Parameter Quantization in Direct-Form Recursive Audio Filters

Brian Neunaber
QSC Audio Products
1675 MacArthur Blvd.
Costa Mesa, CA 92626

Abstract – The effect of coefficient quantization on audio filter parameters using the direct-form filter implementation is analyzed. An expression for estimating the maximum error in frequency and Q resolution is developed. Due to coefficient quantization, appreciable error in the DC gain of some types of second-order direct-form filters may result. Simple techniques are developed for reducing or eliminating this error without increasing filter complexity or coefficient precision.

0 Introduction

The direct-form I (DF1) filter topology is preferred for recursive audio filtering [1], [2], and its efficiency of implementation is hard to beat. However, one disadvantage of the DF1 topology is its poor coefficient sensitivity [3], [4]. Recent trends to increase sampling rates further degrade coefficient sensitivity. Using higher-precision coefficients often comes at an expense, such as increased hardware cost or reduced performance from double-precision arithmetic. We analyze how coefficient quantization affects filter parameters and introduce the concept of *parameter quantization*. We develop methods for minimizing these effects without increasing filter complexity or coefficient precision.

1 Background

For high quality audio using a fixed-point DF1 implementation, a minimum of 24-bit signal precision with 48-bit accumulator precision and some form of error feedback is recommended. With truncation error cancellation, the DF1 has noise performance sufficient for the most demanding audio applications. The DF1 filter topology with truncation error cancellation is mathematically equivalent to double-precision in the signal feedback paths using single-precision coefficients. As a result, truncation error cancellation greatly improves signal-to-quantization noise of the DF1 but does nothing for coefficient sensitivity. For more information on error feedback and truncation error cancellation, the reader is referred to [1] and [2].

High order recursive filters may be broken down into parallel or cascade first- and second-order sections, and there are good reasons to do so. Cascade implementation of first- and second-order sections has better coefficient sensitivity than direct implementation and is easier to analyze [4]. Many audio equalization filters are implemented as parametric first- or second-order sections, such as shelving or boost/cut (also called *peak* or *presence*) filters; and graphic equalizers are implemented as either parallel or cascaded second-order sections. Therefore, we limit our analysis of coefficient quantization to first- and second-order sections only. Higher order filters may be constructed from these basic structures.

1.1 Recursive Filter Transfer Function

Given the parameters of gain, frequency, and Q (in the second-order case), we first develop the coefficients for several types of audio filters.

1.1.1 First-order Case

The general bilinear transfer function, $H_1(s)$, of a first-order filter is written as

$$H_1(s) = \frac{V_H s + V_L \omega}{s + \omega} \tag{1}$$

where V_H is the high-pass gain (at the Nyquist frequency), V_L is the low-pass gain (at DC), and $\omega = 2\pi \cdot f_C$ [5]. f_C is the cutoff frequency (for high- and low-pass filters) or center frequency (for all-pass and shelf filters).

To convert Equation (1) to the digital domain, we use the bilinear transform. We make the following substitutions [3]:

$$s = \frac{z-1}{z+1}$$
$$\omega \to \Omega = \tan\left(\pi \frac{f_C}{f_s}\right) \tag{2}$$

The sampling rate is f_s. After the substitutions, we simplify the general bilinear transfer function to the following:

$$H_1(z) = \frac{(V_L\Omega + V_H) + (V_L\Omega - V_H)z^{-1}}{(\Omega+1) + (\Omega-1)z^{-1}} \tag{3}$$

Given the first-order transfer function, $H_1(z)$, in the form

$$H_1(z) = \frac{a_0 + a_1 z^{-1}}{1 + b_1 z^{-1}} \tag{4}$$

The coefficients of $H_1(z)$ become the following:

$$a_0 = \frac{V_L\Omega + V_H}{\Omega+1} \tag{5}$$

$$a_1 = \frac{V_L\Omega - V_H}{\Omega+1} \tag{6}$$

$$b_1 = \frac{\Omega-1}{\Omega+1} \tag{7}$$

The parameters for common first-order audio filter types are shown in Table 1. The functions *min(x, y)* and *max(x, y)* return the minimum and maximum (respectively) of their arguments.

	High-pass	**Low-pass**	**All-pass**	**High Shelf**	**Low Shelf**
Ω	Ω	Ω	Ω	$\Omega \cdot \min(V_H, 1)$	$\Omega \cdot \max\left(\frac{1}{V_L}, 1\right)$
$\mathbf{V_L}$	0	1	1	1	V_L
$\mathbf{V_H}$	1	0	-1	V_H	1

Table 1. First-order parameters for common audio filters.

1.1.2 Second-order Case

The general biquadratic transfer function, $H_2(s)$, of a second-order filter is written as

$$H_2(s) = \frac{V_H s^2 + V_B \frac{\omega}{Q} s + V_L \omega^2}{s^2 + \frac{\omega}{Q} s + \omega^2} \tag{8}$$

where V_H is the high-pass gain, V_B is the band-pass gain (at f_C), V_L is the low-pass gain, and $\omega=2\pi \cdot f_C$ [5]. f_C is the cutoff frequency (for high- and low-pass filters) or center frequency (for all-pass, shelf and boost/cut filters). To convert Equation (8) to the digital domain, we again use the bilinear transform. After the substitutions, we get the general digital biquadratic transfer function

$$H_2(z) = \frac{\left(V_L\Omega^2 + V_B\frac{\Omega}{Q} + V_H\right) + 2\left(V_L\Omega^2 - V_H\right)z^{-1} + \left(V_L\Omega^2 - V_B\frac{\Omega}{Q} + V_H\right)z^{-2}}{\left(\Omega^2 + \frac{\Omega}{Q} + 1\right) + 2\left(\Omega^2 - 1\right)z^{-1} + \left(\Omega^2 - \frac{\Omega}{Q} + 1\right)z^{-2}} \tag{9}$$

We want $H_2(z)$ in the form

$$H_2(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{10}$$

So, the coefficients become

$$a_0 = \frac{V_L\Omega^2 + V_B\frac{\Omega}{Q} + V_H}{\Omega^2 + \frac{\Omega}{Q} + 1} \tag{11}$$

$$a_1 = \frac{2\left(V_L\Omega^2 - V_H\right)}{\Omega^2 + \frac{\Omega}{Q} + 1} \tag{12}$$

$$a_2 = \frac{V_L\Omega^2 - V_B\frac{\Omega}{Q} + V_H}{\Omega^2 + \frac{\Omega}{Q} + 1} \tag{13}$$

$$b_1 = \frac{2\left(\Omega^2 - 1\right)}{\Omega^2 + \frac{\Omega}{Q} + 1} \tag{14}$$

$$b_2 = \frac{\Omega^2 - \frac{\Omega}{Q} + 1}{\Omega^2 + \frac{\Omega}{Q} + 1} \tag{15}$$

The parameters for common second-order audio filter types are shown in Table 2. The boost/cut filter (based on [6]) is designed such that its frequency response is symmetrical about unity gain for complementary boost and cut gains.

	High-pass	Low-pass	All-pass	High Shelf	Low Shelf	Boost/Cut
Ω	Ω	Ω	Ω	$\Omega \cdot \min\left(\sqrt{V_H}, 1\right)$	$\Omega \cdot \max\left(\frac{1}{\sqrt{V_L}}, 1\right)$	Ω
Q	Q	Q	Q	Q	Q	$Q \cdot \min\left(V_B, 1\right)$
$\mathbf{V_L}$	0	1	1	1	V_L	1
$\mathbf{V_B}$	0	0	-1	$\sqrt{V_H}$	$\sqrt{V_L}$	V_B
$\mathbf{V_H}$	1	0	1	V_H	1	1

Table 2. Second-order parameters for common audio filters.

1.2 Implementation Considerations

We now discuss some considerations for both fixed- and floating-point implementation. As we will show, the topic of coefficient quantization becomes moot when using extended-precision floating-point arithmetic. However, to be comprehensive, floating-point quantization is discussed briefly.

1.2.1 Fixed-point Implementation

With fractional fixed-point implementation, care must be taken to insure that the magnitudes of the filter coefficients are bounded by 1.0. If the gain of the filter does not exceed 1.0 at any frequency, it can be shown that the magnitudes of a_0, a_2, and b_2 are always bounded by 1.0, while the magnitudes of a_1 and b_1 are bounded by 2.0. One way to remedy this is to halve a_1 and b_1, and accumulate their respective terms twice within the filter. This is allowable provided that the filter is known to be stable and the magnitude of its output always bounded by 1.0, regardless of any intermediate overflow that may occur. Jackson's Rule shows this to be true even within accumulator architectures without overflow bits [1].

If the magnitude of the filter's gain exceeds 1.0, the implementer must determine if this causes the magnitude of a_0 or a_2 to exceed 1.0 or a_1 to exceed 2.0. If this is the case, the implementer may scale the feed-forward (a_n) coefficients by the reciprocal of the filter's maximum gain and apply complementary scaling at the output of the filter to restore the filter gain. If the maximum gain is chosen as a power-of-2, the complementary scaling at the filter's output is simplified to a shift operation. Scaling the feed-forward coefficients is preferable to scaling the input signal itself, since the input signal's precision is maintained in the guard bits of the accumulator. Unfortunately, this increases the effects of coefficient sensitivity; as a result, there is a trade-off between the maximum gain (and headroom[1]) of the filter and its feed-forward coefficient sensitivity.

To simplify our analysis, we assume that scaling is not required. We choose examples that conform to this assumption, unless otherwise noted.

1.2.2 Floating-point Implementation

Floating-point implementation circumvents the scaling problem altogether, since the numerical representation is typically normalized. However, floating-point arithmetic does not circumvent the problem of coefficient sensitivity in the DF1 topology. Quantization of the mantissa may still result in response error at low f_C, and these effects must be considered.[2]

Thirty-two bit (single-precision) floating-point numbers have a 24-bit mantissa – at worst case, only one bit more precision than 24-bit fixed-point, due to the implied leading 1. This lack of sufficient guard bits makes 32-bit floating-point unacceptable for high quality audio when using the direct-form filter topology.

That said, many floating-point DSPs and general-purpose microprocessors have native support for double- or extended-precision floating-point arithmetic. The Analog Devices SHARC supports an extended-

[1] Increasing the headroom of the filter can reduce the filter's susceptibility to forced overflow oscillations [1].

[2] A simple modification to the DF1 topology patented by Rossum [10] dramatically improves floating-point coefficient sensitivity for audio filtering.

precision mode that uses 40-bit floating-point representation with a 32-bit mantissa [7]. The Texas Instruments TMS320C6x DSPs support 64-bit floating-point, although at higher latency than 32-bit [8]. The Intel P6 family processors natively support 64- and 80-bit floating-point arithmetic in the x87 FPU[3] [9]. We recommended using double or extended-precision when implementing high quality digital audio filters with the direct-form topology. Not only does this meet the requirement of low noise, but it also greatly reduces coefficient sensitivity.

2 Coefficient Quantization

Quantization of the filter coefficients induces an error upon the filter's response; this effect is referred to as *coefficient sensitivity*. Coefficient sensitivity is a function of the filter topology, and we only consider the DF1 topology here. While this error is negligible for most values of f_C, it can become significant for very low values of f_C. We show that the result of coefficient sensitivity is, for all practical purposes, a perturbation of f_C, Q, and the DC gain, V_L. While V_B and V_H are also affected, it is to a lesser extent and can be considered negligible for audio filtering.

2.1 Fixed-point Quantization Function

Assume the number x is to be quantized to a finite precision of b bits. There is one sign bit, S, and the remaining bits are used to represent the fractional part of the number, formatted as $S.(b-1)$. No bits are used to the left of the radix point to represent an integer part of the number; therefore, coefficients are constrained between –1.0 and $1.0-2^{-(b-1)}$. The quantization function, $q(x)$, becomes

$$\begin{aligned} \varepsilon &= 2^{-(b-1)}, \\ q(x) &= \varepsilon \cdot round\left(\frac{x}{\varepsilon}\right) \end{aligned} \tag{16}$$

where the *quantum*, ε, is the smallest numerical value representable with b bits.

2.2 Floating-point Quantization Function

Here, only the precision of the mantissa is considered. The number x is normalized by n to $1.(m-1)$ format, where the leading 1 of the mantissa is implied by the IEEE format so that its total resolution is m bits. The quantization function becomes

$$\begin{aligned} n &= \begin{cases} 1, \text{ if } x = 0 \\ 2^{\text{int}\left[\log_2(|x|)\right]}, \text{ otherwise} \end{cases} \\ q(x) &= 2^{-m} n \cdot round\left(\frac{2^m x}{n}\right) \end{aligned} \tag{17}$$

3 Parameter Quantization

The net result of coefficient sensitivity is *parameter quantization*: the quantization of a filter's coefficients has a perturbation effect on the filter's actual input parameters. We wish to determine how quantization affects a filter's parameters and begin our analysis with the first-order case.

[3] SSE2, Intel's second iteration of Streaming SIMD (single-instruction multiple-data) Extensions, was introduced with the Pentium 4 processor. SSE2 can operate on two 64-bit floating-point "quadwords" simultaneously, in addition to the x87 FPU.

3.1 First-Order Case

3.1.1 Reverse Calculation of Filter Parameters from Quantized Coefficients

Quantizing the first-order filter coefficients of Equations (5)-(7) and solving this system of three equations in three unknowns for f_C, V_L, and V_H, we get

$$f_C = \frac{f_S}{\pi}\tan^{-1}\left(\frac{1+q(b_1)}{1-q(b_1)}\right) \tag{18}$$

$$V_L = \frac{q(a_0)+q(a_1)}{1+q(b_1)} \tag{19}$$

$$V_H = \frac{q(a_0)-q(a_1)}{1-q(b_1)} \tag{20}$$

3.1.2 Perturbation of f_C

Examining the first-order case, from Equation (7) we observe that $b_1 \rightarrow -1.0$ as $\Omega \rightarrow 0$. However, due to quantization, b_1 can only approach –1.0 in increments of the quantum size, ε. Representing all possible values of b_1 (from –1.0 upward) as a function of ε,

$$b_1 = -1 + i\varepsilon,\ i = 0,1,2,... \tag{21}$$

Substituting into Equation (18),

$$f_C(i) = \frac{f_S}{\pi}\tan^{-1}\left(\frac{i\varepsilon}{2-i\varepsilon}\right) \tag{22}$$

Equation (22) yields the realizable frequencies for the first-order direct-form filter. If we set i=1, we see that the first-order filter can resolve a minimum f_C of 0.00091 Hz with 24-bit fixed-point coefficients, which is sufficient for even the most critical audio applications. We compute the relative error in $f_C(i)$ as

$$error\left(f_C(i)\right) = \left|\frac{f_C(i) - f_C(i-1)}{f_C(i)}\right| \tag{23}$$

Equation (23) is shown in Figure 1 as a function of $f_C(i)$.

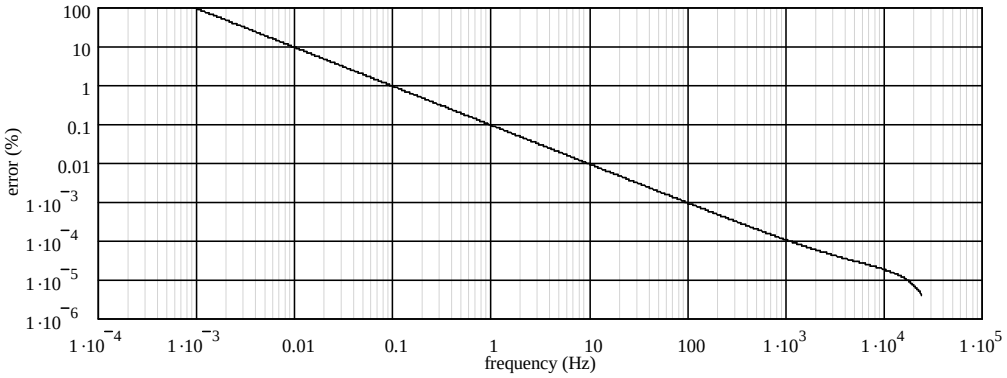

Figure 1. Frequency error of first-order recursive filter, 24-bit coefficients, f_S = 48 kHz.

3.1.3 Perturbation of V_L and V_H

Now, we consider each common audio filter type separately, since each type is affected differently by quantization. A summary of the quantization error in V_L and V_H is shown in Table 3.

Filter Type	Parameter	Max Error (%)	@ Frequency (Hz)	Notes
Low-Pass	V_L	0.005	20	a0 = a1 after quantization
	V_H	0		
High-Pass	V_L	0		a0 = -a1 after quantization
	V_H	3×10^{-5}	20k	
All-Pass	V_L	0		$a_0=b_1$, $a_1=1$ after quantization
	V_H	0		
High Shelf	V_L	0.01	20	V_H = 0.25
	V_H	3×10^{-5}	20k	
Low Shelf	V_L	0.01	20	V_L = 0.25
	V_H	1×10^{-3}	12k	

Table 3. Error in V_L and V_H of first-order recursive filter, 24-bit coefficients, f_S = 48 kHz.

Clearly, quantization effects on the first-order direct-form filter topology are negligible within the range of audio frequencies. The only application where these effects may need to be considered is in a smoothing filter, such as in the smoothing a control value. Even in this case, quantization effects only become significant when the time constant of the filter is greater than about $10^5/f_S$.

Although cascaded first-order filters may be implemented as one or more higher-order filters, we advise otherwise. We will show that the second-order frequency resolution is significantly poorer than that of the first-order filter. In addition, the first-order filter is low in noise [1], its transient response does not exhibit overshoot, and it is simple and efficient to implement. For example, a second-order Linkwitz-Riley crossover simply consists of cascaded first-order Butterworth filters; these filters should be implemented as cascaded first-order sections when practical to do so.

3.2 Second-Order Case

We are familiar with the direct-form pole distribution, shown in Figure 2. This distribution tells us that coefficient sensitivity is greater at low frequencies, but how the filter's parameters are affected is not exactly clear.

Floating-point coefficients are not much help at low frequencies. When comparing N-bit fixed-point to floating-point with an N-bit mantissa (Figure 3), there is only a factor-of-2 increase in pole density at low frequencies due to the implied leading 1 of the IEEE floating-point format. The pole density increases by a factor of 2 in vertical bands corresponding to each time b_1 is halved and in radial bands corresponding to each time b_2 is halved.

Here, "low" frequencies — the region where $-2 \le b_1 < -1$ and $0.5 \le b_2 < 1$ — is a significant portion of the audio band, and this region increases with Q. In the worst case (as $Q \to \infty$), this region is between 0 and $0.167 \cdot f_S$; in a practical best-case (Q = 0.5), this region is between 0 and $0.054 \cdot f_S$. [4]

[4] The maximum frequency values are found by maximizing Equation (24) within the regions specified for the given value of Q.

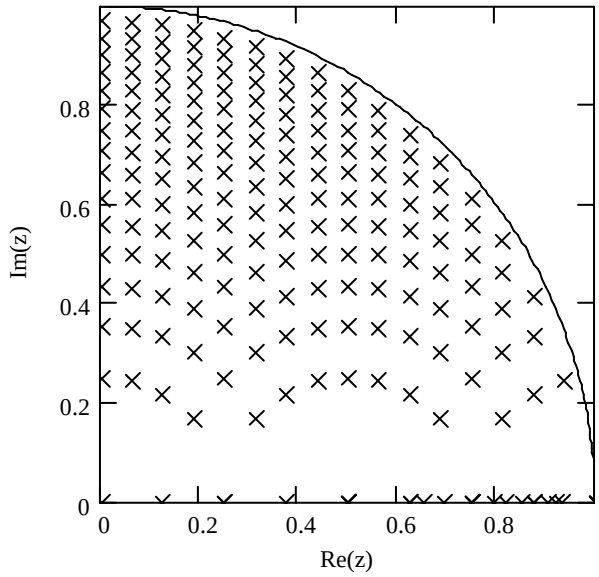

Figure 2. Direct form pole distribution with 5-bit (S.4) fixed-point coefficients.

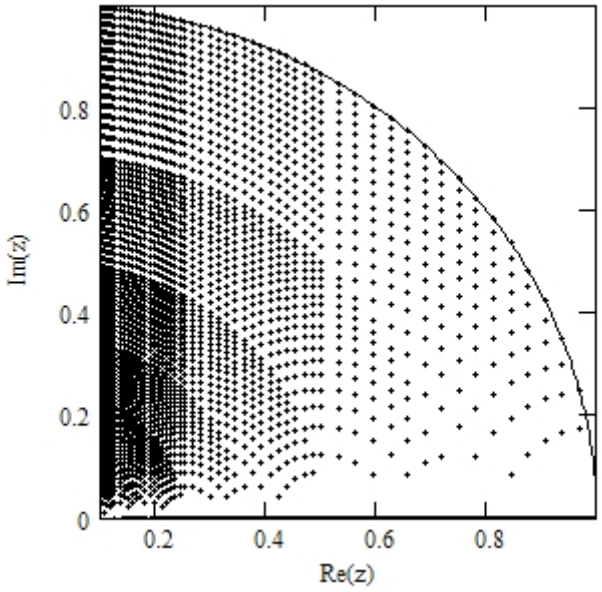

Figure 3. Direct form pole distribution using floating-point coefficients with a 5-bit mantissa. The real axis below 0.1 is not shown, since increasing pole density obscures the plot.

3.2.1 Reverse Calculation of Filter Parameters from Quantized Coefficients

For the second-order case, we use the coefficients of Equations (11)-(15). Solving this system of five equations in five unknowns for f_C, Q, V_L, V_B, and V_H, we get the following:

$$f_C = \frac{f_S}{\pi} \tan^{-1}\left(\sqrt{\frac{1+q(b_1)+q(b_2)}{1-q(b_1)+q(b_2)}} \right) \tag{24}$$

$$Q = \frac{\sqrt{\left(q(b_2)+1\right)^2 - q(b_1)^2}}{2\cdot\left|1-q(b_2)\right|} \tag{25}$$

$$V_L = \frac{q(a_0)+q(a_1)+q(a_2)}{1+q(b_1)+q(b_2)} \tag{26}$$

$$V_B = \frac{q(a_0)-q(a_2)}{1-q(b_2)} \tag{27}$$

$$V_H = \frac{q(a_0)-q(a_1)+q(a_2)}{1-q(b_1)+q(b_2)} \tag{28}$$

3.2.2 Perturbation of f_C

We know the pole quantization of the second-order DF1 filter is poorest at low frequencies, but we wish to know more precisely how f_C is affected by this quantization. Using the same technique we used in the first-order case, we see from Equations (14) and (15) that $b_1 \rightarrow -2.0$ and $b_2 \rightarrow 1.0$ as $\Omega \rightarrow 0$. Representing b_1 and b_2 as a function of ε,

$$\begin{aligned} b_1 &= -2(1-i\varepsilon), \ i = 0,1,2,... \\ b_2 &= 1-j\varepsilon, \qquad j = 0,1,2,... \end{aligned} \tag{29}$$

Substituting into Equation (24)

$$f_C = \frac{f_S}{\pi}\tan^{-1}\left(\sqrt{\frac{\varepsilon(2i-j)}{4-\varepsilon(2i+j)}}\right) \tag{30}$$

Analyzing Equation (30) is difficult since it is a function of both i and j. However, we may approximate this equation for ε of sufficiently small size by observing the following:

$$\begin{aligned} 4-\varepsilon(2i+j) &\rightarrow 4 \text{ as } \varepsilon \rightarrow 0, \\ \tan^{-1}(x) &\approx x \text{ for } x \ll \pi \end{aligned} \tag{31}$$

In addition, the term $(2i - j)$ produces a series of integers equivalent to the series i. This leaves us with the following:

$$f_C(i) \approx \frac{f_S}{2\pi}\sqrt{i\varepsilon}, \ f_C \ll f_S \tag{32}$$

For i=1, we see that the second-order direct-form filter has a minimum realizable f_C of approximately 2.64 Hz with 24-bit fixed-point coefficients. The maximum relative error in f_C as a function of frequency is found using Equation (23):

$$error\left(f_C(i)\right) = 1-\sqrt{1-\frac{1}{i}} \tag{33}$$

We may also calculate the error between the desired frequency and the frequency obtained from Equation (24). For comparison, this is shown in Figure 4 along with the maximum error as calculated by Equation

(33). The maximum error is shown as a function of $f_C(i)$, calculated in Equation (32). At low frequencies, Equation (33) closely matches the peak error calculated from the quantized coefficients. This gives us a simple equation for analyzing *frequency* quantization.

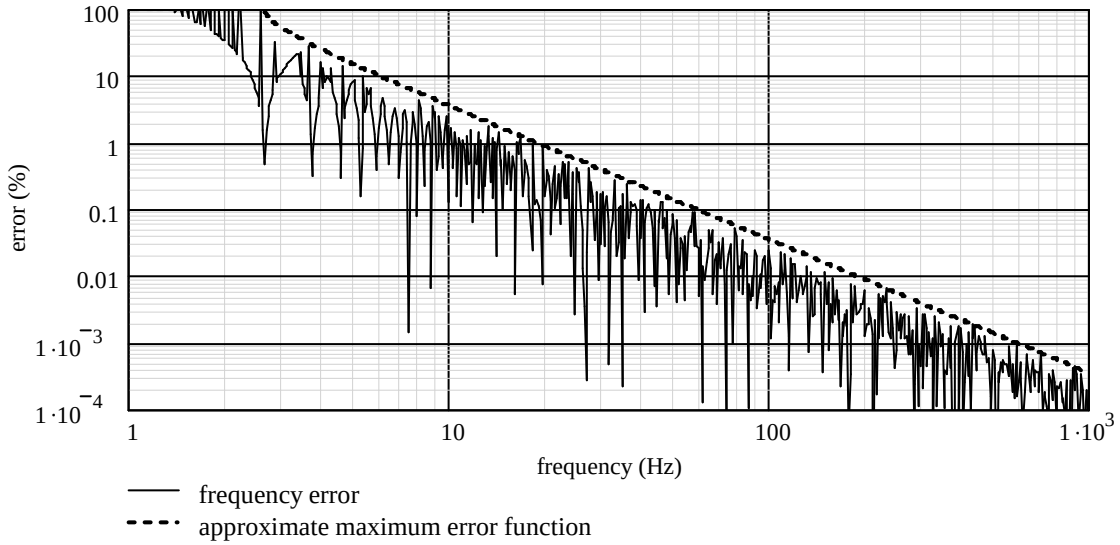

Figure 4. Frequency error of 24-bit fixed-point second-order recursive filter, f_S = 48 kHz.

Frequency error decreases nearly exponentially as f_C increases. By the time f_C reaches 20 Hz, the maximum frequency error has decreased to 0.88% (0.18 Hz, f_S=48 kHz). Changing the quantum size, ε, does not change the shape of the error plot; it simply shifts the error plot along the frequency axis. This shift of the error plot is proportional to the square root of the change in quantum size. Similarly, changing the sampling frequency shifts the error plot along the frequency axis; however, this shift is linearly proportional to the change in sampling frequency.

Consequently, each doubling of the sampling frequency requires a factor-of-4 increase in coefficient resolution – two more bits of precision – to compensate for quantization effects. A summary of the minimum realizable frequencies and maximum frequency error at 20 Hz for common sample rates using 24-bit fixed-point coefficients is presented in Table 4.

Sampling Rate (Hz)	Minimum f_C (Hz)	Maximum Error (Hz) @ 20 Hz
48k	2.64	0.18
96k	5.28	0.73
192k	10.6	2.68

Table 4. Comparison of 24-bit fixed-point quantization effects at common sample rates.

3.2.3 Perturbation of Q

Using the representation of b_1 and b_2 from Equation (29) and substituting into Equation (25) results in the following equation for realizable values of Q:

$$Q = \sqrt{\frac{2i - j}{j^2 \varepsilon} + \frac{1}{4} - \left(\frac{i}{j}\right)^2} \tag{34}$$

Again, this expression is difficult to analyze, because it is a function of two variables. In Equation (35), we make three different approximations that allow us to simplify the expression in Equation (34). In each case, the variable that remains (*i* or *j*) is simply denoted as *i*, since the two series are equivalent.

$$\begin{aligned} &\text{for } i \gg j,\ Q \approx \sqrt{\frac{1}{\varepsilon i}} \\ &\text{for } i \ll j,\ Q \approx \sqrt{i\left(\frac{2}{\varepsilon} - i\right)} \\ &\text{for } i = j,\ Q \approx \sqrt{\frac{1}{\varepsilon i} - \frac{3}{4}} \end{aligned} \tag{35}$$

As in Equation (33), we calculate the error as the maximum relative error between quantized Q values:

$$error\left(Q(i)\right) = \left|\frac{Q(i) - Q(i-1)}{Q(i)}\right| \tag{36}$$

We plot these three approximations of the maximum error in Q as a function of $f_C(i)$ in Figure 5, and we find the three to be very similar at low frequencies. In fact, they are nearly identical to the error in $f_C(i)$ shown in Figure 4. Therefore, we state that the approximate maximum error in both frequency and Q at low frequencies can be estimated by the following expression:

$$\left.\begin{aligned} \text{max error (in f or Q)} &= 1 - \sqrt{1 - \frac{1}{i}} \\ f_C &= \frac{f_S}{2\pi}\sqrt{i\varepsilon} \end{aligned}\right\} \text{ where } i = 1, 2, \ldots \tag{37}$$

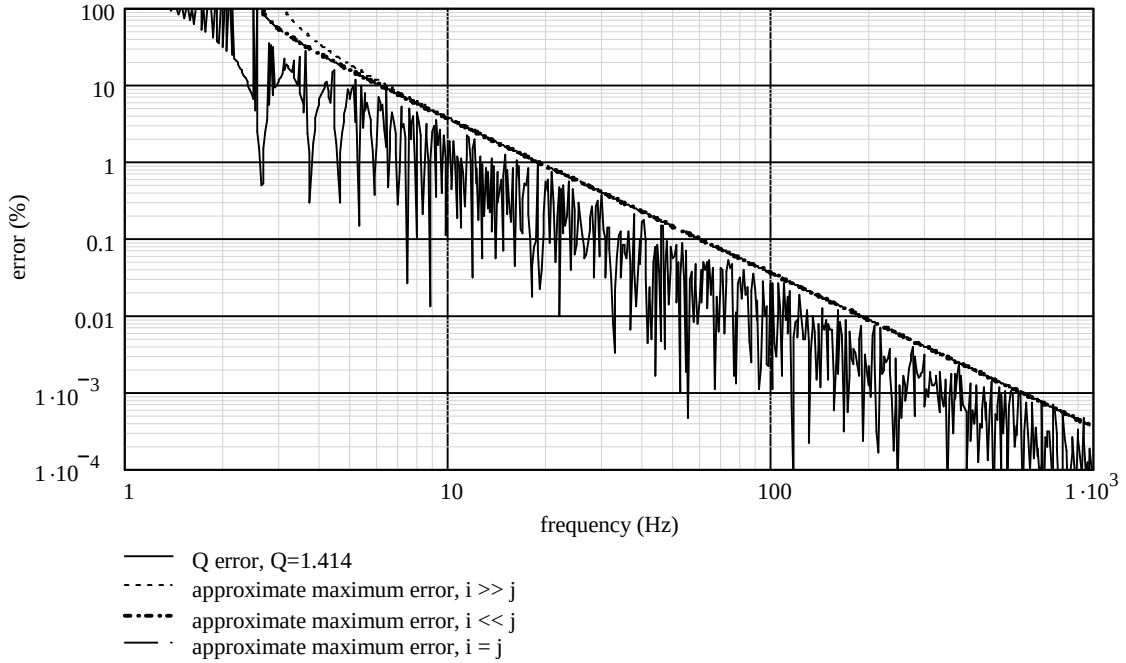

Figure 5. Q error of 24-bit fixed-point second-order recursive filter, f_S = 48 kHz.

3.2.4 Perturbation of V_H, V_B, and V_L

We again consider each common audio filter type separately, since each type is affected differently by quantization. The ranges of Q values were chosen as those typically used in audio filtering:

- Low- and High-Pass: 0.5 to 2.563 (highest Q in an 8th-order Butterworth filter)
- High and Low Shelf: 0.5 to 0.7071 (highest Q with maximally flat frequency response)
- Boost/Cut: 0.3 to 4.318 (Q of a one-third octave graphic equalizer section; bandwidth defined at half power, not half magnitude)

It was noted that while small changes in Q caused significant fluctuations in the maximum error, there was little difference in the general trend of maximum error within the specified ranges. In other words, we are no better off using small or large values of Q, in general.

Filter Type	Parameter	Max Error (%)	@ Freq. (Hz)	Notes
Low-Pass	V_L	5.3	20	$a_0=a_1/2=a_2$ after quantization
	V_B	0		
	V_H	0		
High-Pass	V_L	0		$a_0=-a_1/2=a_2$ after quantization
	V_B	0		
	V_H	1.8×10^{-4}	20k	
All-Pass	V_L	0		$a_0=b_2, a_1=b_1$ after quantization
	V_B	0		
	V_H	0		
High Shelf	V_L	15.4	20	V_H=0.25
	V_B	0.01	20	
	V_H	1.5×10^{-4}	20k	
Low Shelf	V_L	3.5	20	V_L=0.25
	V_B	0.03	12k	
	V_H	100 [5]	12k	
Boost/Cut	V_L	1.8	20	V_B=0.25
	V_B	0.01	20	
	V_H	1.9×10^{-4}	20k	

Table 5. Error in V_L, V_B and V_H of second-order recursive filter, 24-bit fixed-point coefficients, f_S = 48 kHz.

Here, the reader is reminded that the input to the filter is not scaled, which is unrealistic for a fixed-point filter implementation. If input scaling is achieved by modifying the feed-forward (a_i) coefficients, the maximum error in V_L, V_B and V_H (see Table 5) increases significantly.

4 Reducing Second-Order Parameter Quantization

Here we examine each audio filter type separately and, where necessary, suggest methods for reducing the effects of coefficient quantization on the filter's parameters. We commonly find a coupling of quantization effects: for example, a change in frequency or Q may affect the DC gain, V_L. We strive to decouple these effects, when possible.

4.1 All-Pass Filter

Analysis of the second-order direct-form implementation of an all-pass filter reveals that the coefficients of the numerator and denominator become anti-symmetrical. In other words, the transfer function is all-pass if $a_0=b_2$ and $a_1=b_1$ in Equation (10). This being the case, Equation (10) is all-pass regardless of quantization. For the all-pass filter, quantization effects on f_C and Q are fully decoupled from V_L, V_B and V_H.

[5] Narrow spike error; typically < 10^{-4} %

4.2 High-Pass Filter

Analysis of the second-order high-pass coefficients reveals the following:

$$a_0 = -\frac{a_1}{2} = a_2 \tag{38}$$

Provided that this condition is enforced — which is simple despite quantization — equations (26) and (27) show that V_L=V_B=0. From Table 5, we conclude that the error in V_H is negligible.

4.3 Low-Pass Filter

Analysis of the second-order low-pass coefficients reveals that

$$a_0 = \frac{a_1}{2} = a_2 \tag{39}$$

Subsequently, Equations (27) and (28) prove that V_B=V_H=0. However, V_L cannot be guaranteed to be equal to 1; because, due to quantization effects, the numerator and denominator of Equation (26) may not necessarily be equal. As $f_C \rightarrow 0$, the Ω^2 term used in the numerator of the a_n coefficients becomes very small. Consequently, these coefficients become very susceptible to fixed-point quantization error, resulting in a gain error dependent on f_C. Floating-point arithmetic nearly eliminates DC gain error, since the error in V_L is a direct result of the quantization of the a_n coefficients as they become smaller.

Fortunately, this gain error is constant with respect to frequency (for a given f_C), provided that the quantized a_n coefficients meet the conditions in Equation (39). This gain error can be corrected by a simple gain adjustment before or after the filter. The amount of gain adjustment is the reciprocal of V_L computed in Equation (26). Although this method completely corrects the gain error, it does incur a small computational expense.

One method of reducing this error without incurring any computational expense is to remove the double zero in the numerator of the low-pass filter's transfer function, making the transfer function all-pole:

$$L(z) = \frac{a_0}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{40}$$

This double zero is a characteristic of the bilinear transform and guarantees a gain of 0 at $f_S/2$ by warping the frequency axis; removing it results in a response that is subject to frequency-response aliasing. As it turns out, we only need to remove the double zero at very low frequencies — about $f_C < f_S/500$. Typically, at these very low frequencies, there is sufficient attenuation at $f_S/2$ without the need for additional zeros in the transfer function.

If we remove the double zero, the a_0 coefficient becomes

$$a_0 = q\left(\frac{4\Omega^2}{\Omega^2 + \frac{\Omega}{Q} + 1}\right) \tag{41}$$

This moves the factor-of-4 inside the quantization function, effectively reducing the DC gain error by a factor of 4.

We may also force the DC gain to 1 in the all-pole low-pass function. From Equation (26), the DC gain, V_L, for the all-pole filter is computed from the quantized coefficients as follows:

$$V_L = \frac{q(a_0)}{1+q(b_1)+q(b_2)} \tag{42}$$

Forcing V_L to 1 and solving for a_0 yields

$$a_0 = q(1+b_1+b_2) = 1+q(b_1)+q(b_2) \tag{43}$$

While this guarantees that the DC gain is 1, it does incur some error in the cutoff frequency and Q of the filter. This error can become quite apparent when cascading second-order sections to achieve a constrained low-pass response, such as Butterworth, Bessel-Thomson, or Chebyshev. Since the error in the DC gain of a low-pass filter can be quite large for low cutoff frequencies (with fixed-point arithmetic), this is a trade-off that the implementer must consider.

4.4 Shelf Filters

Of all the common types of second-order audio filters, the shelf filters have the worst DC gain error. An example is shown in Figure 6, where the inaccuracy in the DC gain is clearly seen. Fortunately, very low frequency shelf filters are uncommon, which is the region in which this filter has the most DC gain error. Still, it would be nice if we could trade off some frequency and Q accuracy for an improvement in the gain accuracy at DC, since slight deviations in frequency and Q are not as perceptible.

From Equations (24), (25) and (26), we notice that frequency and Q rely only on b_1 and b_2, while V_L relies on all five coefficients. Furthermore, we can allow frequency and Q to deviate slightly to enforce the desired value of V_L. Therefore, we start with Equation (26) and solve for b_1 as a function of V_L and the quantized values of b_2 and a_i:

$$b_1 = q\left(\frac{q(a_0)+q(a_1)+q(a_2)}{V_L} - 1 - q(b_2)\right) \tag{44}$$

Alternatively, we may solve for b_2 as a function of V_L and the quantized values of b_1 and a_i:

$$b_2 = q\left(\frac{q(a_0)+q(a_1)+q(a_2)}{V_L} - 1 - q(b_1)\right) \tag{45}$$

We may also solve for b_1 and b_2 simultaneously as a function of V_L, V_H, and a_i.

$$\begin{aligned} b_1 &= \frac{a_0+a_1+a_2}{2V_L} - \frac{a_0-a_1+a_2}{2V_H} \\ b_2 &= \frac{a_0+a_1+a_2}{2V_L} + \frac{a_0-a_1+a_2}{2V_H} - 1 \end{aligned} \tag{46}$$

Empirically, we find that Equation (45) yields better results than Equations (44) or (46) for minimizing the error in V_L. The summary of these results is shown in Table 6; in the Low Shelf example used, $V_L = 0.4$ and $0.5 \le Q \le 0.707$. We call this *forced DC gain quantization*.

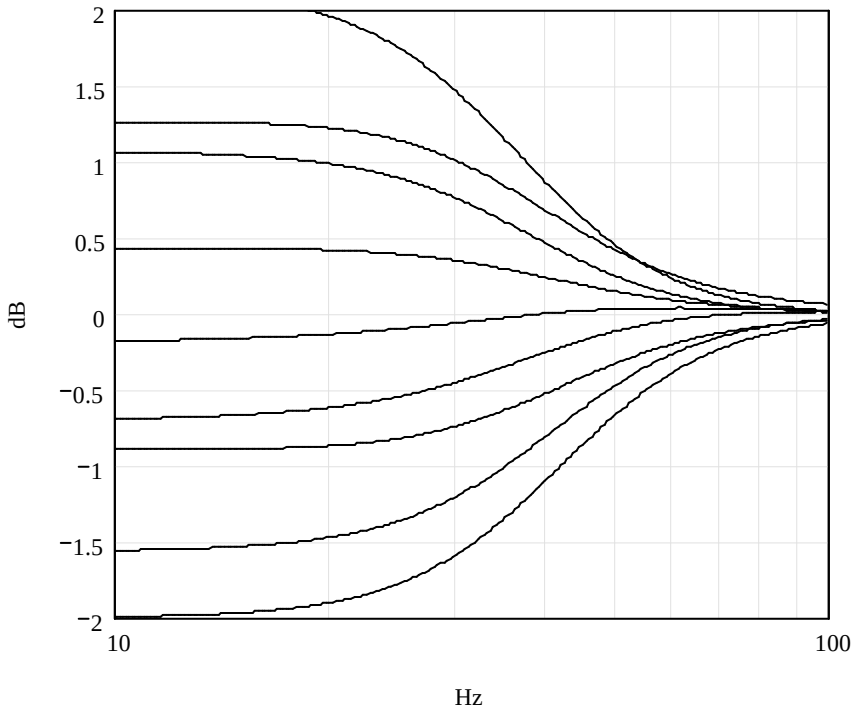

Figure 6. 40 Hz Low Shelf filter, Q = 0.707, 0.5 dB gain increments, 24-bit fixed-point with 2-bit scaling on a_i coefficients, f_S = 48 kHz.

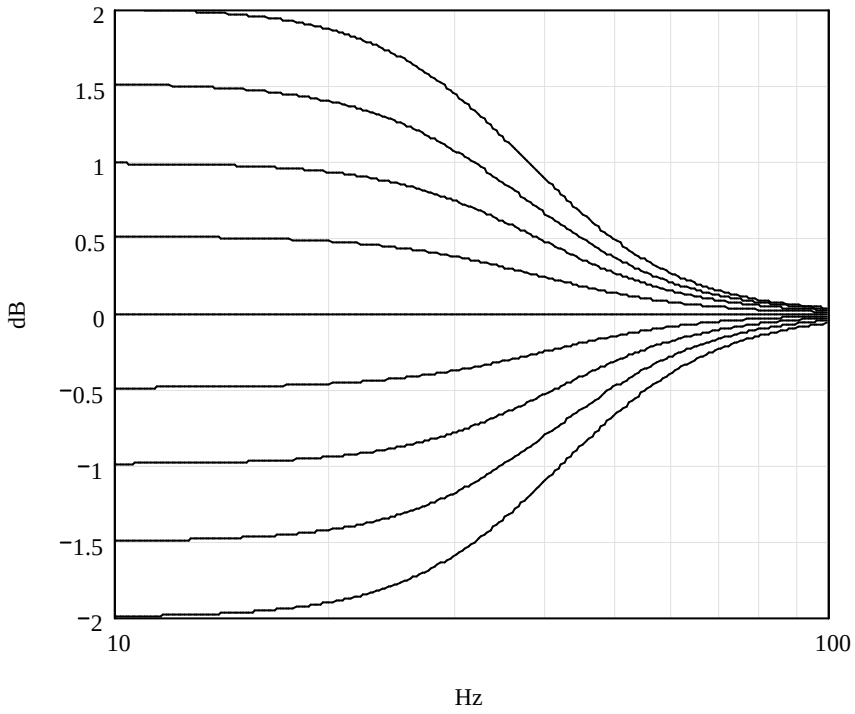

Figure 7. Same filter as Figure 6 except using the forced DC gain method.

	Max Error in V_L @ 20 Hz (%)
Normal quantization	4.76
Using (44)	5.98
Using (45)	0.57
Using (46)	1.12

Table 6. Comparison of error of shelf filter parameters with various quantization methods, 24-bit fixed-point coefficients, f_S = 48 kHz.

The effect on V_B and V_H is not shown in Table 6: while this effect is negligible throughout most of the audio spectrum, there is a significant narrowband error at a high frequency which depends on the value of V_L. For this reason, using the forced DC gain method is only advisable under the following condition:

$$f_c < \frac{f_S}{4}, \text{ for } V_L \geq \frac{V_H}{16} \tag{47}$$

An example of the use of Equation (45) is shown in Figure 7, where a distinct improvement in DC gain accuracy can be seen. The deviation of frequency and Q is imperceptible in the figure.

4.5 Boost/Cut Filter

Like the other second-order filter types, the boost/cut filter can produce significant gain errors. Figure 8 shows an example of the lowest band in a ⅓-octave graphic equalizer at 0.25 dB increments between -1.0 dB and 1.0 dB. Using 24-bit fixed-point coefficients with 2 bits of scaling applied to the a_i coefficients, the resulting frequency responses are severely distorted. This frequency response distortion is a result of error in the DC gain, which can be seen in the figure.

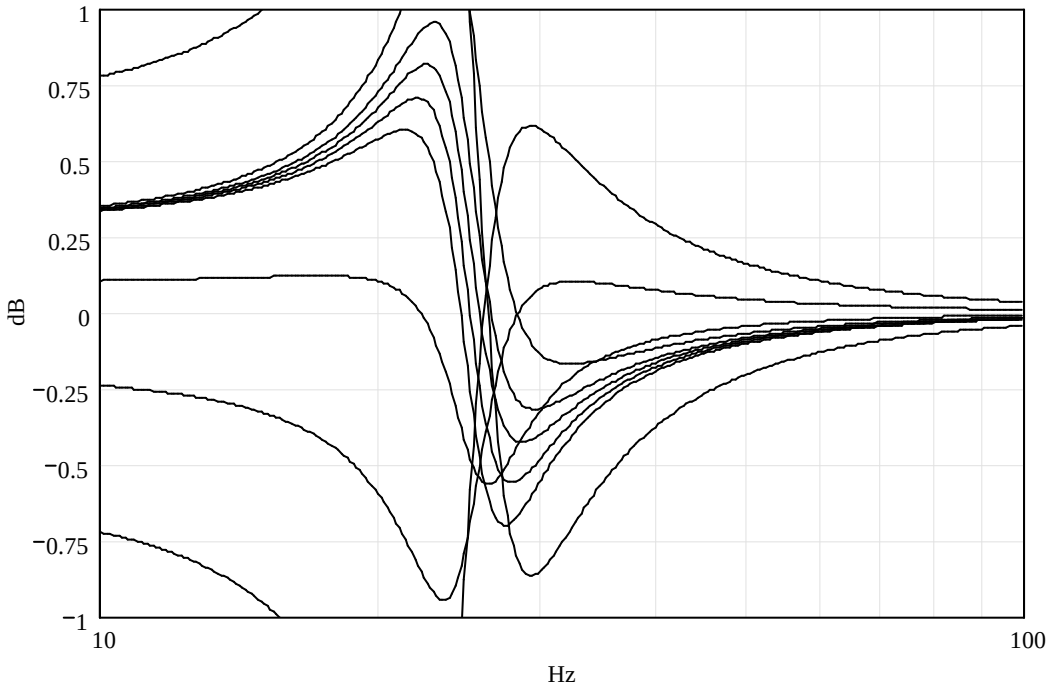

Figure 8. 25 Hz boost/cut filter, Q = 4.318, 0.25 dB gain increments, 24-bit fixed-point coefficients with 2-bit scaling on a_i coefficients, f_S = 48 kHz.

Fortunately, the DC gain error can be eliminated. Regalia and Mitra have shown that a boost/cut filter can be implemented as a linear combination of the input and output of an all-pass filter (Figure 9) [11]. Recall that the all-pass filter remains all-pass despite the quantization of its coefficients. It follows that gain quantization error at DC and Nyquist frequencies are equal to zero in this filter structure.

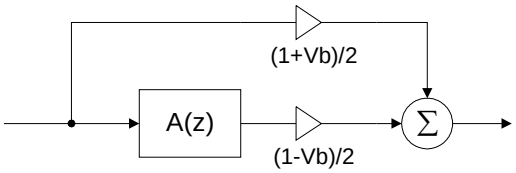

Figure 9. Regalia filter structure.

Simplifying the transfer function of Figure 9 produces our boost/cut transfer function, except that the coefficients in the numerator are mirrored. This results in an equivalent magnitude response, but with excess phase. It therefore seems possible to derive a new set of coefficients for our direct-form boost/cut filter, decomposing the all-pass portion from that of the non-all-pass. For our all-pass function, we use the trivial second-order case. The reason for this becomes obvious: use of a trivial all-pass filter results in a minimum-phase filter. Normally this transfer function would simplify to 1 (hence, it is trivial), but we keep it as such for now:

$$A(z)=\frac{\left(\Omega^2+\frac{\Omega}{Q}+1\right)+2\left(\Omega^2-1\right)z^{-1}+\left(\Omega^2-\frac{\Omega}{Q}+1\right)z^{-2}}{\left(\Omega^2+\frac{\Omega}{Q}+1\right)+2\left(\Omega^2-1\right)z^{-1}+\left(\Omega^2-\frac{\Omega}{Q}+1\right)z^{-2}} \tag{48}$$

The boost/cut transfer function is

$$H(z)=\frac{\left(\Omega^2+\frac{\Omega}{Q}Vb+1\right)+2\left(\Omega^2-1\right)z^{-1}+\left(\Omega^2-\frac{\Omega}{Q}Vb+1\right)z^{-2}}{\left(\Omega^2+\frac{\Omega}{Q}+1\right)+2\left(\Omega^2-1\right)z^{-1}+\left(\Omega^2-\frac{\Omega}{Q}+1\right)z^{-2}} \tag{49}$$

We desire to decompose the coefficients of *H(z)* into their all-pass and non-all-pass components. By inspection, we see that the a_1, b_1 and b_2 coefficients are already equal to their all-pass counterparts. This leaves us with only a_0 and a_2 to compute. Normalizing such that b_0=1, we get

$$\begin{aligned} a_0 &= \frac{\Omega^2+\frac{\Omega}{Q}Vb+1}{\Omega^2+\frac{\Omega}{Q}+1}=1+x_0 \\ a_2 &= \frac{\Omega^2-\frac{\Omega}{Q}Vb+1}{\Omega^2+\frac{\Omega}{Q}+1}=\frac{\Omega^2-\frac{\Omega}{Q}+1}{\Omega^2+\frac{\Omega}{Q}+1}+x_2 \end{aligned} \tag{50}$$

where x_n is the non-all-pass component of the a_n coefficient. Solving for x_n yields

$$x_2 = -x_0 = \frac{\frac{\Omega}{Q}(1-Vb)}{\Omega^2 + \frac{\Omega}{Q} + 1} \tag{51}$$

Quantization of the a_0 coefficient does not change, since

$$q(a_0) = q(1) - q\left(\frac{\frac{\Omega}{Q}(1-Vb)}{\Omega^2 + \frac{\Omega}{Q} + 1}\right) = q\left(1 - \frac{\frac{\Omega}{Q}(1-Vb)}{\Omega^2 + \frac{\Omega}{Q} + 1}\right) = q\left(\frac{\Omega^2 + \frac{\Omega}{Q}Vb + 1}{\Omega^2 + \frac{\Omega}{Q} + 1}\right) \tag{52}$$

However, the new a_2 coefficient becomes quantized as

$$q(a_2) = q\left(\frac{\Omega^2 - \frac{\Omega}{Q} + 1}{\Omega^2 + \frac{\Omega}{Q} + 1}\right) + q\left(\frac{\frac{\Omega}{Q}(1-Vb)}{\Omega^2 + \frac{\Omega}{Q} + 1}\right) \tag{53}$$

Referring to Equations (11)-(15), we observe that we may generalize this approach with Equation (54). Here, b_0 is implied to be 1.

$$q(a_n) = q(b_n) + q(a_n - b_n) \tag{54}$$

Equation (54) implies that the quantization function must be the same for all coefficients. This means that the quantum, ε, must be constant; or in other words, all coefficients must be quantized to the same number of bits to the right of the radix point. We call the quantization method of Equation (54) *all-pass quantization*.

Parameter	**Maximum Error with normal quantization (%)**	**Maximum Error with all-pass quantization (%)**	**@ Freq. (Hz)**	**Notes**
V_L	1.8	0	20	V_B=0.25
V_B	0.01	0.02	20	
V_H	3.2×10^{-6}	0	20k	

Table 7. Error in V_L, V_B and V_H of boost/cut filter with and without all-pass quantization, 24-bit fixed-point coefficients, f_S = 48 kHz.

In Table 7, we see that the error in V_L and V_H has indeed cancelled out. The previous example of the lowest band of a ⅓-octave graphic equalizer is shown in Figure 10, but this time using all-pass quantization on the fixed-point coefficients. There is a significant improvement in the shapes of the filter responses. Although some deviation in center frequency can be observed, it is slight and does not affect the overall shape of the response.

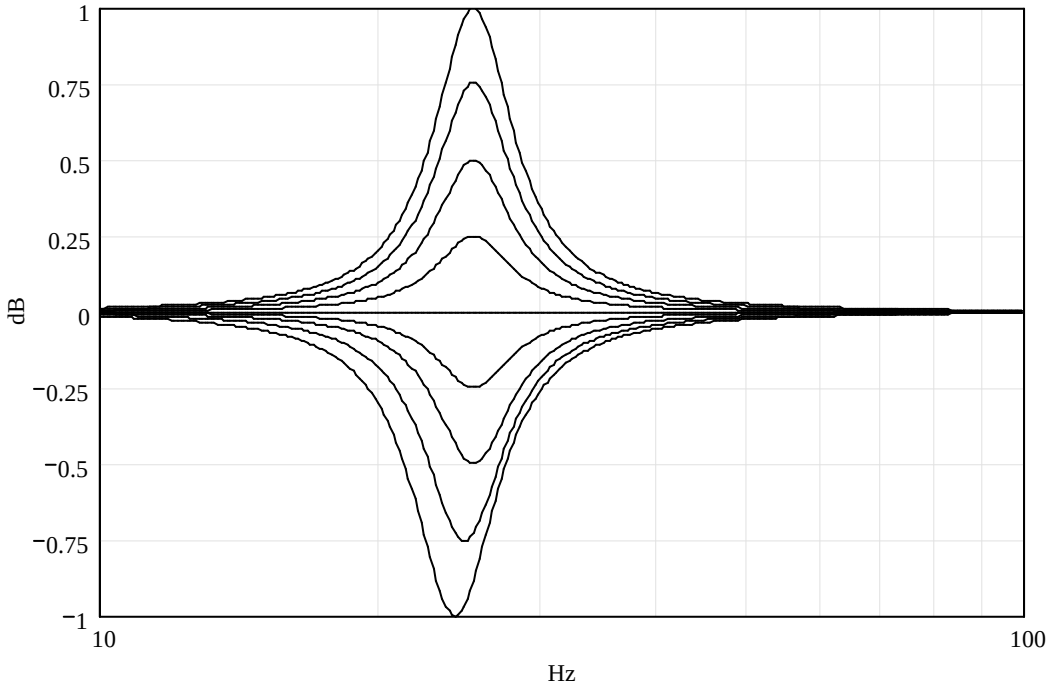

Figure 10. Same filter as Figure 8, but using all-pass quantization.

5 Acknowledgements

The author thanks John Brodie, Laura Mercs, Joe Pham, and Nikhil Sarma for their careful reviews of this manuscript.

6 Conclusion

We have analyzed how coefficient quantization affects the frequency response of the direct-form filter implementation. This analysis is performed by developing filter coefficients from filter parameters, quantizing the coefficients, and reverse calculating the filter parameters. This gives us a clearer understanding of how a filter's parameters are affected by coefficient quantization.

From this analysis, we have developed an expression for estimating the maximum error in frequency and Q resolution at low frequencies. This expression relies only on the number of bits of precision in the quantized coefficients and the sampling frequency. We show that each doubling of the sample rate necessitates two additional bits of coefficient precision to maintain parameter resolution.

Analyzing several types of audio filters, we have found that the gain at DC is susceptible to appreciable quantization error in second-order low-pass, high and low shelf, and boost/cut filters. We have developed simple techniques for reducing or eliminating this error without increasing filter complexity or coefficient precision.

[1] J. Dattorro, "The Implementation of Recursive Digital Filters for High-Fidelity Audio," *J. Audio Eng. Soc.*, Vol. 36, No. 11, pp. 851-878 (1988 Nov.).

[2] R. Wilson, "Filter Topologies," *J. Audio Eng. Soc.*, Vol. 41, No. 9, pp. 667-678 (1993 Sept.).

[3] A. V. Oppenheim, R. W. Schafer and J. R. Buck, *Discrete-Time Signal Processing*, Prentice Hall, New Jersey, 1998.

[4] T. W. Parks and C. S. Burrus, *Digital Filter Design*, John Wiley & Sons, New York, 1987.

[5] M. E. Van Valkenburg, *Analog Filter Design*, Holt, Rinehart and Winston, New York, 1982.

[6] U. Zölzer, *Digital Audio Signal Processing*, John Wiley & Sons, New York, 1997.

[7] *ADSP-2126x SHARC DSP Core Manual*, Rev. 2.0, Analog Devices, 2004.

[8] *TMS320C6000 CPU and Instruction Set*, Texas Instruments, 2000.

[9] *IA-32 Intel® Architecture Software Developer's Manual, Volume 1: Basic Architecture*, Intel, 2004.

[10] D. P. Rossum, "Dynamic Digital IIR Audio Filter and Method which Provides Dynamic Digital Filtering for Audio Signals," U.S. Patent 5,170,369 (1992 Dec.).

[11] P. A. Regalia and S. K. Mitra, "Tunable Digital Frequency Response Equalization Filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, pp. 118-120 (1987 Jan.).