

Bi-quad filters realization on AIC Codecs

HPA Software

ABSTRACT

This application report provides information regarding filter equations and coefficient format representation that can be used to realize digital filters on the AIC3x miniDSP platform. It also explains ways to update the filter coefficients on the fly using a host processor.

Contents

1	Standard Bi-quad filter equations	2
1.1	All Pass (Phase shift) filters	2
1.2	Equalization filters	3
1.3	Notch Filters	5
1.4	Treble Shelf	7
1.5	Bass Shelf Filters	9
1.6	Second order Linkwitz Riley	12
1.7	Second Order Variable Q Filter	13
1.8	Second order Butterworth Filter from Variable Q	15
1.9	Second order Bessel Filter from Variable Q	15
1.10	First Order Butterworth Filters	15
1.11	Second order Chebychev	17
2	Number representation format for filter coefficients	21
2.1	Filter coefficient normalization	22
3	Updating the codec filter coefficients	23
Appendix A. Bi-quad coefficients computation example		25

Figures

Figure 1.	Phase shift filter phase response ($F_c=1200\text{Hz}$, $BW=300\text{Hz}$)	2
Figure 2.	EQ filter with $F_c=1\text{kHz}$, $BW=100\text{Hz}$, $G=6\text{dB}$	4
Figure 3.	Notch filter magnitude response with $F_c=1\text{kHz}$, $BW=100\text{Hz}$	6
Figure 4.	Treble shelf filter magnitude response ($F_c=2\text{kHz}$, $G=-6\text{dB}$)	8
Figure 5.	Bass shelf filter magnitude response ($F_c=500\text{Hz}$, $G=6\text{dB}$)	10
Figure 6.	Second order Linkwitz Riley ($F_c=500\text{Hz}$, $HL=high$)	12
Figure 7.	Second order variable Q filter ($F_c=1200\text{Hz}$, $Q=1$, $HL=low$)	14
Figure 8.	First order Butterworth ($F_c=1200\text{Hz}$, $HL=low$)	16
Figure 9.	Second order Chebychev ($F_c=600\text{Hz}$, $typ=high$, $rip=3\text{dB}$)	19
Figure 10.	Normalizing filter response	23

Tables

Table 1.	EQ Filter coefficient table in host controller for “enhanced” codecs	24
-----------------	---	-----------

1 Standard Bi-quad filter equations

1.1 All Pass (Phase Shift) filters

Filter parameters

BW = Bandwidth in Hz

F_c = Center frequency in Hz

F_s = Sample frequency in Hz

Error Checking

$$0 \leq BW \leq \frac{F_s}{2}$$

$$0 \leq F_c \leq \frac{F_s}{2}$$

$$0 \leq F_s \leq 192K$$

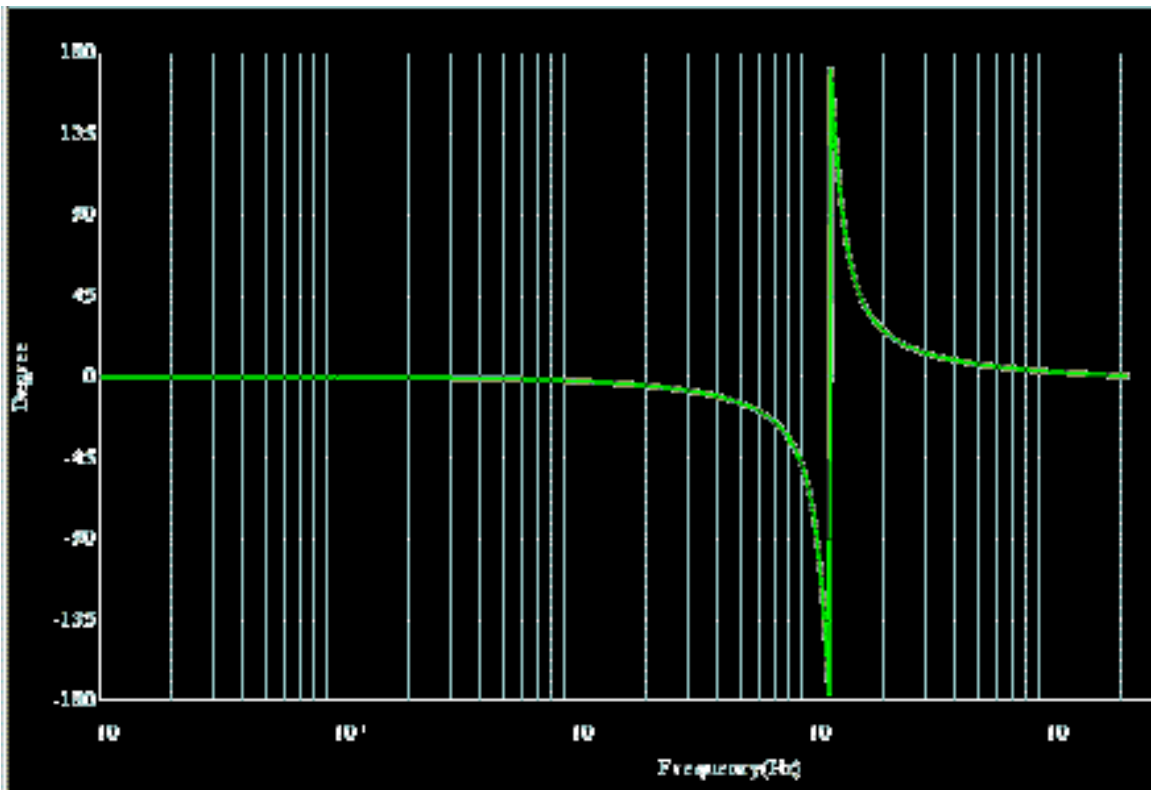


Figure 1. Phase shift filter phase response ($F_c = 1200\text{ Hz}$, $BW = 300\text{ Hz}$)

Equation

$$a = \frac{1 - \tan\left(\pi \times \frac{BW}{F_s}\right)}{1 + \tan\left(\pi \times \frac{BW}{F_s}\right)}$$

$$d = -\cos\left(2 \times \pi \times \frac{F_c}{F_s}\right)$$

FilterCoefficients

$$b0 = a$$

$$b1 = d \times (1 + a)$$

$$a1 = b1$$

$$b2 = 1$$

$$a2 = a$$

$$B = [b0 \ b1 \ b2]$$

$$A = [1 \ a1 \ a2]$$

1.2 Equalizationfilters

FilterParameters

BW=FilterBandwidthinHz

F_c=FilterCenterFrequencyinHz

F_s=SamplerateinHz

G=FilterGainindB

ErrorChecking

$$0 \leq BW \leq \frac{F_s}{2}$$

$$0 \leq F_c \leq \frac{F_s}{2}$$

$$0 \leq F_s \leq 192K$$

$$-140 \leq G \leq 48$$

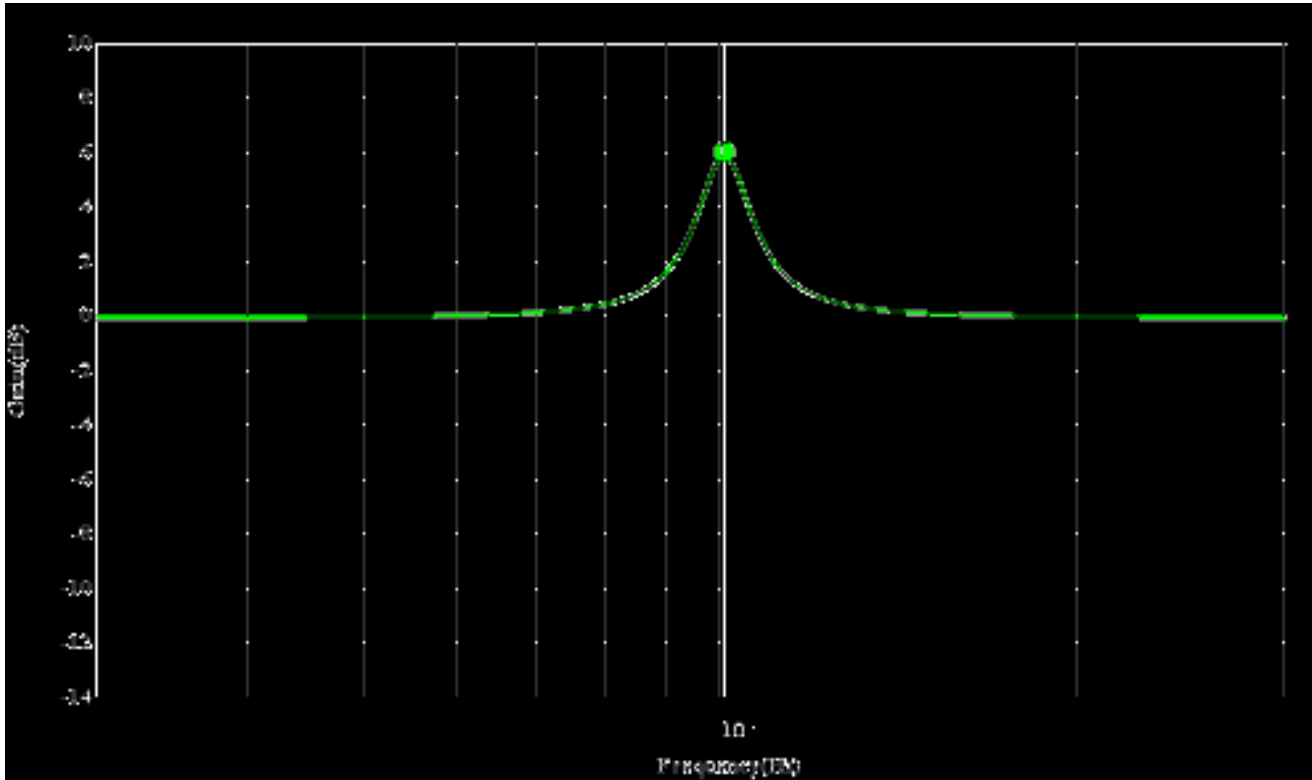


Figure2. EQfilterwith $F_c=1\text{kHz}, BW=100\text{Hz}, G=6\text{dB}$

Equation

$$A = 10^{\frac{G}{20}}$$

$$\text{if}(A < 1)$$

$$a = \frac{\left(\tan\left(\pi \times \frac{BW}{F_s} \right) - A \right)}{\left(\tan\left(\pi \times \frac{BW}{F_s} \right) + A \right)}$$

else

$$a = \frac{\left(\tan\left(\pi \times \frac{BW}{F_s} \right) - 1 \right)}{\left(\tan\left(\pi \times \frac{BW}{F_s} \right) + 1 \right)}$$

$$H = A - 1$$

$$d = -\cos\left(2 \times \pi \times \frac{F_c}{F_s} \right)$$

FilterCoefficients

$$b0 = 1 + (1 + a) \times \frac{H}{2}$$

$$b1 = d \times (1 + a)$$

$$b2 = \left(-a - (1 + a) \times \frac{H}{2} \right)$$

$$a1 = b1$$

$$a2 = -a$$

$$B = [b0 \ b1 \ b2]$$

$$A = [1 \ a1 \ a2]$$

1.3 NotchFilters

FilterParameters

BW=NotchBandwidthinHz

F_c=NotchCenterfrequencyinHz

F_s=SamplerateinHz

ErrorChecking

$$0 \leq BW \leq \frac{F_s}{2}$$

$$0 \leq F_c \leq \frac{F_s}{2}$$

$$0 \leq F_s \leq 192K$$

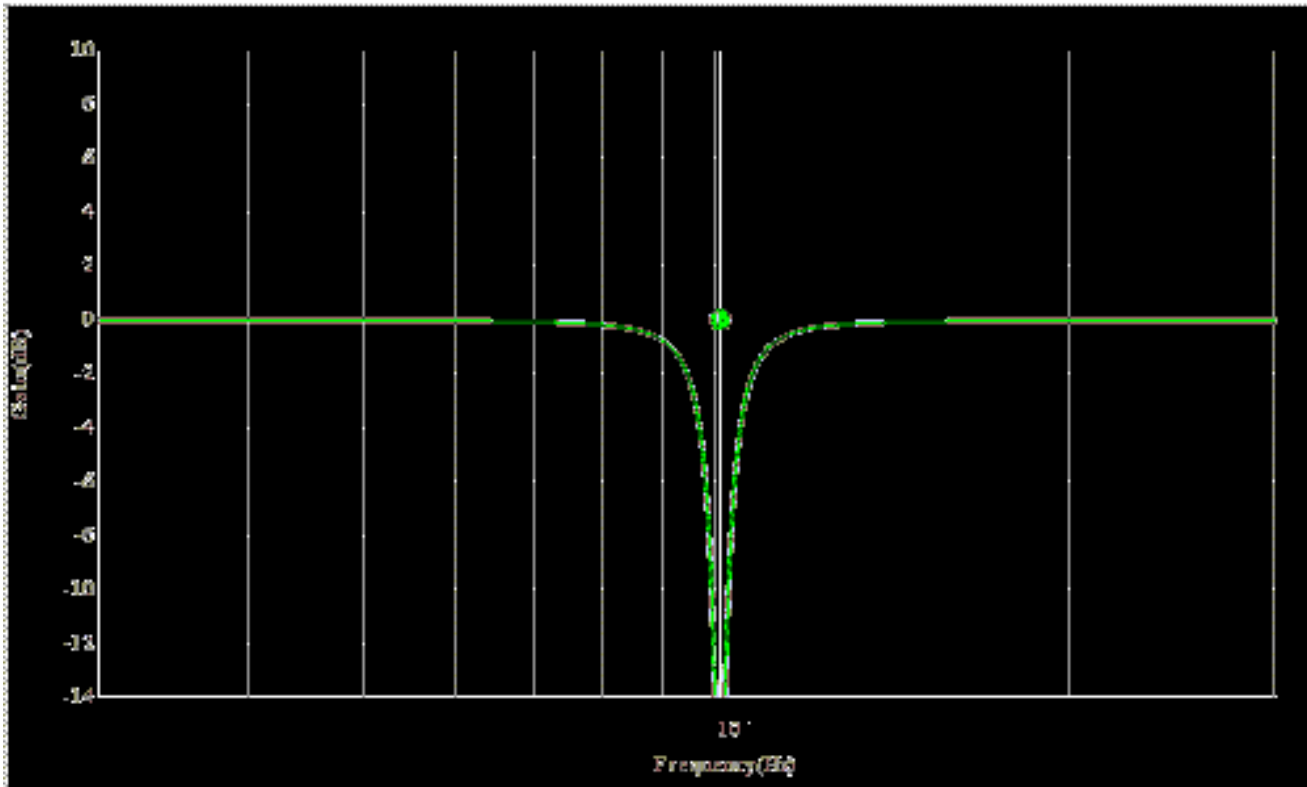


Figure3. Notchfiltermagnituderesponsewith $F_c=1\text{kHz}, BW=100\text{Hz}$

Equation

$$a = \frac{1 - \tan\left(\pi \times \frac{BW}{F_s}\right)}{1 + \tan\left(\pi \times \frac{BW}{F_s}\right)}$$

$$d = -\cos\left(2 \times \pi \times \frac{F_c}{F_s}\right)$$

$$b0 = a$$

$$b1 = d \times (1 + a)$$

$$a1 = b1$$

$$b2 = 1$$

$$a2 = a$$

FilterCoefficients

$$B = [b0 \ b1 \ b2]$$

$$A = [1 \ a1 \ a2]$$

$$B = 0.5 \times (B + A)$$

1.4 TrebleShelf

FilterParameters

F_c =TrebleShelfCornerfrequencyinHz

F_s =SamplerateinHz

G=TrebleShelfGainindB

ErrorChecking

$$0 \leq F_c \leq \frac{F_s}{2}$$

$$0 \leq F_s \leq 192K$$

$$-24 \leq G \leq 24$$

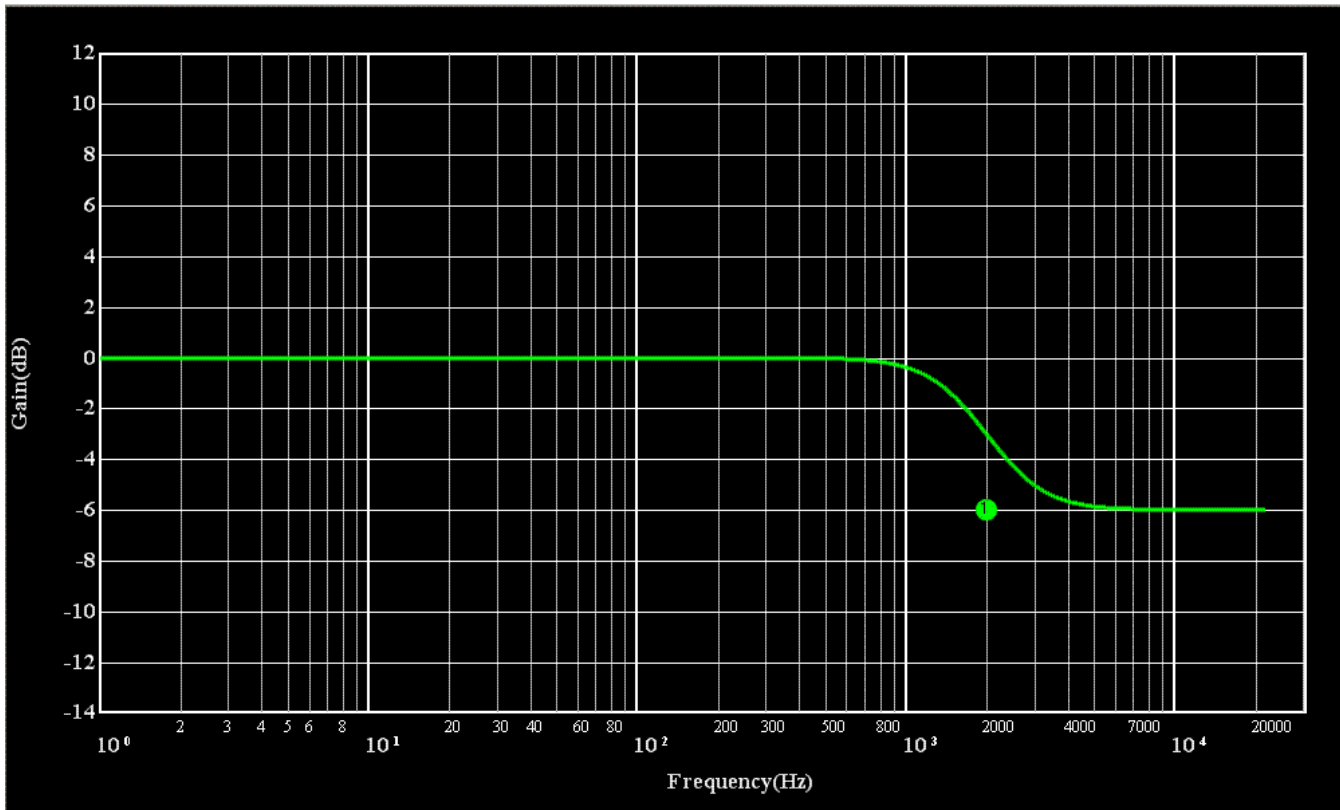


Figure4. Treble shelf filter magnitude response ($F_c=2\text{kHz}, G=-6\text{dB}$)

Equation

$$g = 10^{\frac{G}{20}}$$

$$s = \frac{\sqrt{2}}{2}$$

$$\rho = \frac{\pi}{2}$$

$$\varphi = \frac{F_c}{F_s} \times \pi$$

$$A = g$$

$$G = 20 \times \log_{10}(A)$$


```

If G > -6 & G < 6
    F = sqrt(A)
elseif A > 1
    F = A / sqrt(2)
else
    F = A * sqrt(2)
end

gd = 4 * sqrt((F^2 - 1) / (A^2 - F^2))
gn = sqrt(A) * gd
a = tan(pi * (Fc / Fs - 1/4))

b0 = (gn^2 * a^2 + 2 * s * gn - 2 * gn^2 * a + 1 - 2 * s * gn * a^2 + a^2 + gn^2 + 2 * a) /
(1 + gd^2 + 2 * s * gd - 2 * s * gd * a^2 + gd^2 * a^2 - 2 * gd^2 * a + a^2 + 2 * a)
b1 = (2 - 2 * gn^2 * a^2 + 4 * gn^2 * a + 4 * a - 2 * gn^2 + 2 * a^2) /
(1 + gd^2 + 2 * s * gd - 2 * s * gd * a^2 + gd^2 * a^2 - 2 * gd^2 * a + a^2 + 2 * a)
b2 = (1 + 2 * s * gn * a^2 - 2 * s * gn + 2 * a + a^2 + gn^2 - 2 * gn^2 * a + gn^2 * a^2) /
(1 + gd^2 + 2 * s * gd - 2 * s * gd * a^2 + gd^2 * a^2 - 2 * gd^2 * a + a^2 + 2 * a)
a0 = 1
a1 = (2 - 2 * gd^2 * a^2 + 4 * gd^2 * a + 2 * a^2 - 2 * gd^2 + 4 * a) /
(1 + gd^2 + 2 * s * gd - 2 * s * gd * a^2 + gd^2 * a^2 - 2 * gd^2 * a + a^2 + 2 * a)
a2 = (1 - 2 * gd^2 * a + 2 * a + gd^2 - 2 * s * gd + a^2 + gd^2 * a^2 + 2 * s * gd * a^2) /
(1 + gd^2 + 2 * s * gd - 2 * s * gd * a^2 + gd^2 * a^2 - 2 * gd^2 * a + a^2 + 2 * a)

```

FilterCoefficients

$$B = [b0 \ b1 \ b2]$$

$$A = [a0 \ a1 \ a2]$$

1.5 BassShelfFilters

FilterParameters

F_c =BassShelfCornerfrequencyinHz

F_s =SamplerateinHz

G=BassShelfGainindB

ErrorChecking

$$0 \leq F_c \leq \frac{F_s}{2}$$

$$0 \leq F_s \leq 192K$$

$$-24 \leq G \leq 24$$

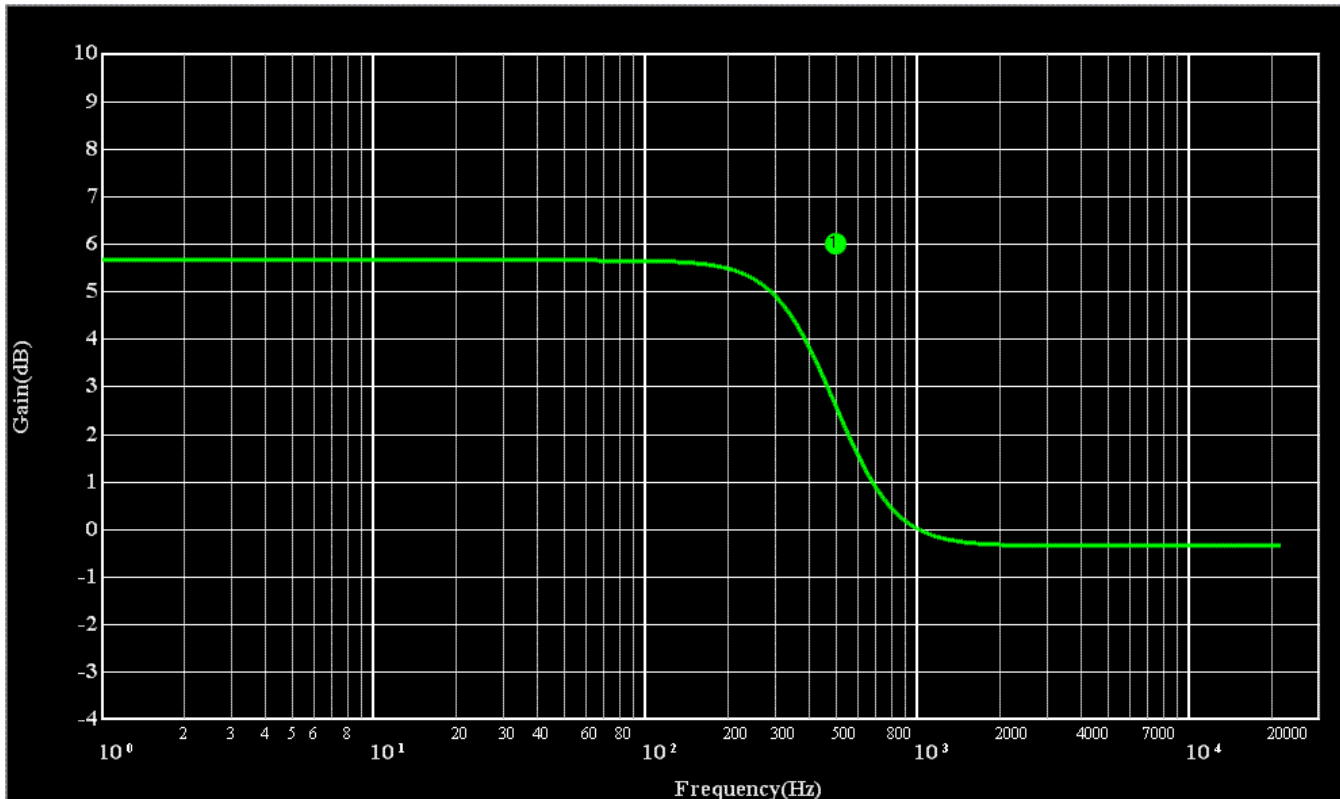


Figure5. Bass shelf filter magnitude response ($F_c=500\text{Hz}, G=6\text{dB}$)

Equation

$$g = 10^{\frac{G}{20}}$$

$$s = \frac{\text{sqrt}(2)}{2}$$

$$\rho = \frac{\pi}{2}$$

$$\varphi = \frac{F_c}{F_s} \times \pi$$

$$A = g$$

$$G = 20 \times \log_{10}(A)$$

If $G > -6$ & $G < 6$

$$F = \text{sqrt}(A)$$

elseif $A > 1$

$$F = \frac{A}{\text{sqrt}(2)}$$

else

$$F = A \times \text{sqrt}(2)$$

end

$$gd = \sqrt[4]{\frac{(F^2 - 1)}{A^2 - F^2}}$$

$$gn = \text{sqrt}(A) \times gd$$

$$a = \tan\left(\pi \times \left(\frac{F_c}{F_s} - \frac{1}{4}\right)\right)$$

$$b0 = \frac{-(-1 - gn^2 \times a^2 - a^2 - 2 \times gn^2 \times a - gn^2 - 2 \times s \times gn + 2 \times s \times gn \times a^2 + 2 \times a)}{2 \times s \times gd + 1 - 2 \times s \times gd \times a^2 + gd^2 \times a^2 + 2 \times gd^2 \times a + a^2 + gd^2 - 2 \times a}$$

$$b1 = \frac{-(2 - 4 \times a - 4 \times gn^2 \times a - 2 \times gn^2 \times a^2 - 2 \times gn^2 + 2 \times a^2)}{2 \times s \times gd + 1 - 2 \times s \times gd \times a^2 + gd^2 \times a^2 + 2 \times gd^2 \times a + a^2 + gd^2 - 2 \times a}$$

$$b2 = \frac{1 + 2 \times s \times gn \times a^2 - 2 \times a + gn^2 - 2 \times s \times gn + 2 \times gn^2 \times a + a^2 + gn^2 \times a^2}{2 \times s \times gd + 1 - 2 \times s \times gd \times a^2 + gd^2 \times a^2 + 2 \times gd^2 \times a + a^2 + gd^2 - 2 \times a}$$

$$a0 = 1$$

$$a1 = \frac{-2 + 2 \times gd^2 \times a^2 + 4 \times gd^2 \times a - 2 \times a^2 + 2 \times gd^2 + 4 \times a}{2 \times s \times gd + 1 - 2 \times s \times gd \times a^2 + gd^2 \times a^2 + 2 \times gd^2 \times a + a^2 + gd^2 - 2 \times a}$$

$$a2 = \frac{gd^2 \times a^2 - 2 \times a + 1 + 2 \times gd^2 \times a - 2 \times s \times gd + a^2 + 2 \times s \times gd \times a^2 + gd^2}{2 \times s \times gd + 1 - 2 \times s \times gd \times a^2 + gd^2 \times a^2 - 2 \times gd^2 \times a + a^2 + gd^2 - 2 \times a}$$

FilterCoefficients

$$B = [b0 \ b1 \ b2]$$

$$A = [a0 \ a1 \ a2]$$

1.6 SecondorderLinkwitzRiley

FilterParameters

F_s = SamplerateinHz

F_c = CutfrequencyinHz

HL=LRFiltertype(high,low)

ErrorChecking

$$0 \leq F_c \leq \frac{F_s}{2}$$

$$0 \leq F_s \leq 192K$$

HL = (high,low)

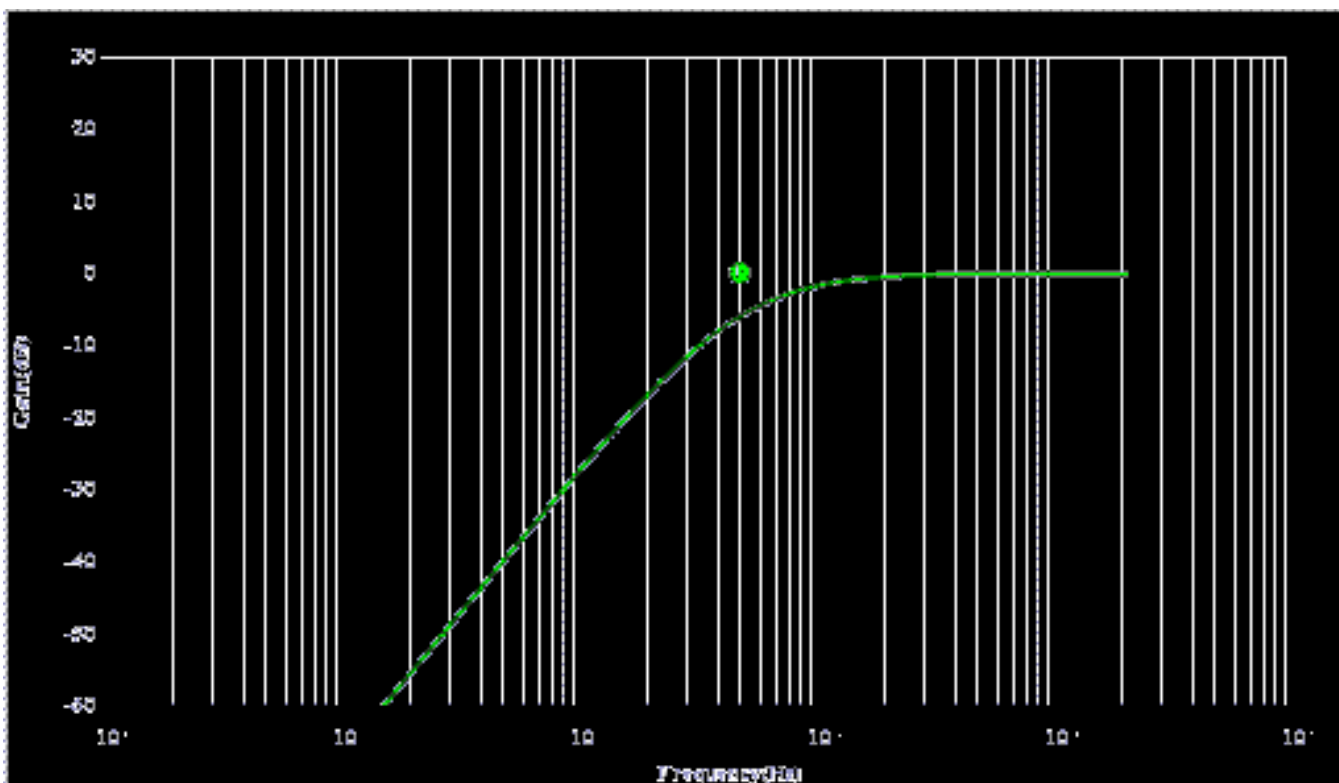


Figure6. SecondorderLinkwitzRiley($F_c=500\text{Hz}$,HL=high)

Equation

```

wc = 2 * pi * Fc
if HL(1:3) == 'low'
    Ba = [0 0 wc^2]
    Aa = [1 2 * wc wc^2]
else
    Ba = [1 0 0]
    Aa = [1 2 * wc wc^2]
end

k = 2 * pi * (Fc / tan(pi * Fc / Fs))

B = [Ba(1) * k^2 + Ba(3) + Ba(2) * k, -2 * Ba(1) * k^2 + 2 * Ba(3), -Ba(2) * k + Ba(1) * k^2 + Ba(3)]
A = [Aa(1) * k^2 + Aa(3) + Aa(2) * k, -2 * Aa(1) * k^2 + 2 * Aa(3), -Aa(2) * k + Aa(1) * k^2 + Aa(3)]

```

FilterCoefficients

$$B = \frac{B}{A(1)}$$

$$A = \frac{A}{A(1)}$$

1.7 SecondOrderVariableQFilter

FilterParameters

F_s = SamplerateinHz

F_c = CutfrequencyinHz

HL = LRFiltertype(high,low)

$$Q = \text{FilterQ} \left(s^2 + \frac{wc}{Q} \times s + wc^2 \right)$$

ErrorChecking

$$0 \leq F_c \leq \frac{F_s}{2}$$

$$0 \leq F_s \leq 192K$$

HL = (high,low)

$$0 \leq Q \leq 100$$

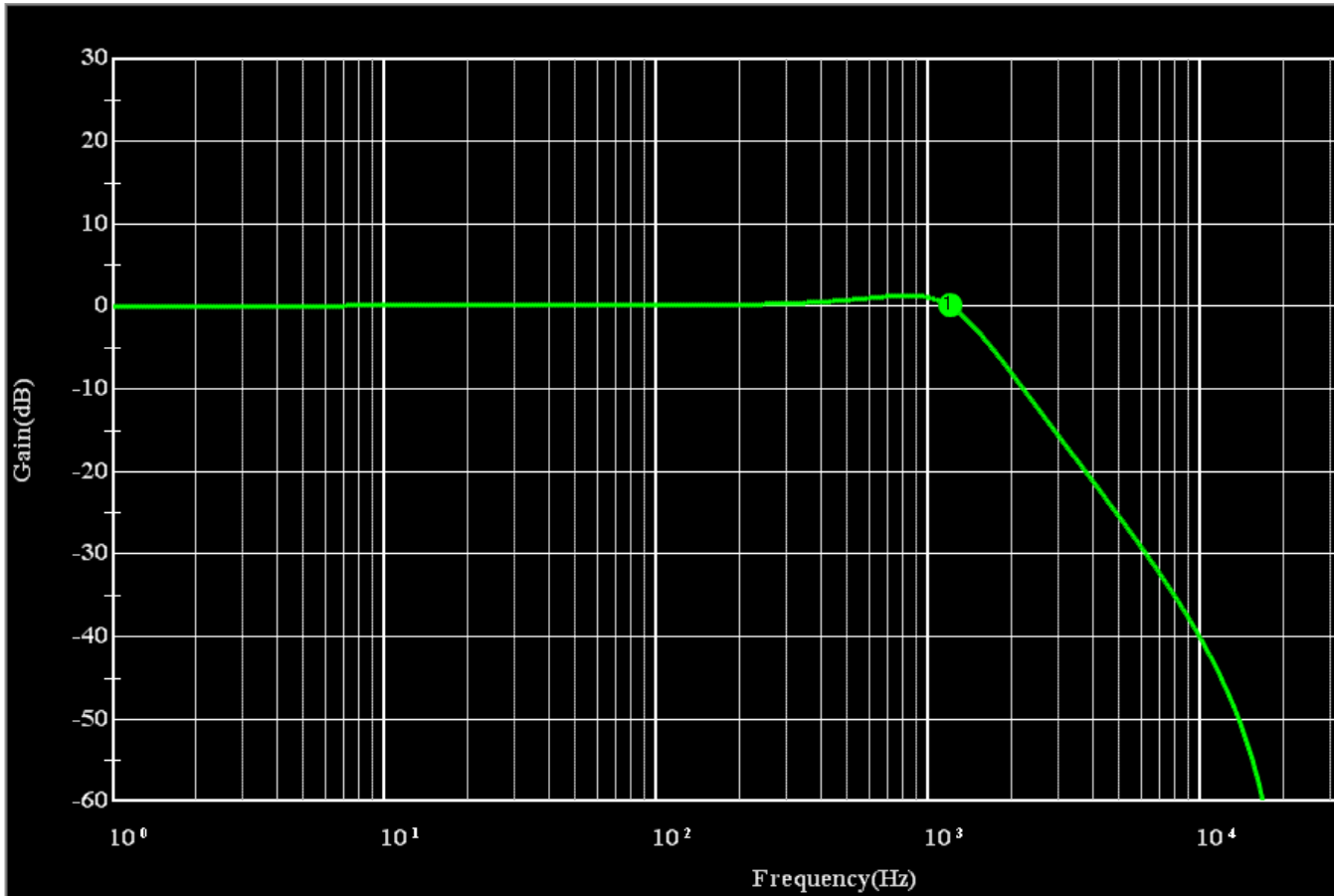


Figure7. SecondordervariableQfilter($F_c=1200\text{Hz}, Q=1, HL=low$)

Equation

```

wc = 2 * pi * Fc
if HL(1:3) == 'low'
    Ba = [0 0 wc^2]
    Aa = [1 wc/Q wc^2]
else
    Ba = [1 0 0]
    Aa = [1 wc/Q wc^2]
end

k = 2 * pi * Fc / tan(pi * Fc / Fs)

B = [Ba(1) * k^2 + Ba(3) + Ba(2) * k, -2 * Ba(1) * k^2 + 2 * Ba(3), -Ba(2) * k + Ba(1) * k^2 + Ba(3)]
A = [Aa(1) * k^2 + Aa(3) + Aa(2) * k, -2 * Aa(1) * k^2 + 2 * Aa(3), -Aa(2) * k + Aa(1) * k^2 + Aa(3)]

```

FilterCoefficients

$$B = \frac{B}{A(1)}$$

$$A = \frac{A}{A(1)}$$

1.8 SecondorderButterworthFilterfromVariable

Q

SecondorderButterworthfiltercanberealizedbyusin

gvariableQfilterwithQ=0.707

1.9 SecondorderBesselfilterfromVariableQ

SecondorderBesselfiltercanberealizedbyusingvari

ableQfilterwithQ=0.5

1.10 FirstOrderButterworthFilters

FilterParameters

F_s = SamplerateinHz

F_c = CutfrequencyinHz

HL=LRFiltertype(high,low)

ErrorChecking

$$0 \leq F_c \leq \frac{F_s}{2}$$

$$0 \leq F_s \leq 192K$$

$$HL = (high, low)$$

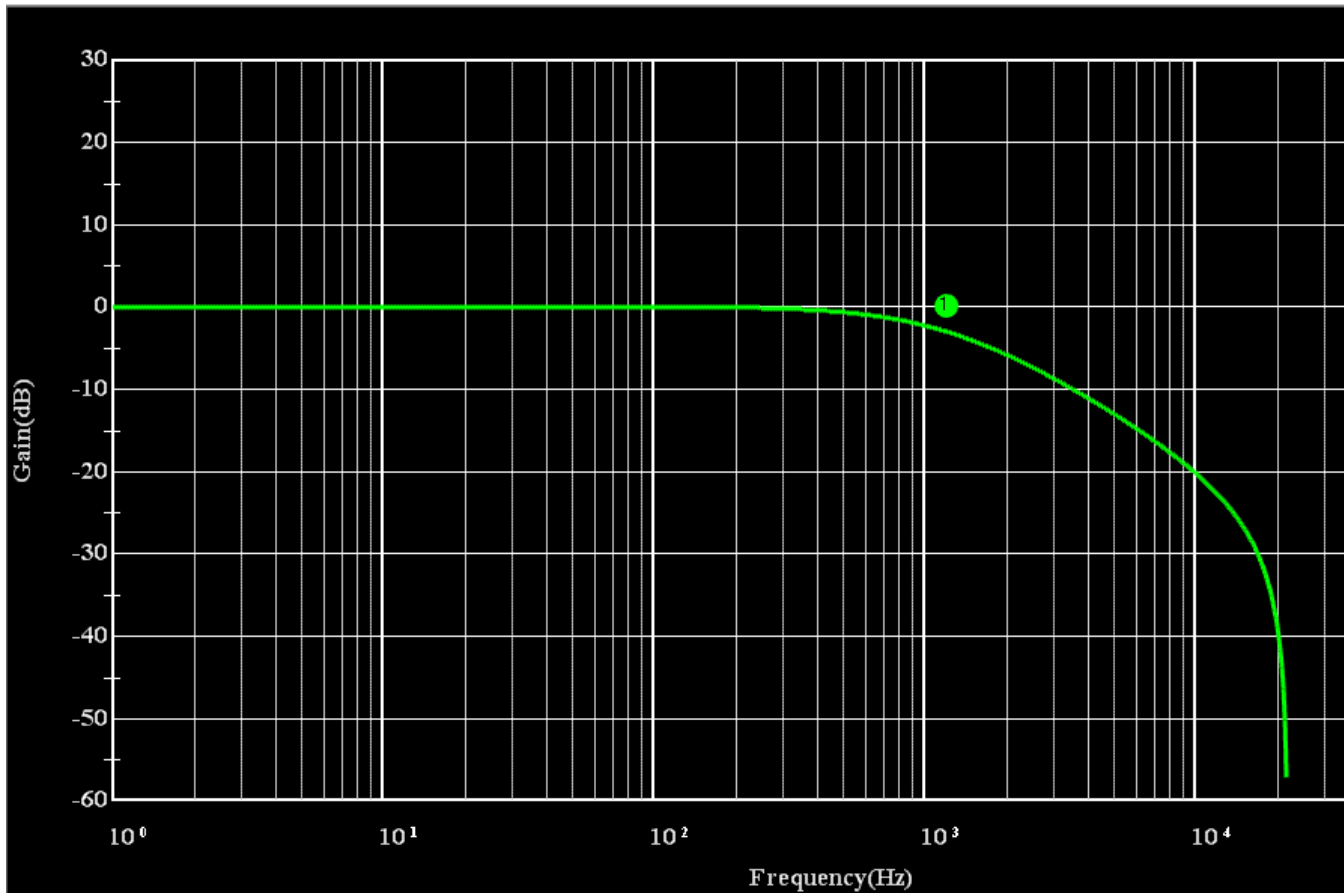


Figure8. FirstorderButterworth($F_c=1200\text{Hz}, HL=low$)

Equation

$$k = \frac{2 \times \pi \times F_c}{\tan\left(\frac{\pi \times F_c}{F_s}\right)}$$

$$Wc = 2 \times \pi \times F_c$$

If $HL(1:3) = 'low'$

$$b0 = \frac{Wc}{k + Wc}$$

$$b1 = \frac{Wc}{k + Wc}$$

else

$$b0 = \frac{k}{k + Wc}$$

$$b1 = -\frac{k}{k + Wc}$$

end

$$a1 = \frac{Wc - k}{k + Wc}$$

FilterCoefficients

$$B = [b0 \ b1 \ 0]$$

$$A = [1 \ a1 \ 0]$$

1.11 SecondorderChebychev

FilterParameters

F_s =SamplerateinHz

rip=RipplespecificationindB

typ=Filtertype(high,low,stop)

```

if typ(1:3) == 'sto'
    Fc=Stop band Input Lower and upper frequencies [f1,f2]
Else
    Fc=Cutoff frequency in Hz
    If Scale peak to 0dB
        Nrm=1
    If Scale PB to 0dB
        Nrm = -1
    if nrm ≅ 1
        rip = rip × -1
    end
End

```

ErrorChecking

$$0 \leq F_c \leq \frac{F_s}{2}$$

$$0 \leq F_s \leq 192K$$

$$0 \leq rip \leq 10$$

$$Nrm = 1, -1$$

$$HL = (high, low, stop)$$

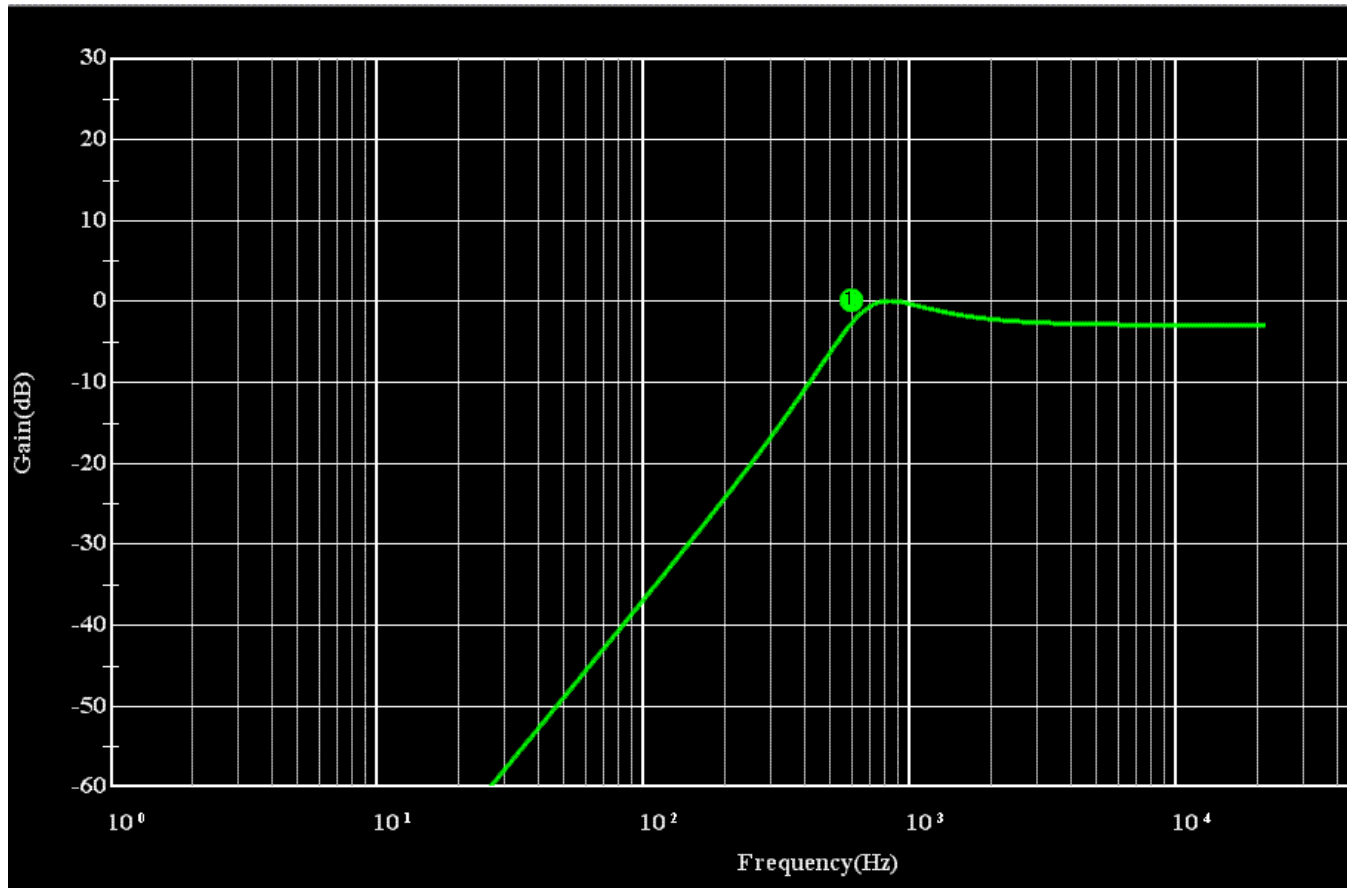


Figure9. SecondorderChebychev($F_c=600\text{Hz}, \text{typ}=\text{high}, \text{rip}=3\text{dB}$)

Equation

```

If typ(1:3)=='sto'
    Call cheby1(ord,rip,  $\frac{2 \times F_c}{F_s}$ ,HL)
else
    Call soCHBI( $F_c, F_s, \text{rip}, \text{HL}$ )
End

```

```

Function soCHBI( $F_c, F_s, \text{rip}, \text{HL}$ )

```

```

    If sign(rip)>0
        Sf=1
    Else
        Sf=0
    End

```

```

R = |rip|
If R == 0
    B = [1 0 0]
    A = [1 0 0]
else
    wc = 2 * pi * Fc
    ε = sqrt(1010 * R - 1)
    α = (a * sinh(1/ε)) / 2
    β1 = 3 * π / 4
    β2 = 5 * π / 4

    s1 = sinh(α) * cos(β1) + cosh(α) * sin(β1) * i
    s2 = sinh(α) * cos(β2) + cosh(α) * sin(β2) * i
    a = real(s1 + s2)
    b = real(s1 * s2)
    c = b
    if sf
        c = c / sqrt(1 + ε2)
    end
    if HL(1:3) == 'low'
        Ba = [0 0 c * wc2]
        Aa = [1 wc * a b * wc2]
    Else
        Ba = [c/b 0 0]
        Aa = [1 wc * a/b wc2/b]
    End
End

```

$$k = 2 \times \pi \times \frac{F_c}{\tan\left(\pi \times \frac{F_c}{F_s}\right)}$$

$$B = [Ba(1) \times k^2 + Ba(3) + Ba(2) \times k, -2 \times Ba(1) \times k^2 + 2 \times Ba(3), -Ba(2) \times k + Ba(1) \times k^2 + Ba(3)]$$

$$A = [Aa(1) \times k^2 + Aa(3) + Aa(2) \times k, -2 \times Aa(1) \times k^2 + 2 \times Aa(3), -Aa(2) \times k + Aa(1) \times k^2 + Aa(3)]$$

$$B = \frac{B}{A(1)}$$

$$A = \frac{A}{A(1)}$$

End

2 Number representation format for filter coefficients

AIC codecs come in two flavors. The “standard” version of the MiniDSP uses a 16-bit coefficient and a 24-bit data word for miniDSP_A and 28-bit data word for miniDSP_D while the “enhanced” version of the MiniDSP uses a 24-bit coefficient and a 32-bit data word. The AIC3254 and AIC3204 are enhanced devices. The TSC2117, AIC36, AC3110, AIC3111, and AIC3120 are standard devices.

All of the AIC codec devices use a 3.x data format (3.21 for the standard devices and 3.29 for the enhanced devices). This permits only two magnitude bits of headroom for signals that are greater than 1. To reduce the chance of clipping these signals in the AIC devices, the overall gain of the filter is moderated by scaling the numerator value based upon the value of the b_0 term.

The coefficient size in the AIC codec family is 16-bit 1.15 format for the standard devices and a 24-bit 1.23 format for the enhanced AIC devices. With these formats the MiniDSP coefficients are able to represent a maximum positive gain of $1-2^{-15}$ for a 16-bit coefficient and $1-2^{-23}$ for a 24-bit coefficient. When filter coefficients are computed for an AIC codec, the gain of the filter response is scaled to permit the values to be represented in a 1.23 or a 1.15 format.

Once we have computed the filter equations from above we then must perform a couple of steps prior to loading the coefficients into the codec.

In the AIC codecs use a specific biquad implementation to accommodate the 1.15 and 1.23 coefficient data format.

$$y(n) = b_0 \times x(n) + 2 \times b_1 \times x(n-1) + b_2 \times x(n-2) + 2a_1 \times y(n-1) + a_2 \times y(n-2)$$

1. For format we first must scale the b terms by the b₀ value. The b₀, b₁, and b₂ terms are multiplied by a scaling value to limit the overall gain of the filter.
 - If b₀ is greater than 1, then the default value of the scaling value is 1/b₀, otherwise it is 1.
 - The scaling factor is then applied to the b₀, b₁, and b₂ terms of the filter.
 - This default value is computed and displayed in a user-modifiable field when the filter coefficients are recomputed.
 - The user is permitted to change this value to a smaller value. However, if they attempt to set it to a larger value than the default scaling value, then the value will snap to the default scaling value.
2. Then both the numerator and denominator coefficients are scaled by a constant value.
 - If the coefficients are being computed for a “standard” device then the b₀, b₁, b₂, a₁, and a₂ terms are multiplied by 2¹⁵ and rounded to integer.
 - If the coefficients are being computed for an “enhanced” device then the b₀, b₁, b₂, a₁, and a₂ terms are multiplied by 2²³ and rounded to integer.
3. Then we scale the b₁ and a₁ terms by 0.5.

Appendix A illustrates the generation of coefficients using the above procedure.

2.1 Filter coefficient normalization

Filter coefficient normalization is performed to limit the size of the coefficient values, as described above, and to limit the maximum gain of the filter to avoid clipping.

There are two places where clipping can occur.

1. Internal signal levels and clipping

The miniDSP is able to internally represent a data value using a 3.29 or 3.21 format. This permits signal levels as large as 12 dB to be represented. However, the intermediate gains inside of a filter can be higher than the signal level that are visible at the output of a component. To account for this, filter gains are typically scaled to reserve part or all of this 12 dB as headroom for internal computations.

2. Output signal levels and clipping

The maximum signal level that can be output without clipping by the I2S output or the DAC interpolator is a data value represented as a by a 1.29 or 1.21 value (0.99 and -1.0). This corresponds to a signal level of 0 dB.

Coefficient scaling can be used to avoid clipping during internal operations and when the signal is sent to the I²S or interpolator outputs.

For example, we wish to use an EQ filter that has a gain of 15 dB at 500 Hz. This filter response is shown as the blue curve in Figure 10. To avoid over-normalizing the filter coefficients so that the signal at the output does not exceed 5 dB. In this example, the filter gain is scaled by 10 dB (multiplying the coefficients by 0.316227766). The scaled filter response is shown by the red curve in Figure 10.

Similarly to avoid clipping when the signal was output not exceed 0 dB. In this example, the filter gain is scaled by 15 dB (multiplying the coefficients by 0.177827941).. This is shown by the green curve in Figure 10.

Similarly to avoid clipping when the signal was output not exceed 0 dB. In this example, the filter gain is scaled by 15 dB (multiplying the coefficients by 0.177827941).. This is shown by the green curve in Figure 10.

, the filter gain should be scaled so that it does not exceed 0 dB. In this example, the filter gain is scaled by 15 dB (multiplying the coefficients by 0.177827941).. This is shown by the green curve in Figure 10.

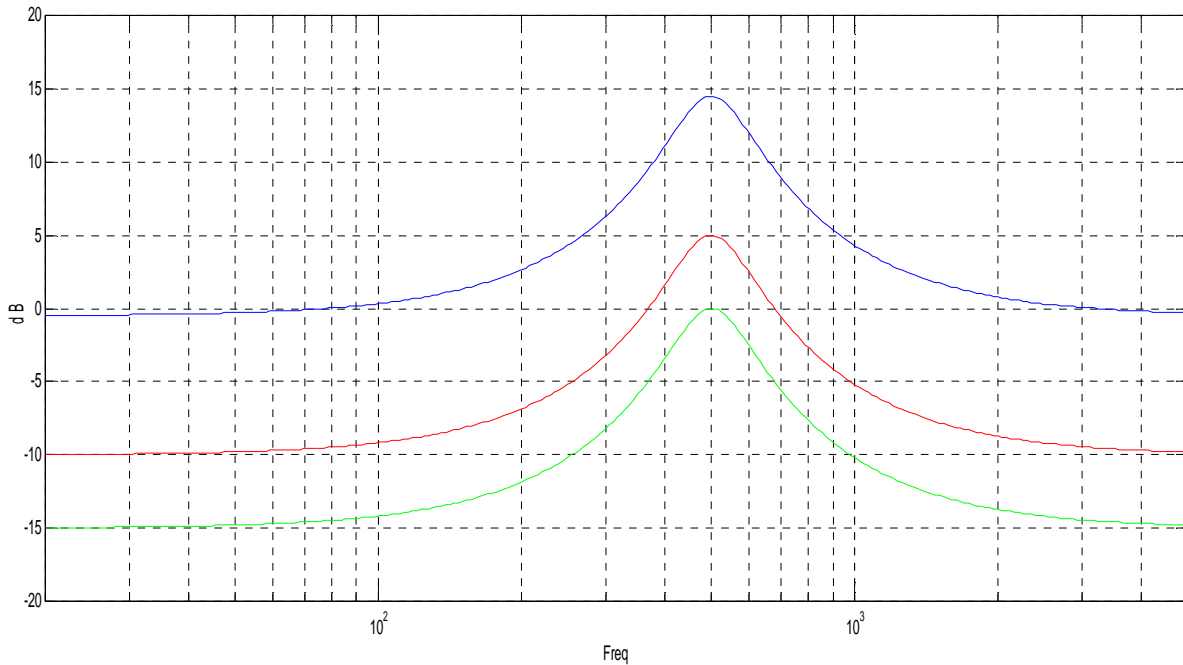


Figure 10. Normalizing filter response

3 Updating the codec filter coefficients

A host controller needs to provide the filter coefficient values over I2C to the AIC codecs. On the controller, the coefficient values need to be precomputed and stored in a table. The host controller will read the values from the table and download the coefficient values in a sequence that starts at the current bi-quad filter setting to the desired new value when an update is requested. The change of gains must be in increments/decrements of 1/4 dB to avoid pops and clicks. For instance, if an EQ filter at 200 Hz needs to be operated between ranges of -8 dB to 8 dB, the host controller needs to maintain a table of b0, b1, b2, a1 and a2 terms through -8 dB, -7.75 dB, -7.5 dB... 7 dB, 7.25 dB, 7.5 dB, 7.75 dB, 8 dB.

ent values over I2C to the AIC codecs. On the controller, the coefficient values need to be precomputed and stored in a table. The host controller will read the values from the table and download the coefficient values in a sequence that starts at the current bi-quad filter setting to the desired new value when an update is requested. The change of gains must be in increments/decrements of 1/4 dB to avoid pops and clicks. For instance, if an EQ filter at 200 Hz needs to be operated between ranges of -8 dB to 8 dB, the host controller needs to maintain a table of b0, b1, b2, a1 and a2 terms through -8 dB, -7.75 dB, -7.5 dB... 7 dB, 7.25 dB, 7.5 dB, 7.75 dB, 8 dB.

Table 1. EQ Filter coefficient table in host controller for “enhanced” codecs

	b0	b1	b0	a1	a2
-8dB	0x7F5119	0x812FC5	0x7E69BC	0x7ED03B	0x82452A
-7.75dB					
-7.5dB					
-7.25dB					
...					
0dB	0x7FFFFFFF	0	0	0	0
...					
7.5dB					
7.75dB					
8dB	0x7F6664	0x81C7F3	0x7D23F5	0x7F7E70	0x80E89C

Appendix A. Biquad coefficients computation example

This appendix illustrates the computation of biquad coefficients and converting them into a format required to load in AIC codecs.

1. Filter Specification: EQ filter with $f_c=5000\text{Hz}$, Gain=6dB and $Q=2.87$ (BW=1742Hz) on AIC3254.

Applying the equations in section 1.2 we get the biquad coefficients as

$$b_0 = 1.03381744095486$$

$$b_1 = -1.85413395878212$$

$$b_2 = 0.898225719031722$$

$$a_1 = 1.85413395878212$$

$$a_2 = -0.932043159986584$$

Before writing these coefficients to codec memory, couple of normalization steps must be performed based on their values according to section 2.

2. Since $b_0 > 1$, we need to scale the numerator coefficients by scale factor $\frac{1}{b_0}$

$$b_0 = 0.992464542388916015625$$

$$b_1 = -1.77996826171875$$

$$b_2 = 0.862296581268310546875$$

3. Scale the b_1 and a_1 by 0.5.

$$b_0 = 0.992464542388916015625$$

$$b_1 = -0.889984130859375$$

$$b_2 = 0.862296581268310546875$$

$$a_1 = 0.92706697939106$$

$$a_2 = -0.932043159986584$$

4. Since the coefficients are to be computed for 'enhance' mode of device, we need to scale it by 2^{23} and rounded to nearest integer. Hence the final coefficients onto the AIC3254 are

$$b_0 = 0x7F0914$$

$$b_1 = 0x8E1500$$

$$b_2 = 0x6E5FBC$$

$$a_1 = 0x76AA20$$

$$a_2 = 0x88B2D1$$