Using Sigma-Delta Converters–Part 2

This is a continuation of a discussion of sigma-delta converters begun in the last issue. We covered antialiasing requirements, idle tones, and loading on the signal source.

Q: *What happens if my input signal is beyond the input range of the sigma-delta converter? I remember hearing something about the converter becoming unstable?*

A: The modulator can become temporarily unstable if it is driven with inputs outside the recommended range. However, this instability is invisible to the user, since decimators are designed to simply clip the digital output and show either negative or positive full scale, just as one would expect with a conventional converter.

Q: *The specifications for sigma-delta converters assume a certain input clock rate and therefore a specific sampling rate. Can I safely use the converter with a higher or lower clock frequency?*

A: While the specs are measured at a particular sampling frequency, we often specify a range of input clock frequencies that the device can be operated with. This translates into a range of possible sampling rates. If you plan to go much beyond that range you can expect some performance degradation. If you sample at higher rates than specified, the internal switched-capacitor circuits may not be able to settle to the required accuracy before a new clock edge comes along. With too slow a sampling rate, capacitor leakage will degrade performance.

The digital filter characteristics of the converter (group delay, cutoff frequency, etc.) scale with sampling rate; so too do the input impedance (unless the input is buffered) and power consumption.

Q: *I am planning to use a sigma-delta converter to digitize several signals by using a multiplexer at the input of the converter. Is that a problem?*

A: While sigma-delta converters have a certain appeal due to their ease of antialiasing, they do not lend themselves well to applications for multiplexed ac signals. The reason for this is that the output of a sigma-delta converter is a function not only of the latest analog input but also of previous inputs. This is mostly due to the memory that the digital filter has of previous inputs, but the modulator has some memory as well. In a multiplexing application, after switching from one input to another, all information the filter has about the old input needs to be flushed out before the converter output word represents the new input.

Most decimation filters in sigma-delta converters intended for ac applications are FIR filters, principally because of their linear phase-response. For FIR filters, it is easy to calculate the time it takes to rid the filter of any information about the old input. The figure shows the structure of a FIR filter; the number of clock cycles required to clock all old data points out (i.e., the filter settling time) is equal to k, the number of taps in the filter. While data corresponding to a new input is propagating through the filter and replacing the earlier data, the output of the filter is calculated from a combination of the old data and the new data. The AD1879, for example, an 18-bit audio A/D converter, has a 4096-tap FIR filter which, when running at 3.072 MHz, has a 1.33-ms settling time.

The effective sampling rate for sigma-delta converters in multiplexed applications is quite low because of this need to wait for the old signal to be flushed out before capturing a valid data point for the new input. Traditional converters, which convert directly, or in a small number of stages, are therefore a much better choice in applications requiring the capture of multiple ac channels.

$$y(n) = a_1 \, x(n) + a_2 \, x(n-1) + a_3 \, x(n-2) + \ldots + a_k \, x(n-k+1)$$
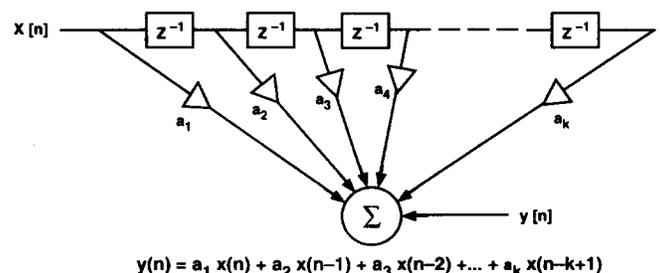
For a multichannel dc application where time is available to wait after switching between channels, or if the application does not require frequent changes between channels, the use of a sigma-delta converter can be very feasible. In fact, Analog Devices offers 16-24-bit converters with multiplexers on the input (AD771x family) specifically for such applications.

Q: *Does this also explain why sigma-delta converters are not suitable for some control applications?*

A: Yes. Since delays in control loops must be minimized for stability, sigma-delta converters are not suitable for control applications where they add a relatively long time delay. However, the actual delay is predictable; in applications that involve relatively slow signals, the converter phase delay, and therefore the effect on pole and zero locations of the control loop, may be negligible. However, even if this is the case, a traditional non-oversampling converter may still be a much better choice for the application, because a sigma-delta converter would need to run at a much faster sampling rate than a traditional converter in order to have the same phase delay. This will unnecessarily burden the circuitry that processes the A/D data.

Q: *Are there any other issues I should be aware of when using sigma-delta converters?*

A: In addition to the general guidelines on grounding, power supply bypassing, etc., that apply to all converters, there are a couple of points worth remembering when designing with sigma-delta converters. The first issue involves their input. As mentioned earlier, some sigma-delta converters (such as the AD1877) have buffers on the input; others (such as the AD1879), without a buffer, present a switched-capacitor load, which needs periodic current transients to charge the input capacitor. It is important that the circuitry driving the converter be as close to the converter as possible to minimize the inductance in the leads between the external circuitry and the switched-capacitor node. This reduces the settling time of the input and minimizes radiation from the input to other parts of the circuit board.

Another issue has to do with interference from clock signals affecting the A/D conversion. As I noted earlier, the digital decimation filter can't provide any filtering of signals whose frequencies are close to multiples of the modulator sampling rate. To be precise, the passbands are $[kF_{ms} \pm f_b]$s where k is an integer, F_{ms} is the modulator sampling rate, and f_b is the decimator cutoff frequency.

Besides the consequences for anti-aliasing discussed earlier, the decimator cutoff frequencies have a bearing on the selection of clock frequencies for devices that operate in the same system as the converter. These frequency bands (i.e., the passbands) embody the converter's greatest vulnerability to interference (inductive or capacitive coupling, power supply noise, etc.), because any signals in these frequency bands that manage to get into the modulator will not be subjected to attenuation in the filter. Therefore one is wise to avoid using clock frequencies that fall in these bands to minimize the possibility of interfering with the conversion—unless they are synchronous with the converter clock.

QUESTIONS ON NOISE IN CONVERTERS

Q: *I recently evaluated a dual-supply A/D converter; one of the tests I did was to ground the input and look at the output codes on a LED register. To my big surprise I got a range of output codes instead of a single code output as I expected?*

A: The cause is *circuit noise.* When the dc input is at the transition between two output codes, just a little circuit noise in even the finest dc converters will ensure that two codes will appear at the output. This is a fact of life in the converter world. In many instances, as in your case, the internal noise may be large enough to cause several output codes to appear. Consider, for example, a converter with peak-to-peak noise of just over 2 LSB. When the input of this converter is grounded, or a clean dc source is connected to the input, we will always see three—and sometimes even four—codes appear at the output. The circuit noise prevents the voltage being sampled from being confined to a voltage bin that corresponds to one digital code. Any external noise on the A/D input (including a noisy signal), on the power supplies, or on the control lines will add to the internal circuit noise—and possibly result in more bits toggling.

Q: *Is there a way to determine how many codes I can expect to appear when I apply a dc signal to a converter?*

A: It would not be hard in the ideal case where you knew the noise distribution, the exact size of the codes where the dc input is at and where within a code quantum the input lies (in the center, on the edge of two codes, etc.). But in reality you don't have this information. However, knowing some of the ac specifications (S/N, dynamic range, etc.) of the converter, you can make an *estimate.* From these specs you can find the magnitude of the rms converter noise relative to full scale. The noise will in all likelihood have a Gaussian amplitude distribution, so the standard deviation (sd) of the distribution equals the rms value. This also means that the codes that appear will not have equal probability of occurring. Using the fact that 99.7% of a Gaussian distribution occurs within ±3 standard deviations from the mean, we can estimate the peak-to-peak noise voltage at six times the standard deviation. If N_{rms} is the rms value of the converter noise and V_{LSB} is the size of the LSB in volts (= $V_{span}/2^b$) the peak to peak noise in terms of LSBs, N_B, is

$$N_B = \frac{6 \times N_{rms}}{V_{LSB}} = \frac{6 \times 2^b \times N_{rms}}{V_{span}}$$

In general, if the signal-to-noise ratio of a converter expresses noise power relative to full scale, we have

$$N_B = \frac{3}{\sqrt{2}} \times 2^b \times 10^{-SNR/20}$$

where b is the number of bits in the output word.

How many codes show up at the output depends where the mean of the input, i.e., the dc input value, is with respect to code transitions. If the mean is close to the boundary between two output codes, more codes are likely to appear than if the mean is half way between two output codes. It can easily be shown that N_C, the number of codes appearing for a particular value of N_B, is either INT(N_B)+1 or INT(N_B)+2, depending on the dc input value [INT(N_B) is the integer portion of N_B]. And don't be surprised to see even more codes from the less-probable noise amplitudes >±3 sd.

How many bits will N_C cause to toggle on the output? The number of bits needed to represent N_C codes is

$$\text{INT}\left(\frac{\log N_C}{\log 2} + 0.5\right)$$

We can, however, see many more bits toggle, since the number of bits toggling is a function of the actual value of the converter's dc input. Consider, for example, that a one-code transition from an output word of -1 to 0 on a 2s-complement-coded converter involves inverting all the output bits.

Lets look at an example using the AD1879, an 18-bit sigma-delta converter with dynamic range of 103 dB. From the definition of dynamic range we have

$$103 = 20 \log \frac{S}{N_{rms}}$$

From the AD1879 data sheet, we find that the rms value of a full-scale input signal, S, is $6/\sqrt{2}$ V rms. This allows us to solve for N_{rms} which turns out to be 30 μV. We next find the LSB size by dividing the full input range by the number of possible output codes:

$$V_{LSB} = \frac{12}{2^{18}} = 45.8\,\mu\text{V}$$

Thus N_B is 3.9. We can therefore expect either 4 or 5 different codes to appear at the AD1879 output when the input is grounded (ground corresponds to a midscale input for the AD1879).
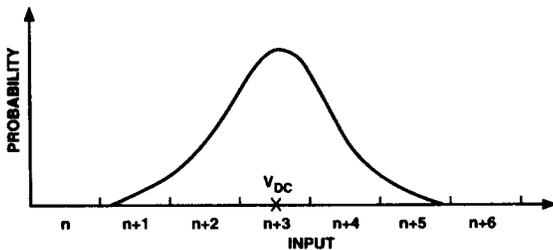

One can take this estimation one step further: If the standard deviation (the rms value) of a Gaussian distribution and the mean (the mean of the noise is 0 in this case) are known, one can use standard tables for the Gaussian distribution to calculate what percent of the time the noise will fall into a voltage interval corresponding to a specific output code. A histogram can be estimated, showing the distribution of codes at the output. Also the process can be reversed: a histogram showing the distribution of noise codes at a given value of dc output permits one to estimate the S/N ratio for a converter.

To make all this real, let's continue our example involving the AD1879. Consider two cases, one where the input lies midway between two output codes and one when the input is on the transition between two codes. From the calculations above, we found that the standard deviation (sd) of the noise (the rms value) was 30 μV. The size of one LSB in terms of sd is

$$\frac{45.78\,\mu\text{V}}{30.0\,\mu\text{V}} = 1.524$$

In the case where the dc input is midway between code transitions, as shown below, it is clear that any noise that falls

within -0.5 LSBs to $+0.5$ LSBs from the input will result in the correct code at the A/D output. This corresponds to the noise being confined to a range of (-0.5×1.524) sd to $(+0.5 \times 1.524)$ sd from its mean (0). From standard tables one can find that the noise will fall in this range 55.4% of the time. If the noise falls within 0.5 LSBs to 1.5 LSBs, the output will be one code too high. Again from standard tables one can find that this will occur 21.2% of the time. Continuing in this manner one can calculate the whole histogram showing the distribution of output codes.
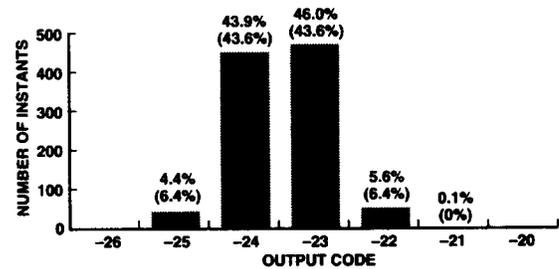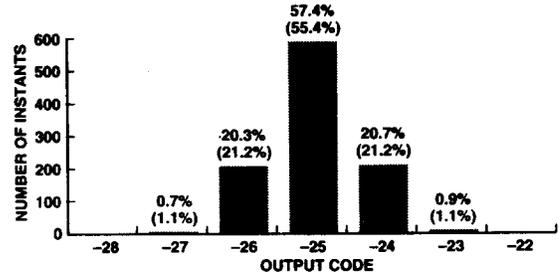
The upper figure shows an actual measurement where the dc input happened to be -25 LSBs. Five output codes, ranging from -27 to -23, appeared. 1024 measurements were taken and the percentage distribution of each code is shown on top of each column. The calculated distribution is listed in brackets on top of each column. As can be seen, the experimental results agree well with the calculated values. The lower figure shows a case where the dc input is close to the boundary between two codes. By following a similar procedure, one can calculate how the histogram should look. Again the experimental and calculated values are in excellent agreement. Note that the actual applied dc input is slightly above the border between the two codes, whereas the calculations assume it is exactly on the border.

The biggest weakness of this estimating technique is the fact that in conventional converters the code width (the amount the dc input has to be increased to increase the digital output by one bit) varies from code to code. This means that if the dc input is in an area where codes are narrow, we can expect more bits to be toggling than in an area where the codes are wide. This method also assumes that the circuit noise within the converter stays constant, whether the applied signal is ac or dc. This is not exactly true in many cases.

The estimate will probably be more accurate when used with sigma-delta converters (except for "dead bands"), because neither of the two factors mentioned above is an issue in such converters.

6

Q: *Ah, now I understand why there are multiple codes at the output. But why not discard the bits that toggle and only bring out the bits that stay steady, since the others are really indeterminate? Isn't that the real resolution of the converter?*

A: Many converters are designed for ac or dynamic applications where THD (total harmonic distortion) and THD+N (total harmonic distortion+noise) are the most important specs. The design therefore focuses on minimizing harmonic distortion for high- *and* low-level input signals, while keeping the noise to acceptable levels. As it turns out, these requirements somewhat contradict the requirements for a good dc converter, which is optimized for precision conversion of slow moving signals where harmonic distortion is not an issue. It is actually desirable to have some noise (called dither) superimposed on the input signal to minimize distortion at very low input signal levels; dither can also be used to improve dc accuracy where repeated measurements can be made.

To understand how this may be, let's start by looking at quantization noise. The output of an ideal A/D converter has finite accuracy because of the finite number of bits available to represent the input voltage. Each one of the 2^b quanta represents with one single value all values in the analog range from –0.5 LSB to +0.5 LSB of its nominal input value. The A/D output can therefore be thought of as a discrete version of the analog input plus an error signal (quantization noise). When a large and varying input signal (dozens, hundreds, or thousands of LSBs in amplitude) is applied to a converter, the quantization noise has very little correlation with the input signal. It is, in other words, approximately white noise. The figure shows the quantization noise of a perfect A/D converter at various instants of time when the input signal is a sinusoid of about 100 LSBs in amplitude.
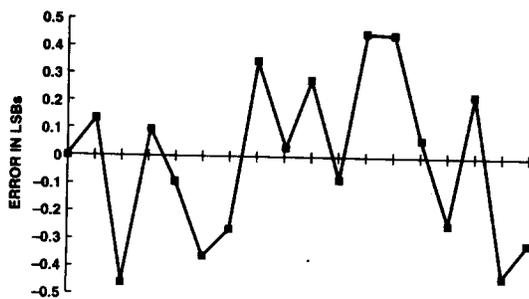
When the A/D input is very low in amplitude, so that the amplitude does not change more than a fraction of a LSB between samples, the samples stay in the same quantum, and are therefore constant for a few sample periods. This is depicted in the figure below, which shows a sinusoidal input signal that has an amplitude of only 1.5 LSBs, the A/D output and the quantization noise. Note that the quantization error follows the input waveform exactly while the samples are staying constant. The longer the samples stay constant, the more the quantization noise looks like the input waveform, i.e., the correlation between the input signal and the quantization noise increases. While the rms of the quantization error may not have changed, the quantization error will take on a non-uniform spectral shape. In fact, the correlated quantization noise shows up as harmonics in the A/D spectrum.
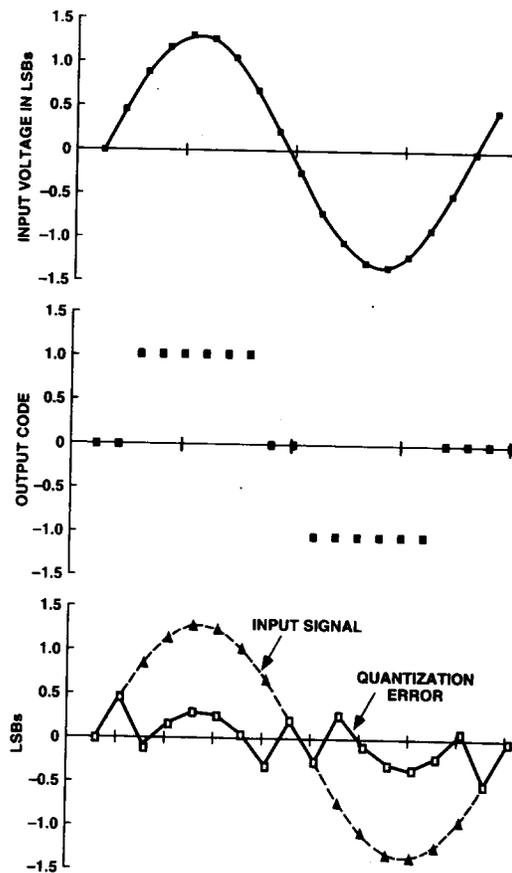
Another way to look at this phenomenon is to consider the case when the (sinusoidal) input signal is only around 1 LSB in size and the digital output resembles a square wave. Square waves are rich in harmonics! The harmonics, or noise modulation products, are very objectionable in many converter applications, especially audio.

To get around this problem, a technique called dithering is used to trade correlated quantization noise for white noise, which is less offensive to the human ear than correlated noise. Dithering is done by using circuit elements to *add* random noise to the input signal. While this will result in an increase of the total converter noise, the added noise breaks up the simple square wave patterns in the output code. The quantization error will not be a function of the input signal but of the instantaneous value of the dither noise. Thus the dither decorrelates the quantization noise and the input signal. The size of the dither signal is often about 1/3 LSB rms (2 LSBs peak-to-peak if the noise is Gaussian). Clearly, this will result in a converter that will have more than two codes at the output when the input is grounded. We saw an example earlier involving the AD1879 which had either four or five codes appear on the output depending on the dc input level.

The figure below shows the simulated output of an A/D converter with an undithered low level input signal. The quantization noise is a function of the input signal magnitude at the sample instant. This correlation between the quantization noise and the input signal shows up as a cluster of harmonically related sticks in the A/D output spectrum. Note that the magnitude scale in the figure is referenced to the input signal (not full scale input).
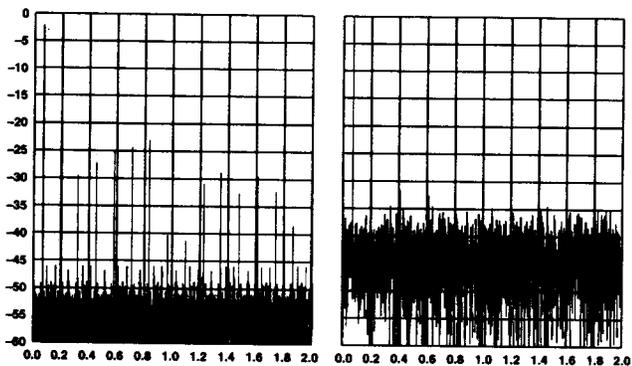
The right-hand figure shows the A/D output after a dither signal that is 4 dB above the quantization noise floor is added to the input. In this case the quantization noise depends on the magnitude of the dither signal at the instant when a sample is taken. Since the value of the dither doesn't depend on the input signal, the quantization noise becomes uncorrelated to the input and the harmonics in the A/D spectrum are eliminated, but at the cost of an overall increase in the noise floor.

Instead of actually adding noise to the A/D input, dithering can be accomplished by using the thermal noise of the converter as the dither signal and calculating enough output bits to ensure a decorrelated quantization noise.

Though I have used A/D converters in my examples, the idea of using dither also applies to D/A converters as well. Dither is applied to D/A converters by adding the output of a digital noise generator to the digital word sent to the D/A.

Q: But in dc applications, I want to make an accurate measurement each time and may not be able to tolerate the uncertainty of having a few LSBs of error in a particular measurement.

A: If you need *n*-bit dc accuracy in each conversion and you have problems finding a suitable *n*-bit converter, you have two options. One is to use an (*n*+2)-bit converter and simply ignore the two LSBs. However, if your hardware has the capability (and time) to do some signal processing, you can enhance the resolution of a noisy (dithered) dc converter and, in fact, get more than *n*-bit accuracy out of an *n*-bit converter if the accuracy is limited by noise.

To understand why this may be so, think of an ideal *n*-bit converter. For a particular value of dc input, you will get one digital code at the output. However, you do not know where the input lies within the code quantum (i.e., in the middle,

close to the upper transition, etc.). That may be sufficiently accurate for your application, but if you add noise to the input of the converter—so that several codes can appear at the output—you will find that the code distribution contains information to place the dc value of the input more exactly.

In the earlier examples involving the AD1879, we saw how the code distribution looks when the input is in the vicinity of a code transition; the two most-frequent output codes are the ones on either side of the transition. Their average is therefore a good estimate of where the input lies. In fact, taking the average of a lot of conversions, while the input stays put, is an excellent way of enhancing the resolution of the converter. One has to be careful, when processing the converter output, to allow the output word length to grow without introducing roundoff errors. Otherwise one actually injects unwanted noise—called *requantization noise*—into the final output. Note that filtering out the noise is only just that; it will have no effect on other error sources of the converter, such as integral and differential nonlinearity.

This concept of resolution enhancement is an interesting one and is not restricted to the dc domain. One can actually trade resolution for bandwidth in the ac domain and combine the outputs of several converters or to construct a more-accurate output. The basic principle is that signal repetitions (which are self-correlated) add linearly, while repetitions of random noise produce root-square increases. Thus, a fourfold increase in number of samples increases S/N by 6 dB. Perhaps we can discuss useful applications of this principle in these pages in the future.

Q: You mentioned a couple of converter ac specifications above. I am somewhat confused about how S/N, THD+N, THD, S/THD, S/THD+N, and dynamic range are measured on A/D and D/A converters and how they relate to each other. Can you shed any light on this?

A: Your confusion is quite understandable. There is unfortunately no industry standard on exactly how these quantities are measured and therefore, what exactly they mean. Sometimes manufacturers are guilty of choosing the definition that portrays their part favorably.

Most often data sheets include a note on the testing conditions and how the different specs were calculated. The best advice I can give is to read these very carefully. By simple calculations you can often convert a specification for one part to a number that allows a fair comparison to a specification for another part.

Most specifications are not expressed in absolute units, but as relative measurements or ratios. Noise, for example, is not specified in rms volts, but as *SNR*, or the ratio between signal power and noise power under particular test conditions. These ratios are usually expressed in decibels, dB, and occasionally as percentages (%). A power ratio, x, expressed in bels, is defined as $\log_{10}x$; multiply by 10 if expressed in decibels (one tenth of a bel): $10 \log_{10}x$. *SNR* is therefore equal to $10 \log_{10}$ (signal power/noise power) dB. Evaluated in terms of rms voltage quantities, $SNR = 20 \log_{10}(V_{signal}/V_{noise})$.

6

Armed with this knowledge, let's see whether we can make sense out of the multiple specifications you mentioned above (many of which are redundant). Those specifications seek to describe how the imperfections of the converter affect the characteristics of an ac signal that gets processed by the converter. For dc applications, a listing of the magnitude of the actual imperfections suffices, but these can only suggest ac performance. For example, integral nonlinearity is a major factor in determining large-signal distortion (along with glitch energy for D/As) while differential nonlinearity governs small-signal distortion. To accurately determine the ac performance, at least two types of tests are performed in the case of A/Ds. The tests are as follows:

Full-scale sine (a)

A sinusoidal signal approaching full-scale is applied to the converter. The signal is large enough so that converter's imperfections cause significant harmonic components to occur at multiples of the input signal frequency. The harmonics will show up in the output spectrum, along with noise. A common performance measure is the relative magnitude of the harmonic components, usually expressed in dB. Relative to what? Two possibilities are the applied input signal and the full scale of the converter (which in most cases is different from the applied input signal). Referring the harmonics to full scale will clearly yield a lower (more attractive) number than referring them to the rms value of the actual input signal. This reference issue causes a lot of confusion when dynamic specifications are evaluated, because there is no universally accepted standard for what each performance measure should be referred to. The best advice I can give you is: never assume anything; read manufacturers' data sheets very carefully.

Sometimes the magnitudes of the individual harmonics are specified, but most often only the total harmonic distortion (THD) is specified. The THD measures the total power of the harmonics and is found by adding the individual harmonics in rss fashion. The formula then for THD when referred to the input signal is

$$20 \log_{10}\left[\frac{\sqrt{\sum_{i=2}^{m} H^2(i)_{rms}}}{S}\right] \text{ or } 10 \log_{10}\left[\frac{\sum_{i=2}^{m} H^2(i)_{rms}}{S^2}\right]$$

where $H(i)_{rms}$ refers to the rms value of ith harmonic component and S to the rms value of the input signal. Usually, harmonics 2 through 5 are sufficient. Note that the input-frequency, or *fundamental*, component is the first harmonic. To refer any harmonic to full scale, add x dB to the formula above, where x is the magnitude of the input signal relative to full scale. This simple conversion formula can be applied to other specifications, but take care to observe proper polarity of the log quantities.

Nowadays, clear distinction is usually made between total harmonic distortion plus noise (*THD+N*) and *THD*. This has not always been the case. *THD+N* includes not only the harmonics that are generated in the conversion, but also the noise. The formula for *THD+N* when referred to the input signal is:

$$20 \log_{10}\left[\frac{\sqrt{N^2_{rms} + \sum_{i=2}^{m} H^2(i)_{rms}}}{S}\right]$$

or

$$10 \log_{10}\left[\frac{N^2_{rms} + \sum_{i=2}^{m} H^2(i)_{rms}}{S^2}\right]$$

where N_{rms} is the rms value of the integrated noise in the bandwidth specified for the measurement.

Another commonly used specification is signal to noise-plus-distortion (*S/[N+D]*, or *S/[THD+N]*), also called *sinad*. This is essentially the inverse of *THD+N*, when referred to the signal; its dB number is the same, but with opposite polarity.

Another performance measure describing the test results is the signal to noise ratio, *S/N* or *SNR*, which is a measure of the relative noise power, most useful for estimating response to small signals in the absence of harmonics. If *S/N* is not specified, but *THD* and *THD+N* are provided, relative to the input signal, *THD* can be rss-subtracted from *THD+N* to obtain the noise to signal ratio [= $1/(S/N)$]. If the numbers are given in dB, the rss subtraction formula for logarithmic quantities in the Appendix can be used as follows

$$SNR = -10 \log_{10}\left(10^{(THD+N)/10} - 10^{THD/10}\right)$$

to yield the input signal power relative to noise power expressed in dB.

Low-level sine (b)

The second test usually performed is to apply a sinusoidal signal well below full scale to the converter (usually –60 dB). At this input level, sigma-delta converters usually exhibit negligible nonlinearities, so only noise (no harmonic components) appears in the spectrum. At this level, $S/N = S/N+D = -THD+N = -THD$, when all are referred to the same level. As a result, one specification indicating the noise level suffices to describe the result of this test. This specification called *dynamic range* (inversely, dynamic-range *distortion*), specifies the magnitude of the integrated noise (and harmonics if they exist) over a specific bandwidth relative to full scale, when a –60-dB input signal is applied to the converter.

Conventional (i.e. not sigma-delta) converters can exhibit harmonics in their output spectrum even with low-level input signals because all the codes may not have equal width (differential nonlinearity). In some such instances, the *S/N*, which ignores harmonics, measured with a –60-dB input signal, is different from dynamic range.

Frequently one sees *THD+N* at –60-dB and dynamic range specified for the same converter. These really are, as explained above, redundant since they only differ in the reference used. The only twist on dynamic range is that sometimes, when audio converters are specified, a filter that mimics the frequency response of the human ear is applied to the converter output. This processing of the converter output is called A-weighting (because an A-weighting filter is used); it will effectively decrease the noise floor, and therefore increase the signal-to-noise ratio, if the noise is white.

Everything discussed above applies to both A/D and D/A converters, with the possible exception of signal to noise ratio. Sometimes (particularly for audio D/A converters) *S/N* is a measure of how "quiet" the D/A output is when zero (midscale) code is sent to the converter. Under these conditions, the S/N expresses the analog noise power at the D/A output relative to full scale output.

It's important to note that the performance measures above are affected by: *bandwidth* of the measurement, the *sampling frequency*, and the input *signal frequency*. For a fair comparison of two converters, one has to make sure that these test conditions are similar for both.

Another Question

Q: I intend to use Analog's AD1800 family of audio D/A converters for a digital audio playback application. I understand that using an interpolator ahead of the D/A will make it easier to filter the D/A output, assuming I want to get rid of all the images at the D/A output. But is it really necessary to filter the output, since all the images will be above the audible range as long as sampling is at >40 kHz.

A: Good question. The audio equipment (audio amplifiers, equalizers, power amplifiers, etc.) that may eventually receive the output of your D/As are typically built to handle 20-Hz to 20-kHz signals. Since they are not intended to respond at frequencies much beyond 20 kHz—and in effect themselves function as filters—they may not have the necessary slew rate and gain to handle incoming signals from an unfiltered D/A output having significant energy well above 20 kHz. With their slew-rate and gain limitations, the amplifiers are driven into nonlinear regions, generating distortion. These distortion products are not limited to high frequencies but can affect the 20-Hz to 20-kHz range as well. Attenuating the high frequency signals at the DAC will therefore reduce the possibility of distortion. CD players often include filters steep enough to reduce the total out-of-band energy to >80 dB below full scale.

APPENDIX

RSS addition of logarithmic quantities: The root-square sum of two rms signals, S_1 and S_2, has an rms value of $\sqrt{S_1^2 + S_2^2}$. One often needs to calculate the rss sum of two numbers that are expressed in dB relative a given reference. To do this one has to take the antilogs, perform the rss addition, then convert the result back to dB. These three operations can be combined into one convenient formula: If D_1 and D_2 are ratios expressed in dB, their sum, expressed in dB, is

$$10 \log_{10}\left(10^{D_1/10} + 10^{D_2/10}\right)$$

Similarly, to find the difference between two rms quantities,

$$x = \sqrt{S_2^2 - S_1^2}$$

the result, x, expressed in dB, is

$$10 \log_{10}\left(10^{D_2/10} - 10^{D_1/10}\right)$$

References:
[1] *Oversampling Delta-Sigma Data Converters—Theory, Design, and Simulation*, edited by J.C. Candy and G.C. Temes, IEEE Press, Piscataway, NJ, 1991.
[2] J. Vanderkooy and S.P. Lipshitz, "Resolution Below the Least Significant Bit in Digital Systems with Dither," *J. Audio Eng. Soc.*, vol. 32, pp. 106-113 (1984 Mar.); correction ibid., p.889 (1984 Nov.).
[3] A.H. Bowker and G.J. Lieberman, *Engineering Statistics*, Prentice Hall, Englewood Cliffs, NJ, 1972.