwards.

1.1.3 Optimal Equiriple Design

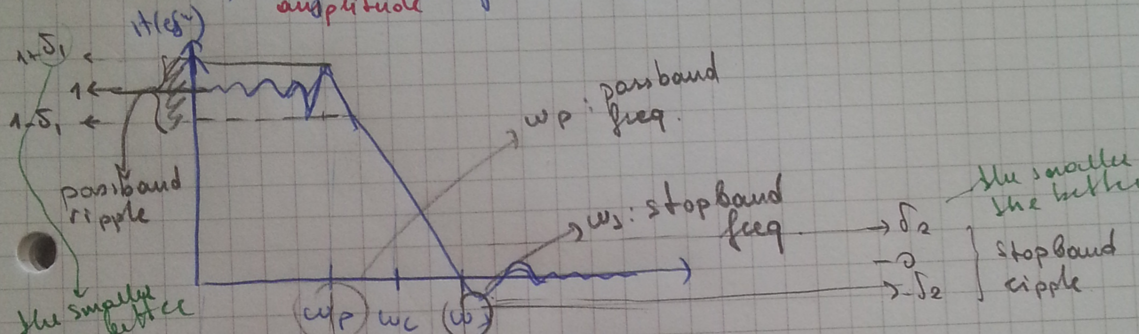
It turns out that the optimal design is reached by distributing the error around ripples uniformly (in contrary to having an increasing error nearer the band edges). This can be performed with by solving what is called a *minimax problem* (see #278 onwards). This requires a polynomial approximation which is performed via the Parks-McClellan algorithm (#284).
All these can be done very simply in MATLAB through the `remez` function. The overall design is as follows:

1. Decide specs: $\omega_p, \omega_s, R_p, A_s$
2. Guess an initial filter length $\hat{M}$ (see code and 7.48 in #284)
3. Run `remez`
4. check A_s, if OK stop, if not, increase $\hat{M}$ and repeat until desired A_s reached. Bear in mind that the restrictions on M based on the filter usage (see Table 1) still apply and therefore M should be incremented carefully (*i.e.* maintain an either odd or even value).

Exemlar piece of code:

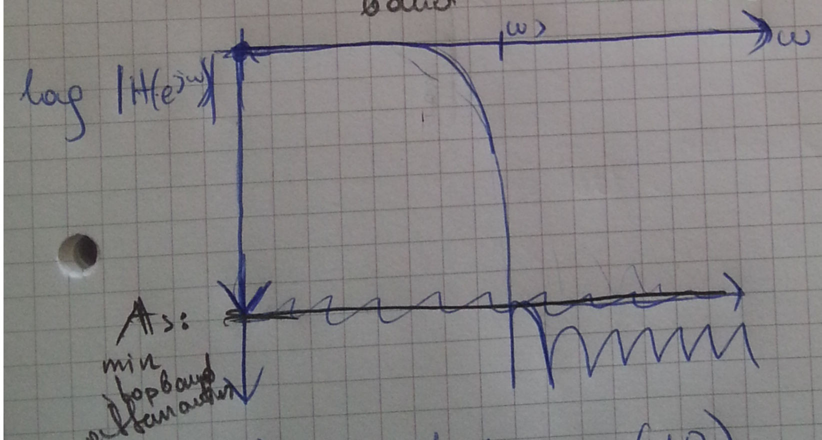


a) ~~magnitude~~ amplitude of ideal filter



b) ~~magnitude~~ amplitude of a filter in practice

Transition band : the narrower the better



c) magnitude resp (dB)

the larger the better. Typically at least 50dB is good.