The Filter Wizard
issue 30: Why Are My DAC and ADC Responses Drooping?
Kendall Castor-Perry

Have you ever experienced unexpected droop when you've plotted out the frequency response of a signal processing system? When you expected that the frequency response would be flat (or at least accurate to the curve you designed to), but instead it rolls smoothly and lazily away from that target value, insulting you with its casual sogginess? If so, you've experienced the consequence of having a sinc() frequency response. You could say that you've had a clo-sinc-ounter – though possibly of a kind other than the third. Let's look at what I mean by that.

This issue can crop up both at the input to and the output from a sampled data system. Let's look at the output first. When you want to turn a stream of sample values back into an analogue system, you apply those digital samples to a DAC. Now, most DAC ICs and modules have a 'held' output. That means that when they receive a new digital sample the output voltage changes promptly to the corresponding new value – and stays there, until the next sample comes along. This behaviour is so commonplace that many engineers assume that it's the norm and that the output voltage of such DACs somehow represents the sample stream correctly (apart from a bit of pesky high frequency noise).

This is not true. This "hold" process causes the frequency response of such a system to differ from that of a system where the output voltage is only asserted very briefly at each sample instant. Such a spiky output voltage is hardly ever convenient in a real world application, which is why you rarely encounter it.

Stretching each sample's voltage out to "fill the space available" is an example of a zero order hold. The output frequency spectrum of such a system is equal to that of an ideal, spiky-output system multiplied by the spectrum of the rectangular impulse that fits between two sample points, i.e. has a width equal to the sample interval. Such a rectangular time response corresponds to a frequency response that has a sinc() characteristic. sinc(x) is shorthand for sin(x)/x, and there's a Fourier looking-glass correspondence between rectangular in one domain and sinc() in another that crops up all over the place, not only in signal theory but in the whole of physics.

Calculating the value of the sinc function – the value of the argument x is pi times the ratio of signal frequency to sampling frequency – shows that the droop is already -3 dB at around 0.444 times Fs. Figure 1 shows the frequency effect of sinc() droop for a 1 per second sample rate. Notice that it has deep but narrow notches at frequencies that are multiples of the sample rate.

Figure 2 shows a 0.444 Hz sinewave and also the result of sampling it once per second. The peak value that is reached by a sample can clearly be the peak value of the input voltage. But as the sample clock 'walks' over the signal there are regions where the output voltage is low for an appreciable period of time.

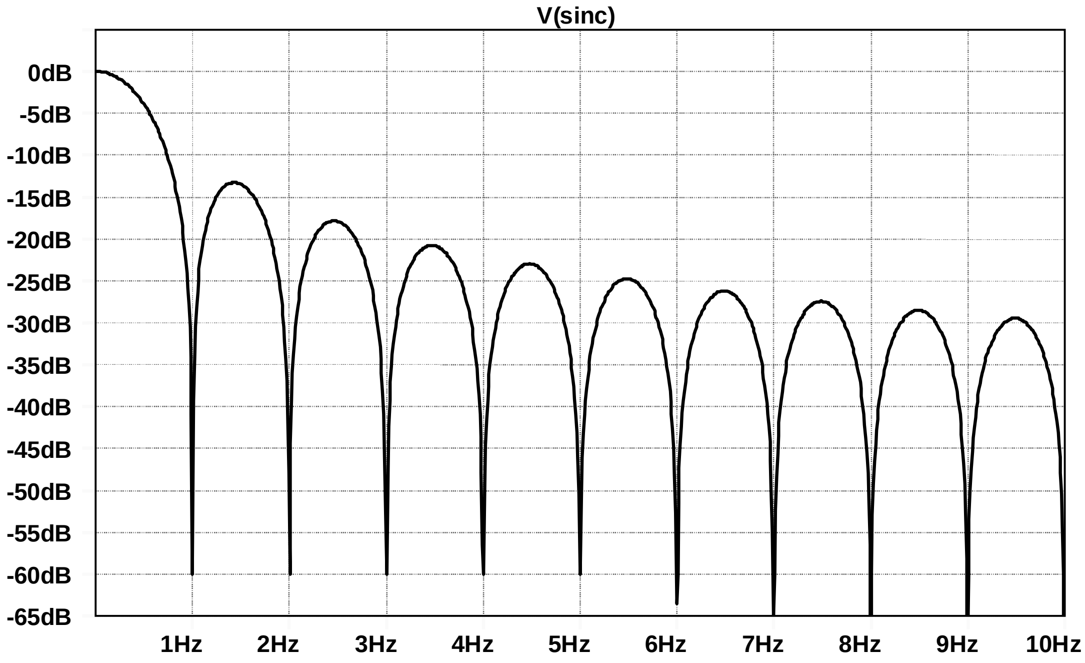

Figure 1: The sinc() response of zero-order hold at 1 sample per second.

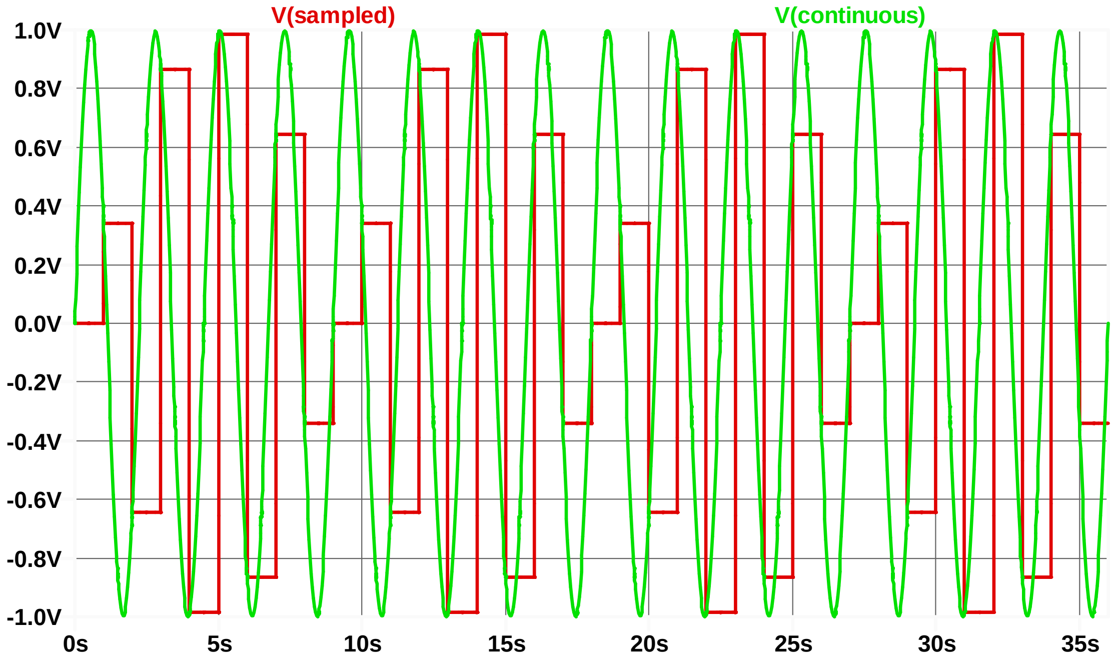

Figure 2: A 0.444Hz sinewave sampled – and held – at 1 sample per second.

Our 0.444 Hz signal is still present, but its level has been reduced, since some of the energy has been moved into higher frequency ‘images’, as shown in figure 3. You can see that the input signal’s single Fourier component peeps up 3 dB above the value of the 0.444 Hz component in the output signal.

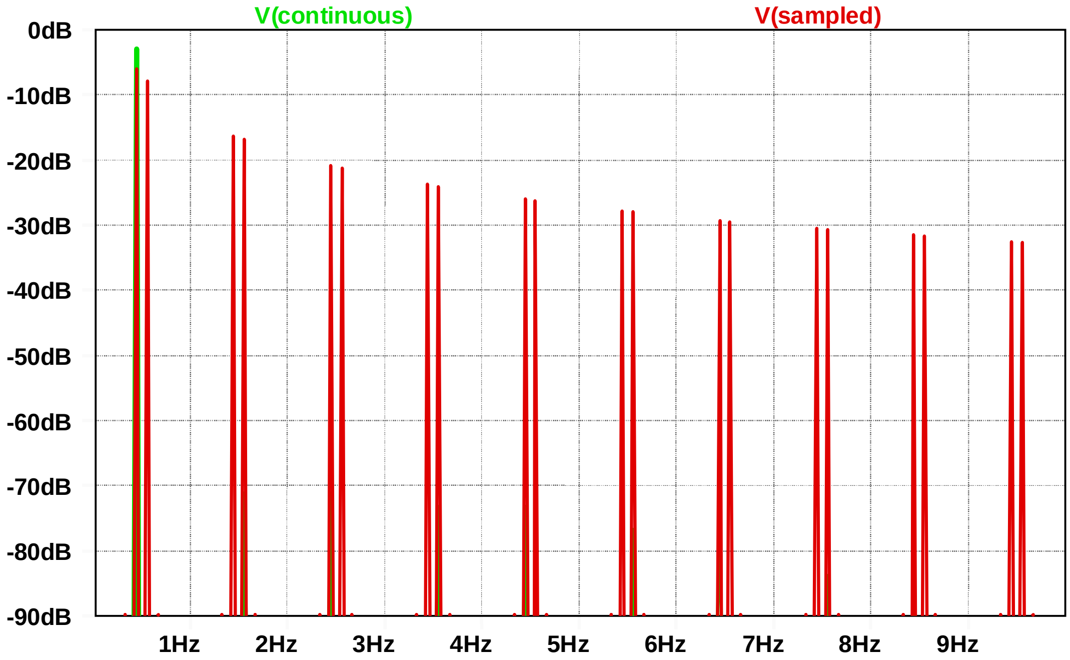

Figure 3: Frequency spectra of the waveforms in figure 2.

The take-away is that the peak-to-peak value of a sampled sinewave is not a good measure of the energy contained at the fundamental frequency. As you increase the input frequency, progressively more and more energy in the output signal resides in those higher frequency image components instead – the muck that we usually try to filter out to get a nice clean output signal. Hence the droop.

Note that modern DACs designed for the audio market don't exhibit this problem. That's because they are not just updating once per sample and holding the signal. Deep in the bowels of the converter they are running much faster, and then applying digital filter techniques to produce an output that magically does not appear to have been sampled at all on casual inspection. The fact that it's so easy to get an audio DAC with a super-flat frequency response also makes it easy for engineers to forget that "old school" sampling DACs don't have this flat response property.

That's the "why" for the output part of our system. Can the input path also cause droop that might be detectable when we analyze the data in the digital domain without even going back to analogue? Well, yes, sometimes it can; let's look at the circumstances under which it does.

If you are using a sampling ADC, the answer is generally "don't worry". Such an ADC takes a snapshot of the input signal over a brief "aperture" of time. This aperture is usually far narrower than the time between samples, so it has insignificant effect on the frequency response. However, if you are using the kind of delta-sigma ADC that is targeted at industrial instrumentation applications, you'll probably get much more droop than you bargained for. (Do people actually bargain for droop? I suppose it's just a figure of speech.)

The reason why delsig (to use the industry vernacular) ADCs have a droopy frequency response is that the averaging filters, used to smooth out the fast pulse streams from their front-end "modulators", have analogous impulse responses to that of the zero-order hold we just looked at. In fact, usually the droop in response is two to four times worse at any given frequency. This is because the filters used are usually, effectively at least, the cascade of two to four averaging filters. In the case of the lovely delsig ADC in Cypress's PSoC3 and PSoC5 devices (OK, so I'm biased) the ADC's decimation filter has four stages (over most of its range) and therefore has a sinc^4() response. At any given signal frequency it therefore has four times (expressed in dB), the droop shown in figure 1. In other words it's -12 dB at 0.443 times Fs. Such a gross departure from flatness of frequency response is not of consequence when what you really want is just the weight of bananas on a scale. But for most audio, communications and vibration measurement systems, it's just plain terrible. And that's before you add on the extra droop you'll get if you feed the signal into a DAC.

There's a good side to this response effect though. Figure 1 showed that the sinc() response "bounces back up" to only about -13.3 dB at about 1.43 times Fs. This is a reminder that the simple averager is not a very good filter for getting rid of high frequency variations. But if you put four of them in series to set sinc^4(), you now have a stopband response that only bounces back to around -53 dB, shown in figure 4. That's quite useful filtering, usually more than enough for precision measurements in the time domain where there's not too much high frequency interference.

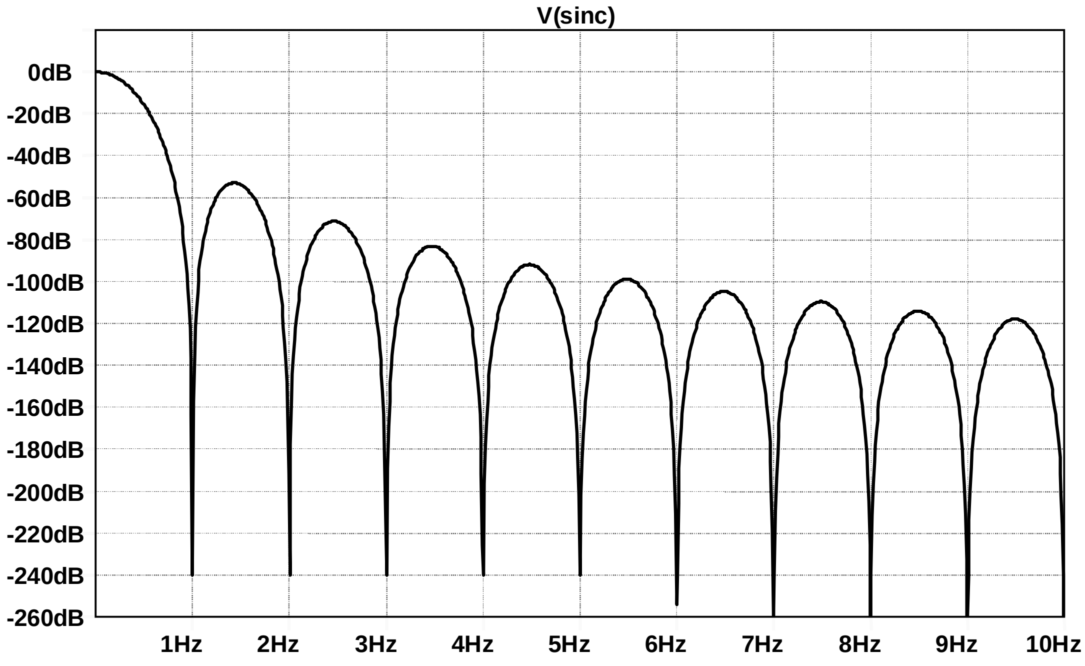

Figure 4: The sinc^4() response of the PSoC3 ADC's decimation filter.

So what can you do when you need a flat frequency response and you really have to use a "held" DAC, an instrumentation delsig ADC or both? You should expect by now that a solution proposed in this column will involve a filter – and I shall not disappoint you! Except in the sense that I'm not going to describe some nice methods of designing such filters until the second part of this article. Until then, keep sampling, and I hope that you don't get too much of that sincing feeling... best / Kendall