ELECTRONOTES
1 Pheasant Lane
Ithaca, NY 14850
(607)-273-8030

APPLICATION NOTE NO. 345

October 1997

IMPROVED SIGNAL/NOISE RATIO WITH FIRST-ORDER NOISE SHAPING: AN EXAMPLE

The concept of oversampling is likely familiar [1-4]. One can list a good number of reasons for using oversampling: less quantization noise, simpler analog anti-aliasing (input guard) and anti-imaging (reconstruction, or smoothing) filters, less phase distortion, and several others. In fact, there is no actual performance downside to using oversampling, and the actual implementation costs (over just ordinary sampling) are usually minimal. It's just a good idea - and that's about it. Perhaps this can be understood in terms of a denser set of samples being, just naturally, a better and more complete representation of a signal.

One of the free benefits of oversampling is an improvement in signal-to-quantization-noise ratio (which is free if we assume we have already decided to do oversampling for some other reason). It is well known that we get an extra half-bit (3db) improvement in S/N for each octave (factor of two) of oversampling. For example, if we have times-8 oversampling (three factors of 2), we would get a 9db improvement in S/N. More importantly, this free benefit of oversampling can in fact be enhanced by the use of so-called noise shaping. It is the purpose of this note to illustrate how noise shaping can enhance performance through a simple example. At the same time, we will look at the original cases (no noise shaping, and no oversampling) for comparison.

We start with the idea that a standard white random-noise model can be used for the quantization error. That is, the noise amplitude is uniformly distributed (conveniently set to 1) from 0 to half the sampling frequency (Fig. 1). This noise is all "audible" for our purposes here. What we usually do is integrate the power spectrum from 0 to half the sampling frequency, which is traditionally done from 0 to π.

$$p_0 = \int_0^{\pi} 1^2 \, d\omega = \pi \tag{1}$$

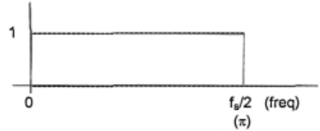


Fig. 1 Uniform Distribution of quantization error according to standard modeling

Here $p_0=\pi$ will serve as a reference level. We are not overly concerned with what the actual value of p_0 is - we just want to look at oversampling and noise shaping relative to p_0 (do we gain or do we lose).

One way to decrease the noise power below p_0 would be to reshape the noise spectrum of Fig. 1 so that the integrated power would be less than π. However, this can not be done arbitrarily. For one thing, we want to shape the noise without shaping the signal itself, so something like an ordinary filter, in series, is out of the question. One possible thing to do, however, would be to integrate from 0 to some frequency less than π. This we could justify if it were true that the noise above this upper frequency is inaudible. When we implement oversampling, we generally get an upper portion of the spectrum that is inaudible. For example, with times-2 oversampling, the sampling frequency is doubled, and the quantization noise is distributed over a frequency range that is twice as wide, while at the same time, the audible region does not change. [Here, by "audible" we are considering the frequency limitations of amplifiers, reconstruction filters, loudspeakers, etc., as well as the frequency characteristics of the human ear.]

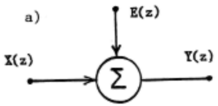
Thus, effectively in the case of times-2 oversampling, we do the integral of equation (1) only to $\pi/2$.

$$p_1 = \int_0^{\pi/2} 1^2 \, d\omega = \pi/2 \qquad (2)$$

which is clearly only half the power. In terms of signal amplitude (voltage if you prefer) this would be $1/\sqrt{2}$, which would be a 3db improvement. Since a 2:1 improvement would be what we would get from one extra bit (the quantization interval would be 1/2 its previous value), we can say that we get a half-bit improvement for each octave of oversampling. Not much - but it's free!

In a sense, oversampling is a crude form of noise shaping, as we are effectively shaping the noise (spreading it over a wider frequency range) so that half of its power is inaudible. It would be more helpful if the noise spectrum could be shaped so that a much greater portion of its power were in the region that will become inaudible. The trick, as we mentioned above, is to shape the noise without shaping the signal itself. To do this, we need to consider how noise is inserted in the quantization process. Fig. 2a shows the standard model: the quantization error E(z) is simply added to the signal X(z). One way to look at this is to consider an A/D converter followed by a D/A converter. The quantization error is the difference between the discrete analog signal at the input of the A/D and the discrete analog signal at the output of the D/A.

In contrast, a more complicated system is seen in Fig. 2b. As in Fig. 2a, the noise is summed with the signal at the point of quantization, but the signal X(z) has already been filtered by the time it reaches this point . In fact, the entire structure is a loop with the signal inserted at one point, and the noise at another.



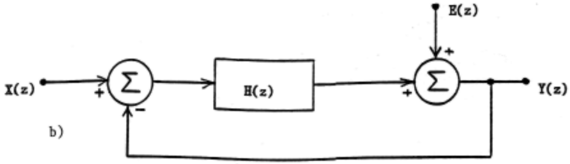


Fig. 2 Ordinary quantization (a) is modeled as an additive error E(z). With first-order noise shaping (b) quantization occurs in a loop structure, and the noise E(z) can be shaped without shaping the signal X(z).

This second structure is a first-order noise shaper, and it can be confusing, since it appears to be a digital filter - and we have represented it as such. It will be useful to recognize that this is actually, a discrete-time filter which represents the functions of the quantizer itself: an A/D converter often called a Sigma/Delta converter. Thus we are seeking to understand the converter and are using this discrete-time filter model to do so. This understood, we note that the output Y(z) is the sum of two terms:

$$Y(z) = E(z) + H(z)[\,X(z) - Y(z)\,] \tag{3}$$

From this, we see that the signal itself is subject to a transfer function:

$$Y(z)/X(z) = H(z) / [\,1 + H(z)\,] \tag{4a}$$

while the noise is subject to a different transfer function:

$$Y(z)/E(z) = 1 / [\,1 + H(z)\,] \tag{4b}$$

These are relatively meaningless until we get to the point of putting in some specific filter H(z). If we make H(z) a simple, first-order discrete-time integrator, $H(z)=z^{-1}/(1-z^{-1})$, we have:

$$H_X(z) = Y(z)/X(z)|_{E(z)=0} = z^{-1} \qquad (5a)$$

which is a pure unit delay, which leave the signal spectrum unshaped. In contrast, the transfer function for the noise is:

$$H_E(z) = Y(z)/E(z)|_{X(z)=0} = 1 - z^{-1} \qquad (5b)$$

which is a simple high-pass filter. This is good because we wanted to move the noise power preferentially to higher frequencies where they would become inaudible. What is not so good is that this particular filter in equation (5b) has a gain of 2 at half the sampling frequency. That is, while we shape the noise the way we want, we also amplify it, and there is no way of simply changing the gain of this filter.

None the less, the situation is quite good as long as we are planning to use a substantial oversampling factor. With the filtering of equation (5b), we have:

$$|H_E(e^{j\omega})|^2 = [(1 - e^{-j\omega})(1 - e^{j\omega})] = 2 - 2\cos(\omega) \qquad (6)$$

The magnitude, and the magnitude squared as in equation (6) is plotted in Fig. 3.

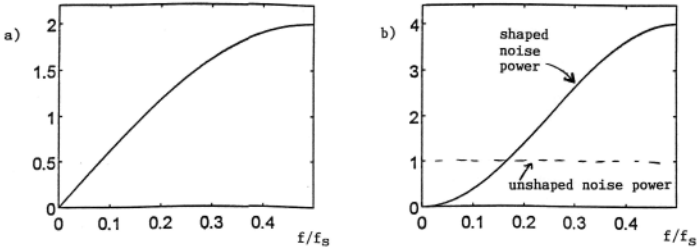


Fig. 3 Magnitude (a) and squared magnitude (b) of $H_E(z) = 1 - z^{-1}$. The squared magnitude (b) corresponse to the shaping of the noise power. For comparison, the unshaped noise power is seen in (b) as the dashed line.

All that remains to do is to integrate equation (6) from 0 up to some fraction of π. It is convenient to think in terms of octaves of oversampling. With no oversampling (0 octaves of oversampling) we integrate to π. With one octave, we integrate to $\pi/2$, with two octaves, we integrate to $\pi/4$, and so on. From Fig. 3b, we see that the integral, when taken all the way to π, is sure to exceed the integration of 1 from 0 to π (dotted line corresponding to no noise shaping). However, when we integrate only to small fractions of π, we can clearly see that there will be far less power integrated, and we can win the game. Specifically:

$$p_m = \int_0^{\omega_m} (2 - 2\cos(\omega))\, d\omega \;=\; 2\omega_m - 2\sin(\omega_m) \tag{7}$$

All we have to do now is plug in values of $\omega_m = \pi/2^m$ where m is the number of octaves of oversampling. We then compare the results to π, the no noise shaping case. It is conventional to report this comparison in db of amplitude (voltage) rather than in power. Thus:

$$\text{dbs} = -20\ \text{Log}_{10}\left[\sqrt{p_m} / \sqrt{\pi}\right] \tag{8}$$

Some results are tabluated in Table 1

Table 1 Improvement in S/N as a result of m octaves of oversampling with first-order noise shaping, relative to the no oversampling, no noise-shaping case.

m	dbs	Hauser's Rule of Thumb	Orphanidis Formula
0	-3.0103	-6	-5.1541
1	4.3964	3	3.8459
2	13.0241	12	12.8459
3	21.9544	21	21.8459
4	30.9602	30	30.8459
5	39.9848	39	39.8459
6	49.0141	48	48.8459
7	58.0446	57	57.8459
8	67.0754	66	66.8459
9	76.1063	75	75.8459
10	85.1372	84	84.8459

Some additional data is offered in Table 1. Hauser [1] offers an excellent review paper on oversampling and noise shaping, and gives the useful rule of thumb for the number of bits that are effectively added due to noise shaping. He says that with first-order noise shaping you gain 1.5 bits per octave of oversampling, but must pay a one-bit "penalty." Converted to db (multiplying by 6db per added bit), we obtain the results shown in the table. A slightly better rule of thumb would be to show one additional db improvement to all values. None the less, the rule is appropriate when discussing added bits rather than db. Note that the largest estimation error occurs when m=0 (which is a loss anyway so would not be considered).

Orfanidis [2] gives a formula for the change of the number of bits that occurs with a k^{th} order noise shaper with m octaves of oversampling:

$$\Delta B = (k + 0.5)m - 0.5 \, \mathrm{Log}_2 [\, \pi^{2k} / (2k+1) \,] \tag{9}$$

For k=1, these results (again converted to db) are shown in Table 1. (In general, all the results are in excellent agreement.)

Orfanidis' formula is an approximation. What he does can be discussed in terms of equation (6) which we recognize as equal to 4 $\sin^2(\omega/2)$, a power of a sine function. If we had higher order noise shapers, we would have this factor raised to the k^{th} power. For small ω_m (large amount of oversampling), the sine can of course be approximated by its argument, and the integration of the power is thus much simpler to do.

The consequences of obtaining additional (effective) bits relate not just to a possible reduction in quantization noise, but also to the possiblilty of reducing the number of actual bits in an A/D or D/A converter and obtaining the same performance we had with a larger number of actual bits. Ultimately, we are able to get down to a one-bit conversion system, with some inherent simplicity and a guaranteed linearity. The conflicting claims of "more-bits-are-better" and "fewer-bits-are-better" is a source of confusion to many engineers, not to mention marketing copywriters, and the audio buying public.

References

[1] M. Hauser, "Principles of Oversampling A/D Conversion," J. Audio Eng. Soc., Vol. 39, No. 1/2, Jan/Feb 1991, pp 3-26

[2] S. Orfanidis, Introduction to Signal Processing, Prentice-Hall (1996)

[3] K. Pohlman, Principles of Digital Audio, Sams (1989)

[4] J.G. Proakis & D.G. Manolakis, Digital Signal Processing, Macmillan (1992)