The Filter Wizard
issue 37: "Perfect" Pseudo-Differential Input ADCs

Kendall Castor-Perry

In this column, the Filter Wizard discusses a practical application of the time realignment filtering technique described in an earlier article.

One of my colleagues asked me to help him understand 'pseudo-differential' inputs, as particularly applied to ADCs that he had read about on the web. Now, I'm always suspicious when I see the prefix 'pseudo'. It's a woolly, imprecise term – a pseudo-prefix, you might even say. Strictly speaking, it means "false, fraudulent, or pretending to be something it is not".

My first thought was of multiplexed differential inputs where each input signal is measured relative to the same reference signal. In other words, the inverting input of the ADC is constantly connected to this reference, while a single-ended multiplexer feeding the non-inverting input cranks round a set of input signals. Turns out that's not what my colleague was talking about. I have a feeling that such inputs are called quasi-differential – quasi, of course, being another of those pseudo-prefixes, this time meaning "almost, having some resemblance to". But the great Richard Feynman was always clear that knowing what people call something tells you next to nothing about what it actually does.

The type of 'pseudo' differential input we're talking about here is where you take a sample of the voltage on one input connection that you think of as the non-inverting input, then a sample from another that you consider to be the inverting input, and then subtract one reading from the other. This gives you an indirect measurement of the difference in voltage between the two inputs. Given only a single-ended measurement channel, this allows you to 'fake' a differential input looking at a floating voltage.

If your signals are static and you can rely on the signal that you're not looking at not moving while your metaphorical back is turned, this can work reasonably well. Pretty obviously, though, if either of those signals are changing with time, you're in trouble. Consider the case in which the two signals are in fact identical. If this input signal is varying with time – let's say that it contains some AC line voltage hum – then the output word from the ADC will change on each sample, and the calculated difference between successive sample pairs will be non-zero even though you know jolly well that it should be zero, because the two signals are the same.

In differential input terms, the pseudo-differential ADC you've made has deteriorating common mode rejection at the input, as the signal frequency rises. For a given frequency, things improve as you sample at a higher rate, because the input signal doesn't move as much from sample to sample. But it's quite possibly not enough to make the technique workable in the presence of significant AC common mode signal. So, it sounds like this is a bit of a rubbish solution to the high-performance differential signal monitoring problem. Unless you're prepared to use a little Filter Wizardry, that is!

We can make use of the 'time realignment' filtering process that I first mentioned in Sample Multiple Channels 'Simultaneously' With A Single ADC. We'll see how this can help you make not only a high performance pseudo-differential input ADC, but even a multi-channel quasi-pseudo-differential system!

Let's recap the technique. The process starts with the selection of a lowpass FIR filter response that's suitable for use as a decimation filter to reduce the sample rate by a factor equal to the number of channels N being processed. This requires the filter to have a stopband beginning at $F_s/2N$, with F_s being the sample rate that the ADC is taking data at. The passband behaviour of our final output channel is going to be determined by the passband properties of this filter, while the attenuation in the stopband is going to determine the ultimate common mode rejection performance. The tap count is chosen to be divisible by N.

Since this starting filter's stopband is positioned to permit subsampling at the output by a factor of N, it follows that any of the N ways of taking every Nth sample is a legitimate decimation-by-N of the filtered signal. So this one filter can be "peeled apart", sequentially assigning its coefficients to the N subfilters. Each of these subfilters is then used to process the corresponding channel at the F_s/N rate.

An important consequence is that all N filters can be executed in one go after one sample from each of the N channels has been acquired, and all the filtered outputs are then available simultaneously, to do time-aligned cross-channel calculations.

If successive samples of the same signal are applied to the input of every subfilter – in other words, if the ADC input is not multiplexed but is just converting a test signal – the variation in group delay through each of these different subfilters exactly compensates for the variation in signal delay caused by the channel's position in the sampling sequence. If all of the outputs are viewed simultaneously on a scope, there will be no delay between the representations of that signal at the output. The original article has some nice example plots of this.

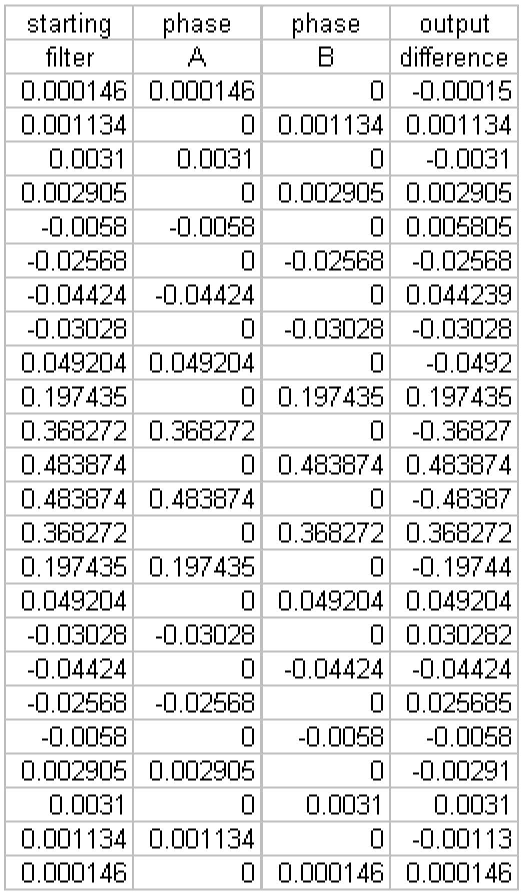
Provided the initial filter stopband attenuation is selected to ensure suitably low levels of aliasing, the passband frequency responses of the subfilters will be almost identical. This means that the difference signal obtained by subtracting two channels will be very low not only at DC, but for AC signals over the entire operating frequency range of the system. If a disturbing signal is present either on a common external reference potential or in the processing circuits on the ADC front end, it will be rejected by a predictable amount that is only dependent on the digital filter coefficients and is therefore quite unaffected by component tolerance or environmental factors. Meanwhile, any signal component that's unique to only one of the selected inputs will be represented as expected in that channel's data stream.

What we've done is create common mode rejection – suppression of a signal that's common to two or more inputs – using purely digital techniques. The stopband rejection of the filter (and therefore the common mode rejection) can, in principle, be increased without limit, and no analogue matching requirements are necessary.

There's another almost identical configuration that can be improved in this way: correlated double sampling, or CDS. The process is exactly the same; take one sample that consists of a signal plus an error, and then take a second sample that contains only the error. In some systems this error might be the input offset or low frequency noise from an analogue front end. Subtracting two successive readings is a common technique for removing DC offset but, as we've deduced and will soon see, it doesn't do a good job of removing higher frequency noise. If the error signal is AC rather than DC, using time alignment filters on the data streams can really help to suppress it.

Let's look at some actual examples. No fancy design tools are needed to use this technique. The PSoC Creator Filter customizer was use to create some initial filters and present the results; a spreadsheet was used to speed up manipulation of the coefficients.

First, a two-channel system. Let's take a 24-tap Blackman filter to start with, with cutoff at 0.125 F_s and a passband gain of 2 (because half the energy is going to go into each subfilter – does that make sense?).



starting	phase	phase	output
filter	A	B	difference
0.000146	0.000146	0	-0.00015
0.001134	0	0.001134	0.001134
0.0031	0.0031	0	-0.0031
0.002905	0	0.002905	0.002905
-0.0058	-0.0058	0	0.005805
-0.02568	0	-0.02568	-0.02568
-0.04424	-0.04424	0	0.044239
-0.03028	0	-0.03028	-0.03028
0.049204	0.049204	0	-0.0492
0.197435	0	0.197435	0.197435
0.368272	0.368272	0	-0.36827
0.483874	0	0.483874	0.483874
0.483874	0.483874	0	-0.48387
0.368272	0	0.368272	0.368272
0.197435	0.197435	0	-0.19744
0.049204	0	0.049204	0.049204
-0.03028	-0.03028	0	0.030282
-0.04424	0	-0.04424	-0.04424
-0.02568	-0.02568	0	0.025685
-0.0058	0	-0.0058	-0.0058
0.002905	0.002905	0	-0.00291
0.0031	0	0.0031	0.0031
0.001134	0.001134	0	-0.00113
0.000146	0	0.000146	0.000146

Table 1: Coefficient development for two-channel system.

Stopband attenuation is better than 82 dB down, relative to DC, from $F_s/4$ upwards. The response, plotted from the coefficients in the first column of table 1, looks like figure 1. If we take every other coefficient (columns 2 or 3 of table 1) and plot the response for the same sample rate, we get figure 2 (the amplitude response is essentially identical for either half, though the phase and delay plots will look different). The response comes back up again because we have effectively halved the sample rate by leaving out alternate coefficients. Apart from the 6 dB gain difference, the left hand halves of figures 1 and 2 are essentially identical.

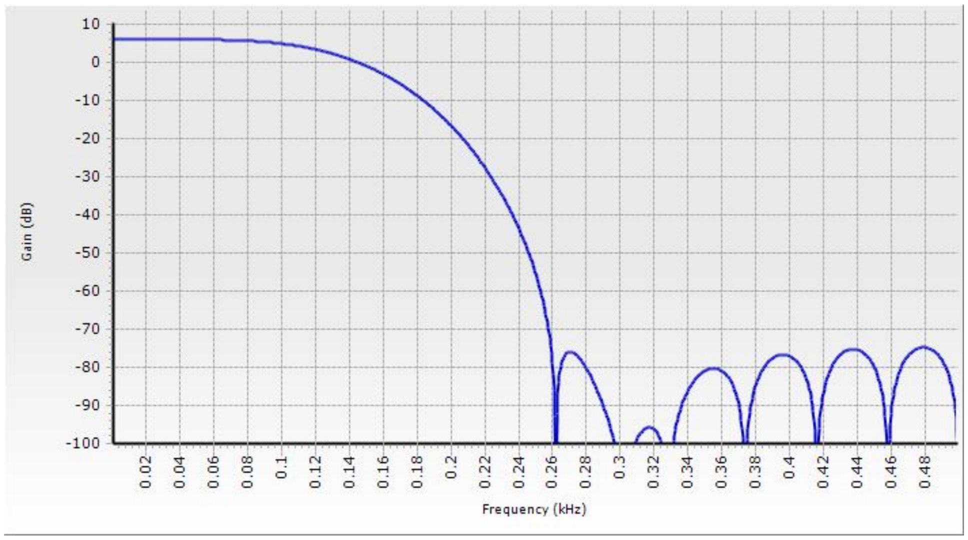


Figure 1: Response of starting 24-tap filter up to 0.5 F_{sin}.

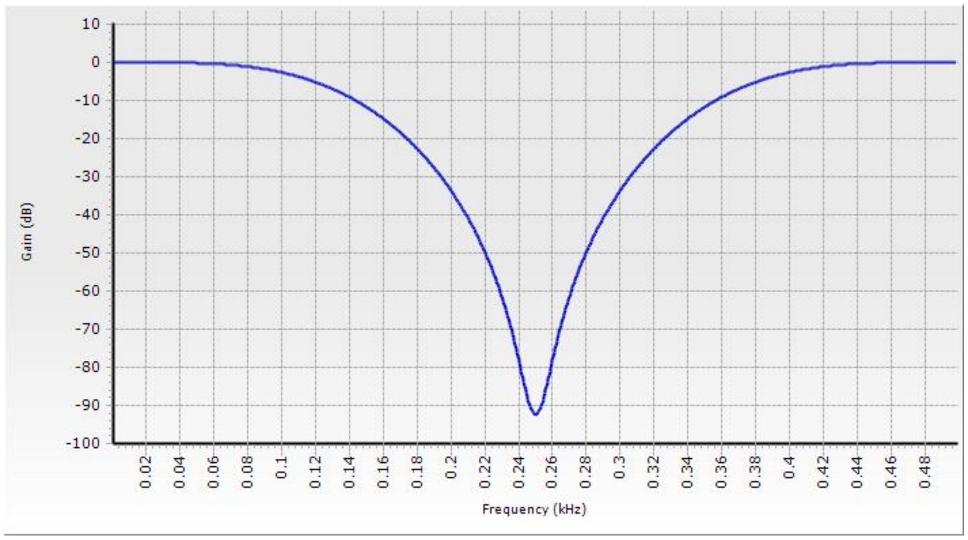


Figure 2: Equivalent response of each channel up to 0.5 F_{sin}.

To get the frequency response of the difference between the two channels, we simply plot the response of the difference between the two subfilter coefficient sets. The alternate zero coefficient values are aligned with the zero sample values (when the ADC is sampling the other channel). Analyzing this gives figure 3, which is nothing other than the highpass complement of figure 1. That's precisely what happens when you invert the sign of every other coefficient in a lowpass filter. One last detail – this plot is evaluated at the ADC's sample rate F_s, but actually, we are only taking output samples from the filters at $F_s/2$ So the important part of the response is the left-hand side, shown in figure 4. We have achieved a rejection of better than 75 dB for all valid passband frequencies between zero and half the output sample rate.

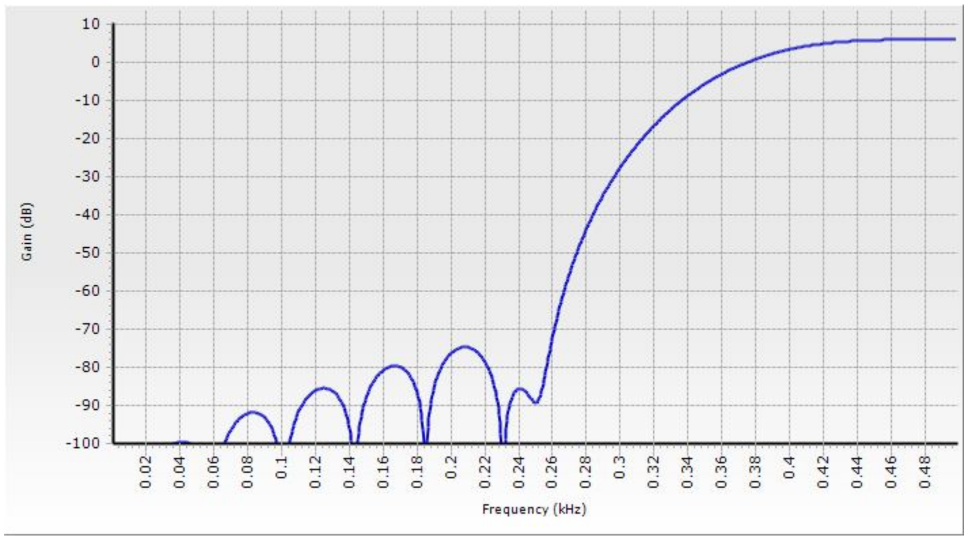


Figure 3: Response of the difference between channels up to 0.5 F_{sin}.

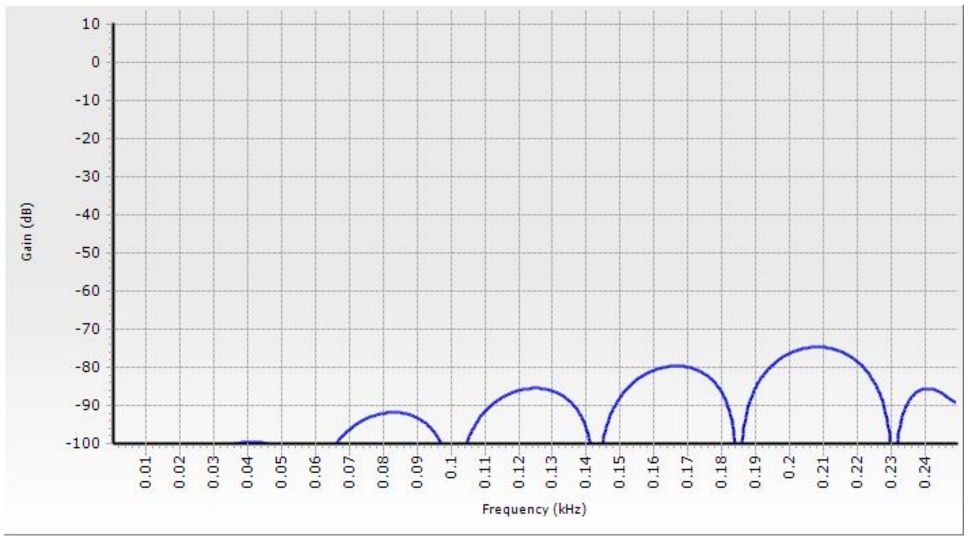


Figure 4: Response difference plotted up to 0.5 F_{sout}.

It's instructive to compare this with what we would have got if we had used the same filter for both channels, instead of a de-interleaved pair. I picked a 12-tap Blackman filter with cutoff at 0.26 F_s, producing an equivalent result table shown in table 2. Each coefficient is represented twice in the table, because it's applied to two successive samples from the ADC, one for each channel.

The result is shown in figures 5 (compare figure 3) and 6 (compare figure 4) and shows what we already knew; just subtracting two channels when they are not appropriately processed does cancel out the DC, but lets a great deal of AC through because the difference response only has a single zero at DC. The new process has an difference transfer function represented by a highpass filter of arbitrarily high order, dependent only on the choice of the initial lowpass prototype filter.

regular filter	phase A	phase B	output difference
0.00029	0.00029	0	-0.00029
0.00029	0	0.00029	0.00029
0.004507	0.004507	0	-0.00451
0.004507	0	0.004507	0.004507
-0.0111	-0.0111	0	0.011104
-0.0111	0	-0.0111	-0.0111
-0.04999	-0.04999	0	0.04999
-0.04999	0	-0.04999	-0.04999
0.104973	0.104973	0	-0.10497
0.104973	0	0.104973	0.104973
0.451344	0.451344	0	-0.45134
0.451344	0	0.451344	0.451344
0.451344	0.451344	0	-0.45134
0.451344	0	0.451344	0.451344
0.104973	0.104973	0	-0.10497
0.104973	0	0.104973	0.104973
-0.04999	-0.04999	0	0.04999
-0.04999	0	-0.04999	-0.04999
-0.0111	-0.0111	0	0.011104
-0.0111	0	-0.0111	-0.0111
0.004507	0.004507	0	-0.00451
0.004507	0	0.004507	0.004507
0.00029	0.00029	0	-0.00029
0.00029	0	0.00029	0.00029

Table 2: Equivalent table of coefficients when the channels use the same filter.

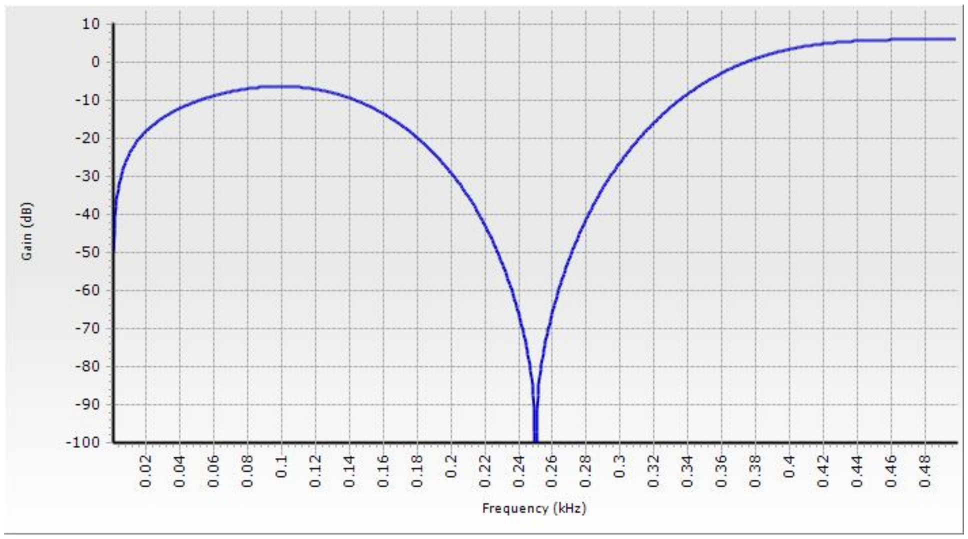


Figure 5: Rejection response to 0.5 F_{sin} with same filters on each channel.

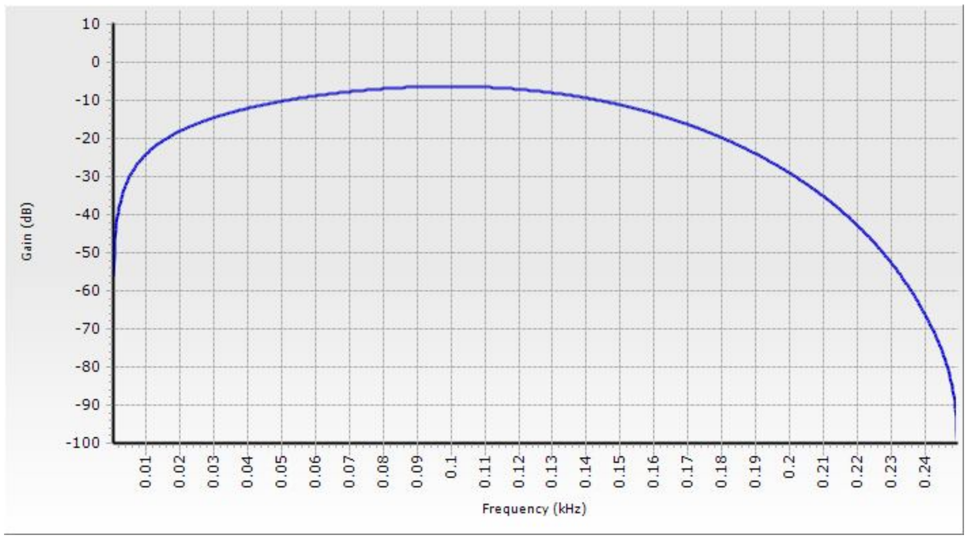


Figure 6: Rejection response up to 0.5 F_{sout} for same filters; compare figure 4.

Let's have a look at an eight-channel case. A 96-tap Blackman filter with a declared cutoff frequency of 0.0315 F_s has good rejection beginning at 0.0625 F_s, which is $F_s/16$, as required for the eight-channel system. The de-interleaving into eight 12-tap filters follows as before, and figure 7 shows the result of differencing, plotted up to 0.5 F_{sin}. Of course, in the final system after decimation, the only interesting results are below Fs/16, and figure 8 zooms into that range. It shows at least 85 dB rejection over the entire frequency range.

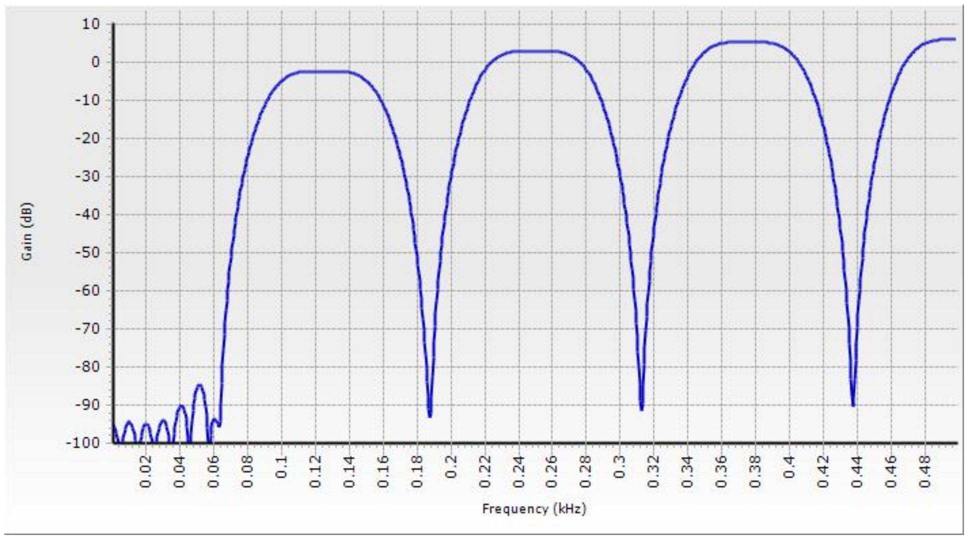


Figure 7: Channel difference for the 8-channel case, to 0.5 F_{sin}.

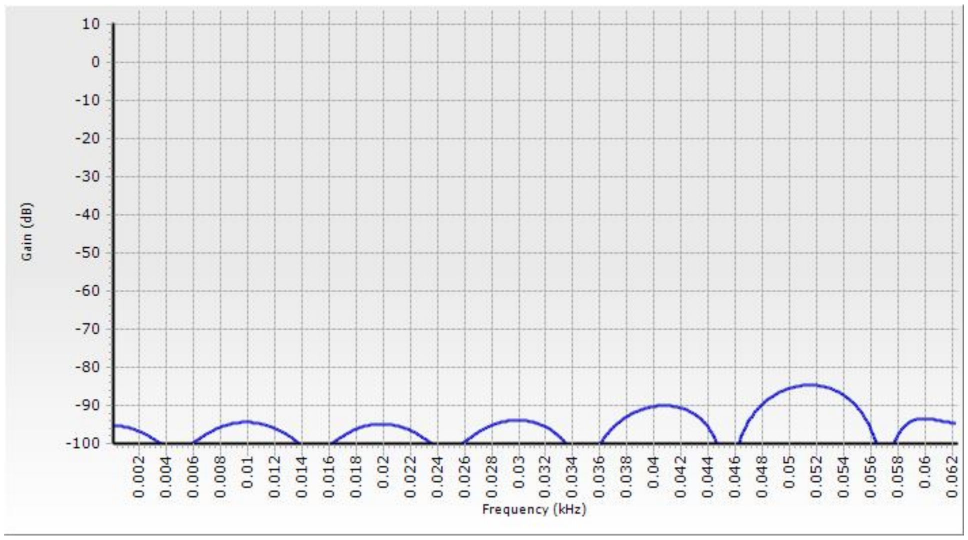


Figure 8: Channel difference for the 8-channel case, to 0.5 F_{sout}.

If you want to try this out, all these examples will fit comfortably in the Digital Filter Block in Cypress's PSoC 3 and PSoC 5 devices, and can be fed either from the high resolution delta-sigma ADC or the high-speed SAR converter (that one's only in the PSoC 5 family). This really illustrates the value of having a powerful chunk of filtering capability embedded in a general purpose system-on-chip.

So, whether you're doing correlated double sampling or just a bit of pseudo-differencing on your inputs, a set of time-alignment filters will help you to get all your channel ducks in a row. Nothing false or fraudulent about it at all; in fact, I'd say that it resembles a miracle! Happy 2013 / Kendall.