SigmaStudio Filter Coefficient Calculations

This document shows how to calculate B0, B1, B2, A1, and A2 for various IIR filter types in SigmaStudio.
Sin = sine, **Cos** = cosine, **Tan** = tangent, **Sinh** = hyperbolic sine, **Log** = base 10 logarithm

Parametric EQ Calculation

Values from Control:

```
boost (boost of filter)
frequency (center frequency)
Q (Q of filter; must be greater than or equal to 0.01)
gain (linear gain applied to the signal)
Fs (sample rate of project)

      In the case that boost = 0,

        gainlinear = 10^(gain / 20)
        B0 = gainlinear
        B1 = 0
        B2 = 0
        A1 = 0
        A2 = 0


      If boost is not 0,

        Create 5 new variables: a0, omega, sn, cs, alpha, Ax

        Ax = 10^(boost / 40)
        omega = 2 * PI * frequency / Fs
        sn = Sin(omega)
        cs = Cos(omega)
        alpha = sn / (2 * (Q))

        a0 = 1 + (alpha / Ax)
        A1 = -( 2 * cs) / a0
        A2 = (1 - (alpha / Ax)) / a0
        gainlinear = 10^(gain / 20) / a0

        B0 = (1 + (alpha * Ax)) * gainlinear
        B1 = -( 2 * cs) * gainlinear
        B2 = (1 - (alpha * Ax)) * gainlinear
```

Tone Control

Values from Control:

Freq_T (treble cutoff frequency)
Boost_T (treble boost)
Freq_B (bass cutoff frequency)
Boost_B (bass boost)
Fs (sample rate of project)

$$Boost\_T = 10^{(Boost\_T / 20)}$$
$$Boost\_B = 10^{(Boost\_B / 20)}$$

```
A_T = Tan(PI * Freq_T / Fs)
A_B = Tan(PI * Freq_B / Fs)
Knum_T = 2 / (1 + (1 / Boost_T))
Kden_T = 2 / (1 + Boost_T)
Knum_B = 2 / (1 + (1 / Boost_B))
Kden_B = 2 / (1 + Boost_B)

a10 = A_T + Kden_T                b10 = A_T + Knum_T
a11 = A_T - Kden_T                b11 = A_T - Knum_T

a20 = (A_B * Kden_B) + 1          b20 = (A_B * Knum_B) - 1
a21 = (A_B * Kden_B) - 1          b21 = (A_B * Knum_B) + 1

a0 = a10 * a20
A1 = ((a10 * a21) + (a11 * a20)) / a0
A2 = a11 * a21                            / a0
```

$$gainlinear = 10^{(cell\_gain / 20)}$$

```
B0 = (b10 * b20) / a0 * gainlinear
B1 = ((b10 * b21) + (b11 * b20)) / a0 * gainlinear
B2 = (b11 * b21) / a0 * gainlinear
```

For double 1st Order Filter cells, simply use 2 Cascaded 1st Order Filters

All Pass

Values from control:

frequency (center frequency)
Q (Q of filter)
gain1 (linear gain)
fs (sample rate of project)

```
gain1 = 10^(gain / 20)
omega = 2 * PI * frequency / fs
sins = Sin(omega)
coss = Cos(omega)
alpha = sins / (2 * Q)

norm = 1 + alpha

B0 = gain1 * (1 - alpha) / norm
B1 = gain1 * ( - 2 * coss) / norm
B2 = gain1 * (1 + alpha) / norm
A1 = - 2 * coss / norm
A2 = (1 - alpha) / norm
```

Notch Filter

Values from control:

frequency1 (center frequency)
Q (Q of filter)
g (gain)
fs (sample rate of project)

```
omega = frequency1 * 2 * System.PI / fs
deltaomega = omega / Q
b = 1 / (1 + Tan(deltaomega / 2))
gain = 10^(g / 20)
B0 = gain * b
B1 = gain * ( - 2 * b * Cos(omega))
B2 = gain * b
A1 = - 2 * b * Cos(omega)
A2 = (2 * b - 1)
```

Chebyshev Low Pass

Values from control:

frequency (cutoff frequency; must be 1 Hz or greater)
gain (linear gain)
ripple (ripple of filter; must be 0.1 or greater)
Fs (sample rate of project)

wp = (2 * PI * frequency) / Fs
Omegap = Tan(wp / 2)
epass = (10$^{(0.1 * ripple)}$ - 1)$^{0.5}$
alpha = (0.5) * Log(1 / epass + (1 / (epass^{2} + 1))$^{0.5}$
Omega0 = Omegap * Sinh(alpha);
theta = (PI / 4) * 3
Omega1 = Omegap * Sin(theta)
H0 = (1 / (1 + epass^{2}))$^{0.5}$
Den = 1 – (2 * Omega0 * Cos(theta)) + Omega0^{2} + Omega1^{2}
G = (Omega0^{2} + Omega1^{2}) / Den

A1 = (2 * (Omega0^{2} + Omega1^{2} - 1)) / Den
A2 = (1 + (2 * Omega0 * Cos(theta)) + Omega0^{2} + Omega1^{2}) / Den
B0 = H0 * G * 10$^{(gain / 20)}$
B1 = B0 * 2
B2 = B0

ChebyshevHighPass

Values from control:

frequency (cutoff frequency; must be 1 Hz or greater)
gain (linear gain)
ripple (ripple of filter; must be 0.1 or greater)
Fs (sample rate of project)

wp = (2 * PI * frequency) / Fs
Omegap = 1 / Tan(wp / 2)
epass = $(10^{(0.1 * ripple)} - 1)^{0.5}$
alpha = (0.5) * Log(1 / epass + $(1 / (epass^2) + 1))^{0.5}$
Omega0 = Omegap * Sinh(alpha)
theta = (PI / 4) * 3
Omega1 = Omegap * Sin(theta)
H0 = $(1 / (1 + (epass^2)))^{0.5}$
Den = 1 - 2 * Omega0 * Cos(theta) + ($Omega0^2$) + ($Omega1^2$)
G = (($Omega0^2$) + ($Omega1^2$)) / Den

A1 = (- 2 * (($Omega0^2$) + ($Omega1^2$) - 1)) / Den
A2 = (1 + 2 * Omega0 * Cos(theta) + ($Omega0^2$) + ($Omega1^2$)) / Den
B0 = H0 * G * ($10^{(gain / 20)}$)
B1 = - B0 * 2
B2 = B0

Linkwitz-Riley – 12 dB/oct = 2 cascaded 1st order butterworths (2 biquads)

frequency (the cutoff frequency)
First Order **type** (the type of filter, can be lowpass, highpass, or allpass)
g (the linear gain)
fs (the sample rate of the project)

1st Order Butterworth

gain = $(10^{(g / 20)})$

```
omega, sn, cs, alpha, a0
omega = 2 * PI * frequency / fs
sn = Sin(omega)
cs = Cos(omega)

a0 = sn + cs + 1;
```

For a First Order **lowpass…**

```
A1 = (sn - cs - 1) / a0
B0 = gain * sn / a0
B1 = gain * sn / a0
```

For a First Order **highpass…**

```
A1 = (sn - cs - 1) / a0
B0 = gain * (1 + cs) / a0
B1 = - gain * (1 + cs) / a0
```

For a First Order **allpass…**

A1 = $(2.7^{(- 2 * PI * frequency / fs)})$

```
B0 = - gain * A1
B1 = gain
```

Linkwitz-Riley – 24 dB/oct = 2 cascaded 2nd order butterworths (2 biquads)

frequency (the cutoff frequency)
gain (the linear gain)
Fs (the sample rate of the project)

2^{nd} Order LOWPASS

```
omega, sn, cs, alpha, a0;
omega = 2 * PI * frequency / Fs
sn  = Sin(omega)
cs  = Cos(omega)

alpha = sn / (2 * (1 / (2)^0.5))
a0 = 1 + alpha
A1 = -( 2 * cs) / a0
A2 = (1 - alpha) / a0
B1 = (1 - cs) / a0 * (10^(gain / 20))

B0 = B1 / 2
B2 = B0
```

2^{nd} Order HIGHPASS

```
omega = 2 * PI * frequency / Fs
sn  = Sin(omega)
cs  = Cos(omega)

alpha = sn / (2 * (1 / (2)^0.5))

a0 = 1 + alpha
A1 = -( 2 * cs) / a0
A2 = (1 - alpha) / a0
B1 = -( 1 + cs) / a0 * (10^(gain / 20))

B0 = - B1 / 2
B2 = B0
```

Linkwitz-Riley – 36 dB/oct = 2 cascaded 3rd order butterworths
3rd order butterworth is implemented by cascading a "HigherOrder" + 1st order
1st Filter: [HigherOrder] orderindex = 3, i = 0
2nd Filter: 1st Order Butterworth
3rd Filter: [HigherOrder] orderindex = 3, i = 0
4th Filter: 1st order Butterworth

frequency (the cutoff frequency)
gain (the linear gain)
Fs (the sample rate of the project)
orderindex (described above… changes based on type)
i (described above)

LOW PASS HIGHER ORDER

```
omega, sn, cs, alpha, a0 , orderangle
omega = 2 * PI * frequency / Fs
sn = Sin(omega)
cs = Cos(omega)

    orderangle = (PI / orderindex) * (i + 0.5)
alpha = sn / (2 * (1 / (2 * Sin(orderangle))))
a0 = 1 + alpha
A1 = -( 2 * cs) / a0
A2 = (1 - alpha) / a0
B1 = (1 - cs) / a0 * (10^(gain / 20))

B0 = B1 / 2
B2 = B0
```

HIGH PASS HIGHER ORDER

```
       omega = 2 * PI * frequency / Fs
       sn  = Sin(omega)
       cs  = Cos(omega)

       orderangle = (PI / orderindex) * (i + 0.5)
alpha = sn / (2 * (1 / (2 * Sin(orderangle))))
       a0 = 1 + alpha
       A1 = -( 2 * cs) / a0
       A2 = (1 - alpha) / a0
       B1 = -( 1 + cs) / a0 * (10^(gain / 20))

       B0 = - B1 / 2
       B2 = B0
```

Linkwitz-Riley – 48 dB/oct = 2 cascaded 4th order butterworths
4th order butterworth is implemented by cascading 2 2nd "Higher Order" using equations shown above.
1st Filter: orderindex = 4, i = 0
2nd Filter: orderindex = 4, i = 1
3rd Filter: orderindex = 4, i = 0
4th Filter: orderindex = 4, i = 1

Butterworth 12 dB/oct = One 2nd order butterworth

Butterworth 18 dB/oct = One Higher order butterworth + One 1st Order

Filt 1: [HigherOrder]: orderindex = 3, i = 0

Filt 2: 1st Order butterworth

Butterworth 24 dB/oct = 2 Higher order butterworths

Filt 1: orderindex = 4, i = 0

Filt 2: orderindex = 4, i = 1

Bessel 12 dB/oct = One 2nd order Bessel

```
frequency,
gain,
Fs
```

Low Pass Bessel

```
omega, sn, cs, alpha, a0
omega = 2 * PI * frequency / Fs
sn  = Sin(omega)
cs  = Cos(omega)

alpha = sn / (2 * (1 / (3)^0.5))

a0 = 1 + alpha
A1 = -( 2 * cs) / a0
A2 = (1 - alpha) / a0
B1 = (1 - cs) / a0 * (10^(gain / 20))

B0 = B1 / 2
B2 = B0
```

High Pass Bessel

```
omega = 2 * PI * frequency / Fs
sn  = Sin(omega)
cs  = Cos(omega)

alpha = sn / (2 * (1 / (3)^0.5))

a0 = 1 + alpha
A1 = -( 2 * cs) / a0
A2 = (1 - alpha) / a0
B1 = -( 1 + cs) / a0 * (10^(gain / 20))

B0 = - B1 / 2
B2 = B0
```

Bessel 18 dB/oct = one 2nd order Bessel + One 1st order Butterworth

Bessel 24 dB/oct = Two 2nd order Besels