Case Study of Scaling and Product Quantization in Digital Filter Realizations

Ross L. Penniman, *Graduate Student*

***Abstract*—Digital filters are always subject to adverse effects of computational precision when implemented in hardware or software. This paper shows that the errors introduced by the computations can be represented as a noise source within the digital filter structure. As an example, a fourth order high pass filter is examined. This filter is taken from the post-filter of the HVXC Decoder, which is part of the MPEG-4 audio specification.**

***Index Terms*—Finite wordlength effects, HVXC, L_p Norm, Product quantization**

I. Introduction

DIGITAL filters are essentially a collection of multiplication and addition operations, which when implemented in systems with finite precision, are subject to errors. These errors can be manifest in many ways. When the coefficients of the filter are quantized, the locations of the poles and zeros may change. This includes the possibility that poles which are close to the unit circle can actually move outside the unit circle and make the filter unstable. Another possible consequence of round-off errors is the phenomenon of limit cycles, which can either be very small or very large oscillations that occur without any input to the filter. The aspects of finite precision that are the focus of the this paper, however, are the effects of scaling and product quantization.

Whenever two numbers are multiplied together in a digital system, the product of those two numbers requires twice as many bits as each individual number. This number must then be reduced back to the bit-depth of the system. Methods such as truncation and rounding each have advantages and disadvantages. In both cases, information about the number is being lost, and is generally referred to as quantization.

Implementing a digital filter with fixed-point number representations can perform much faster than using floating-point number representations. One of the challenges with using fixed-point numbers, however, is the possibility of overflow. Close attention must be paid to the magnitudes of the numbers so as not to exceed the valid range of values the system can represent. Overflow has serious consequences in a digital filter, including unwanted output values of large amplitude.

In order to prevent overflow, digital filters are analyzed to find how values can be scaled up or down as they pass through each section of the structure. From this analysis, scaling factors are determined which are applied to the signal before, after, or within the structure of the digital filter in order to keep the signal within the required bounds.

The filter to be studied in this paper is a post-processing filter applied to an HVXC decoder. HVXC stands for Harmonic Vector Excitation Coding and is a parametric method of encoding speech signals as part of the MPEG-4 audio specification [1], [2], and [3]. The filter is a fourth order high-pass filter implemented in two cascaded second order sections (bi-quads). Each section (bi-quad) is implemented using the direct form II structure. The overall filter is given by the transfer function

$$H_{HPF}(z) = K_{HPF}\left(\frac{1 + b_{11}z^{-1} + b_{12}z^{-2}}{1 + a_{11}z^{-1} + a_{12}z^{-2}}\right)\ldots \left(\frac{1 + b_{21}z^{-1} + b_{22}z^{-2}}{1 + a_{21}z^{-1} + a_{22}z^{-2}}\right) \quad (1)$$

with the coefficients given by Table I:

TABLE I
FILTER COEFFICIENTS

Coefficient	Value
K_{HPF}	+1.100000000000000
b_{11}	-1.998066423746901
b_{12}	+1.000000000000000
a_{11}	-1.962822436245804
a_{12}	+0.9684991816600951
b_{21}	-1.999633313803449
b_{22}	+0.9999999999999999
a_{21}	-1.858097918647416
a_{22}	+0.8654599838007603

II. Coefficient Quantization

The frequency response of the filter is not affected significantly by finite wordlength effects. Since it is a relatively low-order filter, and is implemented in cascade form, the locations of the poles and zeros are relatively insensitive to round-off errors. The difference in frequency response for an ideal representation versus a 12 bit representation can be seen to be fairly small, Fig 1.

Manuscript received December 4, 2012. R. L. Penniman is a Graduate Student in the Music Engineering Department, Frost School of Music at the University of Miami, Coral Gables, FL 33124 USA (e-mail: rosspenn@ gmail.com).

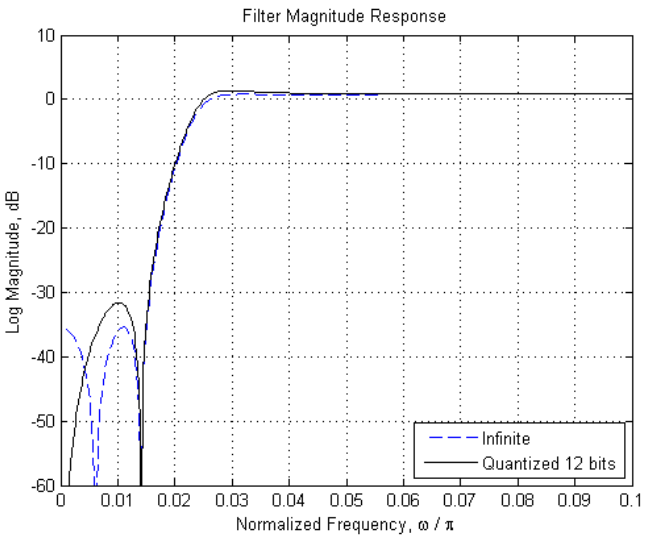

Fig. 1. Comparison of frequency response of infinite precision vs. coefficients quantized to 12 bits.

III. The L_p Norm

The L_p norm is a useful tool for finding an appropriate scaling factor to ensure that a filter does not overflow. The norm is given by equation

$$\|H\|_p = \left[\frac{1}{2\pi}\int_{\omega=-\pi}^{+\pi} |H(\omega)|^p \, d\omega\right]^{1/p}, p \geq 1 \qquad (2)$$

with the requirement that:

$$\int_{\omega=-\pi}^{+\pi} |H(\omega)|^p \, d\omega < \infty \qquad (3)$$

The most commonly used norms are $p = 1$, 2, and ∞. For $p = 1$, the norm represents the mean of the magnitude over the frequency. For $p = 2$, the norm represents the average power over the frequency. Finally, for $p = \infty$, the norm is simply the maximum of the transfer function:

$$\|H\|_\infty \equiv \lim_{p\to\infty} \|H\|_p = \max_{\omega\in[-\pi,+\pi)} |H(\omega)| \qquad (4)$$

As a more intuitive approach, we can see that the L_2 norm is more appropriate for cases where the input is a random signal, and the L_∞ case is more appropriate when the input signal is concentrated at a single frequency [5]. In the case of the HVXC decoder, the expected input signal is speech, which contains a wide spectrum of frequency components, and thus is most similar to a random signal. Using the L_∞ norm results in a more conservative scaling, and is less likely to produce overflow. Using the L_2 norm results in better signal-to-noise ratio but at a higher risk of overflow. The effect of scaling factors will be examined in greater detail later. In the case of the speech signal, it may be necessary to use the L_∞ norm in order to ensure robust performance without any overflow, since the exact characteristics of the signal are unknown. For this study only the L_2 and L_∞ norms were examined.

The norm can be calculated in MATLAB by calling the `filternorm` function or by calculating an approximate integral of $H(\omega)$ directly.

IV. Choosing Scaling Factors

The filter being examined consists of two bi-quad sections in cascade. Each of these sections is implemented using the direct form II, Fig 2.

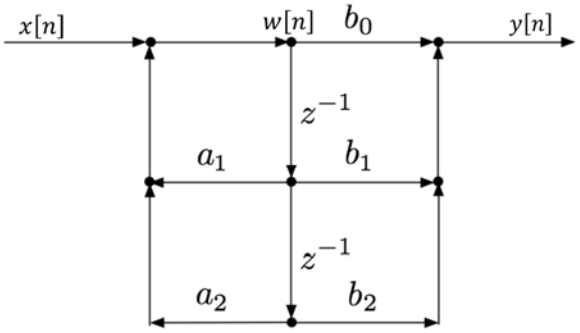

Fig 2. Direct Form II structure of an IIR filter

The difference equation for direct form II can be given in two parts:

$$w[n] = x[n] + a_1 w[n-1] + a_2 w[n-2] \qquad (5)$$

$$y[n] = b_0 w[n] + b_1 w[n-1] + b_2 w[n-2] \qquad (6)$$

In this paper we will examine the case of fixed-point numbers, as they are much easier to analyze. Fixed-point calculations are especially susceptible to overflow, however. Typically, numbers are represented as some fraction of 1, therefore the permissible range is ± 1. Any point where two or more numbers are added together is subject to overflow. When using the two's complement method of representing negative numbers, partial sums are allowed to overflow as long as the total summation does not overflow. Looking at the direct form II structure, there are two nodes where such summations happen. In order to do a full analysis, it is best to consider these two nodes separately. Equations (4) and (5) show that the poles and zeros of the bi-quad filter section can be handled independently. Therefore we can represent the bi-quad by two transfer functions:

$$H(z)_p = \frac{W(z)}{X(z)} \qquad H(z)_z = \frac{Y(z)}{W(z)} \qquad (7)$$

Since most of the gain of the filter comes from the poles, we would ideally like to have separate scaling factors for the $H(z)_p$ and $H(z)_z$ portions of the transfer function. The complete system, with scaling factors is shown in Fig. 3.

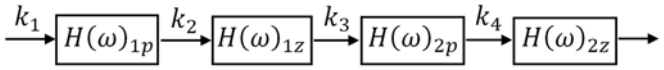

Fig 3. Flowchart of complete system.

Factors k_1 through k_4 scale the signal prior to each respective filter stage to prevent overflow in that stage. Determining the values of the scaling factors is an iterative process.

$$k_1 = \frac{1}{\left\| H(\omega)_{1p} \right\|_p} \tag{8}$$

$$k_1 k_2 = \frac{1}{\left\| H(\omega)_{1p} H(\omega)_{1z} \right\|_p} \tag{9}$$

$$k_2 = \frac{k_1 k_2}{k_1} \tag{10}$$

This iteration can be continued to attain values for all four scaling factors. The total product of the scaling factors should be as close to 1 as possible, so as to not change the overall gain of the filter.

While having four scaling factors is the ideal case, in order to use as few delays in the filter structure as possible, and implement the direct form II exactly, there is no way to implement the k_2 and k_4 scaling factors as they are indicated. Placing a multiplier before $w[n]$ would affect the filter poles, and placing a multiplier after $w[n]$ would skew the ratio of the b coefficients, thus changing the locations of the zeros. As a compromise, factors k_2 and k_4 are moved after their respective filter sections, so that $\hat{k}_3 = k_2 k_3$, and k_4 is now a post-scaling value. The need for scaling factors is illustrated in Fig. 4.

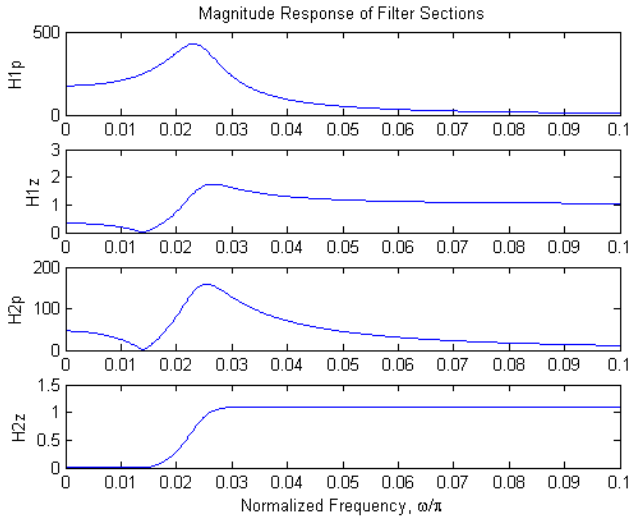

Fig. 4 Magnitude response at outputs of the four filter sections.

As can be seen in Fig. 4, the gain of each section containing poles is very large. It should also be noted that the factor K_{HPF} has been left out, as it only exacerbates the problem of overflow. If it can be determined that the signal will not cause overflow with the K_{HPF} term in place, it can be combined with k_4 as a post-scaling term. It is more likely that an additional reduction in each scaling term will be required to account for possible overflow values created by quantization errors. The idealized scaling terms for the HVXC filter using the L_2 norm and L_∞ norm are shown in Tables II and III respectively.

TABLE II
SCALING FACTORS USING L_2 NORM

k_1	0.018897845538730
k_2	51.465560648575796
k_3	0.052621485721066
$\hat{k}_3$	2.708194264795685
k_4	18.093735057825690

TABLE III
SCALING FACTORS USING L_∞ NORM

k_1	0.002342592572839
k_2	244.3464731016665
k_3	0.011000784483160
$\hat{k}_3$	2.688002889811761
k_4	144.8772169828914

V. EFFECTS OF PRODUCT QUANTIZATION

A. *Theory of Quantization Noise*

For a given number of bits, b, and a system that can handle values in the range $\pm A$, the smallest value, or smallest difference, able to be represented by the system is:

$$q = \frac{2A}{2^b} \tag{11}$$

To simulate the effects of quantization, we add a noise source to each multiplier (for coefficients that are not equal to 1). Each noise source has the following properties [4]:

1) Wide-sense stationary process with mean and variance given by:

$$\mu_e = 0, \qquad \sigma_e^2 = \frac{q^2}{12} \tag{12}$$

2) Uniformly distributed in the quantization interval $[\frac{-q}{2}, \frac{q}{2}]$

3) Uncorrelated to the input of the quantizer and to other noise sources.

The variance of the noise signal σ_e^2 is also the average power of each noise source. Since the noise sources are uncorrelated, multiple noise sources can be added linearly. The overall signal-to-quantization-noise ratio (SQNR) is given by:

$$SQNR = \frac{\sigma_x^2}{\sigma_e^2} = 1.76 + 6.02b \ \ dB \tag{13}$$

Where x is the input signal to the given quantizer.

An estimate of the total noise in the system can be estimated by multiplying the total number of multipliers by the error power σ_e^2. This can be compared to the average power of the signal σ_x^2 which can be approximated by

$$\sigma_x^2 = \frac{A^2}{2} \tag{14}$$

where A is the amplitude of the signal (generally taken to be 1). For our example filter, we have 7 coefficients not equal to 1, and 3 scaling factors. If we assume that the input signal has an amplitude of ± 1 then we can compute the estimated total noise power to be:

$$\sigma_{eT}^2 = (7+3) * \left(\frac{1}{3 * 2^{2b}}\right) \tag{15}$$

We assume an average input power of $1/2$ and a resolution of 12 bits, we have an estimated SQNR of 2.52×10^6 or about 64 dB. This is only a rough estimate, however, as there is a significant dependence on the scaling factors and the frequency response of the filter.

B. Measurement of quantization noise

The SQNR of the filter can be found in practice by simulating it in MATLAB. The difference equations (5) and (6) were implemented in a `for` loop, with the `dec2beqR` function (provided by [4]) used to simulate quantization. Scaling factors were also inserted in the appropriate places and also quantized. The input signal was 10,000 points of a random signal uniformly distributed between (-1, 1). The signal was processed both at infinite precision and at some reduced precision. The empirical SQNR is given by:

$$SQNR_{empirical} = \frac{\sigma_{inf}^2}{\sigma_{quant}^2} \tag{16}$$

$$SQNR_{dB} = 10 \log_{10}\left(SQNR_{empirical}\right) \tag{17}$$

The results are given in Table IV.

TABLE IV
QUANTIZATION NOISE MEASUREMENTS

***b*, bits**	***Lp norm***	***SQNR***	$SQNR_{dB}$
12	2	684.09	28.4
12	∞	24.854	14.0
16	2	169463	52.3
16	∞	5295.6	37.2

C. Analysis of Measured Results

The data follows the expected trends of having better SQNR for larger values of b, as well as better SQNR for using the L_2 norm instead of the L_∞ norm. What is clear, is that even in the best test case, the SQNR is still significantly worse than what was predicted by (13), (14), and (15). The reasons for this are most likely that the noise introduced by quantization did not have a uniform distribution as was assumed. The noise was also most likely correlated with the quantization inputs and other noise signals, violating another one of our assumptions.

The large discrepancy between predicted and measured results also indicates that the scaling factors and the frequency response of the filter play a large part in determining the SQNR. The frequency response of the filter has a sharp cutoff, and this is achieved by having its poles close to the unit circle. The scaling required to keep these poles from causing overflow has a serious detrimental effect on the SQNR.

VI. CONCLUSION

Digital filters are generally designed with assumption of infinite precision. When filters are implemented in hardware or software, however, consideration must be made for how the finite precision of the system affects the performance of the filter. This paper reviewed the principles of scaling and product quantization when using fixed-point computations. The principles were then used to analyze the performance of a fourth order high-pass filter that is part of the HVXC decoder. The necessary scaling parameters were determined based on L_p norms, and the product quantization effects were measured at both 12 bits and 16 bits resolution.

REFERENCES

[1] Battista, Stefano, Franco Casalino, and Claudio Lande. "MPEG-4: a multimedia standard for the third millennium. 1." Multimedia, *IEEE* 6, no. 4 (1999): 74-83.

[2] Battista, Stefano, Franco Casalino, and Claudio Lande. "MPEG-4: a multimedia standard for the third millennium. 2." Multimedia, IEEE 7, no. 1 (2000): 76-84.

[3] *Information Technology—Very Low Bitrate Audio-Visual Coding, Part 3 Audio*, MPEG Working Group, International Standards Organization/ International Electrotechnical Commission (ISO/IEC) Std. ISO/IEC FCD 14 496-3 Subpart 1, May 1998. [Online]. Available: http://www.mp3-tech.org/programmer/docs/ISO_14496-3.pdf

[4] D. G. Manolakis, V. K. Ingle, *Applied Digital Signal Processing*, New York, NY: Cambridge University Press, 2011, pp. 936-944.

[5] Dattorro, Jon. "The implementation of recursive digital filters for high-fidelity audio." *J. Audio Eng. Soc* 36, no. 11 (1988).