The Cooper Union Department of Electrical Engineering
Prof. Fred L. Fontaine
ECE310 Digital Signal Processing
Problem Set IV: Quantization
November 4, 2019

1. In this problem, we define "truncation" to mean simply discarding the LSBs on the right, without adjusting the other bits that are kept. Also, we assume rounding operations are performed first, followed by overflow operations. For example, first round up if necessary, then test for overflow.

   Consider the following two's complement fixed point values:

   00110011.010101001
   110101010101.1010101

   Do not compute the numerical values. Just produce the two's complement codes as specified. In all cases, show the code AFTER you have performed roundoff but BEFORE you have applied the overflow operation, and then show the FINAL ANSWER.

   (a) Produce the (5).(4) codes assuming: roundoff by rounding with two's complement overflow; and roundoff by truncation with saturation overflow.

   (b) Produce the (4).(3) codes assuming roundoff by rounding with two's complement overflow; and roundoff by truncation with saturation overflow.

2. For each of the following two's complement values, write the code with the fewest number of bits that can represent the same value exactly. Do this without computing the numerical values.

   0001011.011100
   11111111111101.01100

3. Now compute the numerical values for your answers to 2, left in the form of a fraction $\left(3\frac{3}{4}, \text{etc.}\right)$.

4. **Sensitivity Properties of Parallel Allpass Realizations**

   When a filter is decomposed as the sum of two lower order filters, e.g., $H(z) = H_1(z) + H_2(z)$, it is called a parallel configuration; this contrasts with the cascade configuration which corresponds to $H(z) = H_1(z) H_2(z)$. It turns out that a large class of digital filters (including Butterworth, Chebyshev and elliptic filters) can be realized by a parallel pair of allpass filters. In general, the allpass filters will have complex coefficients, even if the overall filter transfer function has real coefficients. However, there are certain situations where the allpass functions can have purely real coefficients, and that is the case in the example you will explore here. It turns out using lossless building blocks in DSP structures, as in this case, can provide significant advantages in terms of quantization effects. Here you will explore the sensitivity advantages of parallel allpass realizations.

The case we take is a sixth order bandpass elliptic filter, 2dB passband ripple, 30dB stopband ripple, passband from 0.2 to 0.4 on a normalized scale (where 1 is the Nyquist bandwidth). This filter can be designed in MATLAB with the code:

$$\begin{aligned} [z,p,k] &= ellip(3,2,30,[0.2,0.4]); \\ [b,a] &= zp2tf(z,p,k); \\ [d0,d1] &= tf2ca(b,a); \\ p0 &= fliplr(d0); \\ p1 &= fliplr(d1); \end{aligned}$$

Note that the order parameter passed here is 3, which is the order of the lowpass prototype that underlies the sixth order bandpass filter. Here let:

$$H(z) = \frac{b(z)}{a(z)} \quad H_1(z) = \frac{p_0(z)}{d_0(z)} \quad H_2(z) = \frac{p_1(z)}{d_1(z)}$$

Then it turns our H_1, H_2 are all-pass and:

$$H(z) = \frac{1}{2}\left[H_1(z) - H_2(z)\right]$$

The coefficients b, a would be the multiplier coefficients in a direct form II (or II transposed) realization of H. The MATLAB function $tf2ca$ performs this allpass decomposition on a transfer function. Note in this case H is expressed as a difference of two allpass functions, not a sum! Also, in practice the $\frac{1}{2}$ factor would be implemented as a shift, so does not count as a multiplier.

In this problem you will be rounding off the coefficients in $b, a, p0, p1, d0, d1$ and observing the effect on the system. It turns out $a, d0, d1$ are all set so the lead coefficient in the vector is 1 (recall that happens when we try to convert a transfer function to a direct form type realization). Since $p0, p1$ are flipped from $d0, d1$, they also contain a coefficient value 1. However, if you look at b, all the coefficients are fairly small. Let us introduce a scaling factor of 8; that is, define $bscale = 8$. This will allow us to extract a shift, and then round b coefficients without zeroing them out!

For simplicity, we will not try to exactly match the bit-level operations performed in fixed-point. Instead, here, to round the vector x to the nearest 1/4th, I mean $round(x * 4)/4$. For the case of b, remove the scaling factor first, for example:

$$round((b * bscale) * 4)/(bscale * 4)$$

Before you commence the problem, I will make a few comments. There are a number of reasons that allpass systems exhibit excellent robustness. For the moment, I will focus on the fact that the numerator and denominator polynomials are the same but in reverse order. If we round off the coefficients to finite precision, then (in most realizations) the multiplier coefficients occur in matched pairs, and will remain matched even when rounded, since they are rounded to the same values. Now consider a parallel form $(H_0 \pm H_1)/2$. If the allpass property of H_0, H_1 are STRUCTURALLY ensured

(meaning it is ensured regardless of the values of the multiplier coefficients; this is almost universally employed for allpass systems), then the maximum gain is 1 (it could be less when they do not match in phase, but cannot exceed 1). Suppose the composite filter H has nominal gain 1 at some frequency ω_0 (e.g., the peak of a ripple in the passband). Imagine the variation of $|H(\omega_0)|$ with respective to some multiplier coefficient in the system, say α. Since the value $|H(\omega_0)| = 1$ is maximal, we must have $\partial |H(\omega_0)| / \partial \alpha = 0$!!! This is a manifestation of good sensitivity.

(a) Write a function in MATLAB to do the following. The input is a pair of transfer functions H_1, H_2 represented with vectors $[b1, a1], [b2, a2]$, respectively. Compute the numerator and denominator vectors for $\frac{1}{2}(H_1 - H_2)$. **Hint:** Pretend you were doing this by hand, placing everything over a common denominator; code this up in MATLAB, as a hint using the *conv* function. Also, do not worry about trying to detect and cancel out common factors. In any case, do not use the symbolic toolbox for this.

(b) Compute the frequency response of H and $\frac{1}{2}[H_1 - H_2]$, and compute the maximum absolute error between the two.

(c) Plot the magnitude and phase responses of H, the frequency axis scaled from 0 to 1 (being Nyquist bandwidth, i.e. equal to π rad). The magnitude response should go from -50dB up to a value large enough to capture the maximum gain of H.

(d) Round off b, a to the nearest 16th, and again to the nearest 4th, following the comments given above. Superimpose the magnitude responses of the three (the original and the two quantized forms); the decibel scale should go from -50dB to the maximum gain of any of them.

(e) Using 3 subplots, obtain the zero-pole plots for H and the two quantized forms of it. Comment on the poles!!!

(f) Now round off $p0, p1, d0, d1$ first to the nearest 16th, then to the nearest 4th. Compute the resulting frequency responses. Superimpose the magnitude response of the original and the two quantized forms, as you did with the direct form H above.

(g) Obtain the *zplane* plots for the original filter and the ones associated with quantizing the allpass sections, as above. Comment on the poles!!!