Matched Two-Pole Digital Shelving Filters

Martin Vicanek

23. March 2024

1 Introduction

In digital audio processing, equalization is used as an enhancement or suppression of certain frequencies in order to compensate for spectral distortions in the transmission chain, or to account for room acoustics, or simply for personal preference. Filtering is often performed with recursive digital filters, aka IIR filters. Common designs of such filters often have an unwanted transfer function cramping towards high frequencies. This is shown in figure 1.

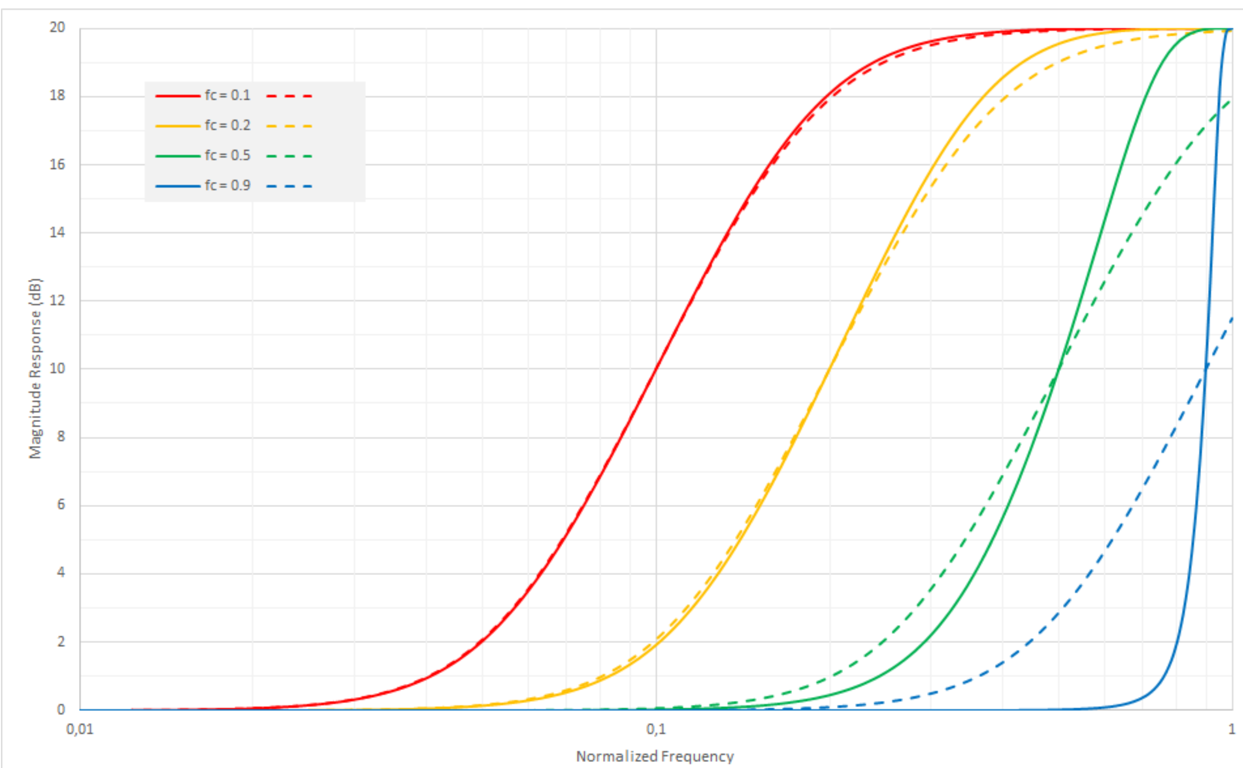

Figure 1: Magnitude responses of various high-shelf filters. Solid lines: digital filter using bi-linear transform. Dashed lines: analog prototype. In all cases the high-shelf gain is 20 dB.

In this note I present a design scheme for the specific class of second order Butterworth shelving filters, with the objective to closely match the magnitude transfer curve of its analog counterpart.

2 Shelving Filters

Shelving filters have some gain factor G_0 for low frequencies, some other gain factor G_1 for high frequencies, and a transition region around some characteristic frequency f_c. For the sake of definiteness consider a high-shelf with $G_0 = 1$. Since G_1 is the only remaining gain parameter, we may omit the subscript and simply use G for the high shelf gain. In the analog domain, the transfer function of a second order Butterworth high-shelf filter is [1]

$$H(s) = \frac{1 + \sqrt{2}gs + g^2s^2}{1 + \sqrt{2}s/g + s^2/g^2}, \tag{1}$$

where $g = G^{\frac{1}{4}}$, $s = if/f_c$, $i = \sqrt{-1}$ and f is the frequency. A Butterworth filter has, for a given filter degree, the sharpest possible passband-stopband transition without overshoots or ripples in the magnitude response.

In the digital domain, a general second order filter (aka biquad) has a transfer function

$$H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{a_0 + a_1z^{-1} + a_2z^{-2}} \tag{2}$$

where $z = \exp(i\pi f)$. We have chosen to denote the frequency f in units of the Nyquist frequency. Furthermore, we are free to choose $a_0 = 1$.

Given the filter specification in terms of f_c and G, the objective is to find suitable filter coefficients a_1, a_2, b_0, b_1, and b_2 so that the magnitudes of the analog and the digital filter match.

3 Filter Design

Take the modulus squared of transfer function in equation (2). The result may be written as [2]

$$|H(z)|^2 = \frac{B_0(1-\phi) + B_1\phi + 4B_2\phi(1-\phi)}{A_0(1-\phi) + A_1\phi + 4A_2\phi(1-\phi)} \tag{3}$$

Here we have introduced

$$\phi = \sin^2(\tfrac{\pi}{2}f) \tag{4}$$

as a monotonic function of f, which goes from 0 at $f = 0$ (or DC for short) to 1 at the Nyquist frequency $f = 1$. The A_0, A_1, A_2 and B_0, B_1, B_2 coefficients in eq.(3) are related to the a_0, a_1, a_2 and b_0, b_1, b_2 coefficients in eq.(2) by the following set of equations [2]:

$$\begin{aligned} V &= \tfrac{1}{2}\left(\sqrt{A_0}+\sqrt{A_1}\right) & W &= \tfrac{1}{2}\left(\sqrt{B_0}+\sqrt{B_1}\right) \\ a_0 &= \tfrac{1}{2}\left(V+\sqrt{V^2+A_2}\right) & b_0 &= \tfrac{1}{2}\left(W+\sqrt{W^2+B_2}\right) \\ a_1 &= 1-V & b_1 &= 1-W \\ a_2 &= -\tfrac{1}{4}A_2/a_0 & b_2 &= -\tfrac{1}{4}B_2/b_0 \end{aligned} \tag{5}$$

We may re-normalize all a_i and b_i coefficients by dividing them by a_0 without changing the result in eq.(2).

As the first matching condition, we require the DC filter magnitude response to be unity, which results in $A_0 = B_0$. We are free to choose $A_0 = B_0 = 1$.

Since both the numerator and denominator in eq.(3) are second order polynomials of ϕ, we may write it in a slightly different form,

$$|H(z)|^2 = \frac{1-\phi+\beta_1\phi(1-\phi)+\beta_2\phi^2}{1-\phi+\alpha_1\phi(1-\phi)+\alpha_2\phi^2}, \tag{6}$$

where α_i and β_i are related to the A_i and B_i coefficients, respectively, by

$$\begin{aligned} A_1 &= \alpha_2 & A_2 &= \tfrac{1}{4}(\alpha_1-\alpha_2) \\ B_1 &= \beta_2 & B_2 &= \tfrac{1}{4}(\beta_1-\beta_2) \end{aligned} \tag{7}$$

Our second matching condition will be maximum flatness of the filter magnitude response at DC, as is the case for a Butterworth filter. Technically, this means that $|H(z)|^2$ has zero slope in terms of ϕ at $\phi = 0$, which translates to

$$\beta_1 = \alpha_1. \tag{8}$$

A third condition is that the magnitude response of the digital filter shall match the analog value at Nyquist frequency. Evaluating, thus, eq.(6) at $\phi = 1$ yields

$$\beta_2 = h_{\mathrm{Ny}}\alpha_2, \tag{9}$$

where h_{Ny} is the magnitude squared at Nyquist. From eq.(1) we obtain an expression for h_{Ny} in terms of the filter design parameters f_c and G,

$$h_{\mathrm{Ny}} = \frac{f_c^4+G}{f_c^4+1/G}. \tag{10}$$

Up to now, we have imposed three conditions on the biquad filter response. As a result, we can use equations (8) and (9) to eliminate β_1 and β_2 from eq.(6). This leaves us with two remaining parameters α_1 and α_2, which will be determined by two more matching requirements at some frequencies f_1 and f_2. These two matching points should be chosen to warrant a good match across the entire Nyquist interval. A particular challenge is to make sure that all roots in eq.(5) be real. Based on a thorough numerical analysis, the author proposes the following choice:

$$f_1 = \frac{f_c}{\sqrt{0.160 + 1.543 f_c^2}}, \quad f_2 = \frac{f_c}{\sqrt{0.947 + 3.806 f_c^2}}. \tag{11}$$

Eq.(11) ensures that f_1 and f_2 always remain below Nyquist, even for f_c above Nyquist.

Applying the matching conditions at f_1 and f_2 is straightforward and leads to a set of linear equations for α_1 and α_2,

$$\begin{aligned} c_{11}\alpha_1 + c_{12}\alpha_2 &= d_1 \\ c_{21}\alpha_1 + c_{22}\alpha_2 &= d_2, \end{aligned} \tag{12}$$

where the c_{ij} and d_i are given by

$$\begin{aligned} d_1 &= (h_1 - 1)(1 - \phi_1) & d_2 &= (h_2 - 1)(1 - \phi_2) \\ c_{11} &= -\phi_1 d_1 & c_{21} &= -\phi_2 d_2 \\ c_{12} &= (h_{\mathrm{Ny}} - h_1)\phi_1^2 & c_{22} &= (h_{\mathrm{Ny}} - h_2)\phi_2^2 \end{aligned} \tag{13}$$

The abbreviations ϕ_i and h_i denote the following expressions:

$$\begin{aligned} \phi_1 &= \sin^2(\tfrac{\pi}{2} f_1) & \phi_2 &= \sin^2(\tfrac{\pi}{2} f_2) \\ h_1 &= \frac{f_c^4 + f_1^4 G}{f_c^4 + f_1^4 / G} & h_1 &= \frac{f_c^4 + f_1^4 G}{f_c^4 + f_1^4 / G} \end{aligned} \tag{14}$$

Eq.(12) is easily solved for α_1 and α_2,

$$\alpha_1 = \frac{c_{22} d_1 - c_{12} d_2}{c_{11} c_{22} - c_{12} c_{21}}, \quad \alpha_2 = \frac{d_1 - c_{11}\alpha_1}{c_{12}}. \tag{15}$$

Having determined α_1 and α_2, we can use equations (8) and (9) to calculate β_1 and β_2. Then, eq.(7) yields the quantities A_i and B_i, which in turn determine the biquad coefficients a_i and b_i using eq.(5).

Figure 2 shows some magnitude responses as proposed in this work in comparison to the analog high-shelf filter responses, respectively. The overall agreement is fair, with deviations at high frequencies when the center frequency is close to Nyquist. These deviations never exceed 1 dB for a shelf gain of 20 dB. Note that the matching scheme also works for shelves with center frequencies above Nyquist.

4 Other Filter Types

The results of the previous section may be applied to other filter types with some adaptation. For instance, a low-shelf filter may be viewed as a high shelf filter with gain $1/G$ instead of G and the b_i coefficients scaled by G to ensure unity gain at high frequencies.

In order to facilitate use of the derived formulas, we present pseudo-codes in the appendix for an implementation of high- and low-shelf filters.

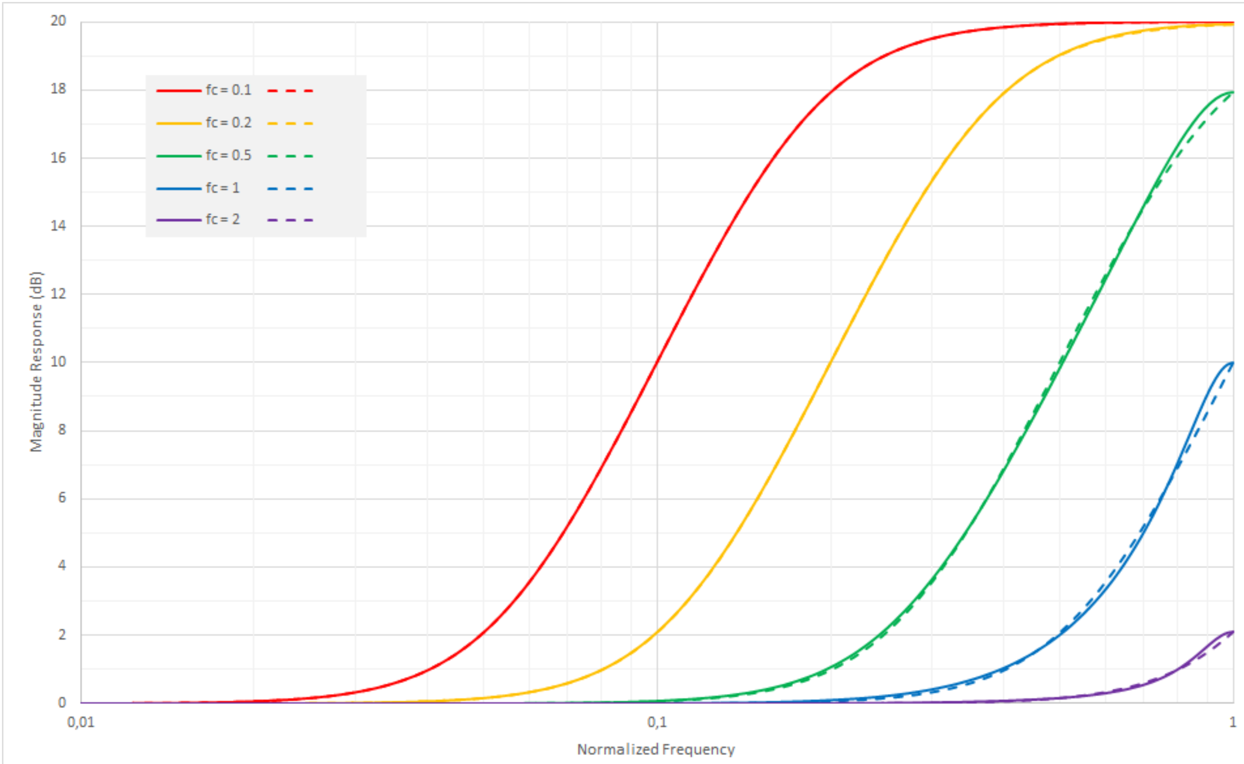

Figure 2: Magnitude responses of various high-shelf filters. Solid lines: present work. Dashed lines: analog prototype.

5 Conclusion

In this article we provide closed-form expressions for the design of second-order digital shelving filters. The resulting magnitude transfer functions match the analog prototype quite well over the entire audio range, even for f_c above Nyquist.

References

[1] W3C Working Group, Audio EQ Cookbook. https://www.w3.org/TR/audio-eq-cookbook/

[2] Martin Vicanek, Matched Second Order Digital Filters. https://vicanek.de/articles/BiquadFits.pdf

A Pseudocodes

A.1 Matched High Shelf Filter

```
// Matched 2-Pole Butterworth High Shelf Filter
// ---------------------------------------------

// Inputs:
// fc (center frequency/ Nyquist frequency)
// gain (shelf gain factor)

// Outputs:
// a1, a2, b0, b1, b2 (biquad filter coefficients)

// special case gain = 1 (flat response)
if ( abs(1 - gain) < 1e-6 ) then
g = 1.00001
else
g = gain

// abbreviations
pihalf = 1.5708
invg   = 1.0/g

// matching gain at Nyquist
fc4  = fc^4
hny = (fc4 + g)/(fc4 + invg)

// matching gain at f_1
f1   = fc/sqrt(0.160 + 1.543*fc*fc)
f14  = f1^4
h1   = (fc4 + f14*g)/(fc4 + f14*invg)
phi1 = sin(pihalf*f1)^2

// matching gain at f_2
f2   = fc/sqrt(0.947 + 3.806*fc*fc)
f24  = f2^4
h2   = (fc4 + f24*g)/(fc4 + f24*invg)
phi2 = sin(pihalf*f2)^2

// linear equations coefficients
d1  = (h1 - 1.0)*(1.0 - phi1)
c11 = -phi1*d1
c12 = phi1*phi1*(hny - h1)
```

```
d2  = (h2 - 1.0)*(1.0 - phi2)
c21 = -phi2*d2
c22 = phi2*phi2*(hny - h2)

// linear equations solution
alfa1 = (c22*d1 - c12*d2)/(c11*c22 - c12*c21)
aa1 = (d1 - c11*alfa1)/c12
bb1 = hny*aa1

// compute A_2 and B_2
aa2 = 0.25*(alfa1 - aa1)
bb2 = 0.25*(alfa1 - bb1)

// compute biquad coefficients scaled with 1/a_0
v  = 0.5*(1.0 + sqrt(aa1))
w  = 0.5*(1.0 + sqrt(bb1))
a0 = 0.5*(v + sqrt(v*v + aa2))
inva0 = 1.0/a0

a1 = (1.0 - v)*inva0
a2 = -0.25*aa2*inva0*inva0

b0 = (0.5*(w + sqrt(w*w + bb2)))*inva0
b1 = (1.0 - w)*inva0
b2 = (-0.25*bb2/b0)*inva0*inva0
```

A.2 Matched Low Shelf Filter

```
// Matched 2-Pole Butterworth Low Shelf Filter
// ------------------------------------------

// Inputs:
// fc (center frequency/ Nyquist frequency)
// gain (shelf gain factor)

// Outputs:
// a1, a2, b0, b1, b2 (biquad filter coefficients)

// special case gain = 1 (flat response)
if ( abs(1 - gain) < 1e-6 ) then
g = 1.00001
else
g = 1.0/gain

// abbreviations
pihalf = 1.5708
invg   = 1.0/g

// matching gain at Nyquist
fc4  = fc^4
hny = (fc4 + g)/(fc4 + invg)

// matching gain at f_1
f1   = fc/sqrt(0.160 + 1.543*fc*fc)
f14  = f1^4
h1   = (fc4 + f14*g)/(fc4 + f14*invg)
phi1 = sin(pihalf*f1)^2

// matching gain at f_2
f2   = fc/sqrt(0.947 + 3.806*fc*fc)
f24  = f2^4
h2   = (fc4 + f24*g)/(fc4 + f24*invg)
phi2 = sin(pihalf*f2)^2

// linear equations coefficients
d1  = (h1 - 1.0)*(1.0 - phi1)
c11 = -phi1*d1
c12 = phi1*phi1*(hny - h1)

d2  = (h2 - 1.0)*(1.0 - phi2)
c21 = -phi2*d2
```

```
c22 = phi2*phi2*(hny - h2)

// linear equations solution
alfa1 = (c22*d1 - c12*d2)/(c11*c22 - c12*c21)
aa1 = (d1 - c11*alfa1)/c12
bb1 = hny*aa1

// compute A_2 and B_2
aa2 = 0.25*(alfa1 - aa1)
bb2 = 0.25*(alfa1 - bb1)

// compute biquad coefficients scaled with 1/a_0
v  = 0.5*(1.0 + sqrt(aa1))
w  = 0.5*(1.0 + sqrt(bb1))
a0 = 0.5*(v + sqrt(v*v + aa2))
inva0 = 1.0/a0

a1 = (1.0 - v)*inva0
a2 = -0.25*aa2*inva0*inva0

b0 = gain*(0.5*(w + sqrt(w*w + bb2)))*inva0
b1 = gain*(1.0 - w)*inva0
b2 = gain*(-0.25*bb2/b0)*inva0*inva0
```