Question 1. We find the spec in our datasheet about offset error match between side A and side B is 150ppm. For this error, do we have calibrate solution?(Any offset calibrate equation or solution.)

Offset error		±500	±1000	ppm of FSR
Offset error match[5]		±150		ppm of FSR

Currently test result from customer side is different channel with different offset error between A and B. such as channel 1 A is bigger than B 200 ppm. Channel 2 B is bigger than A 300ppm. Channel 3 maybe the A is nearly same with B.

One would do an offset and gain calibration like one would do any other calibration. Not apply a signal (disconnect the output or at least have zero signal) and measure the output; or apply a known signal almost full-scale and measure the output. Then correct all data respect to a line going through those two points.

The usual question comes from the fact that every channel in the DDC has two integrators that alternate each other taking the samples. Then basically the calibration procedure is the same, just applied to each stream of data separately (A or B side) and then, after correction, both streams can be combined if needed.

Another approach is to sample at 2x the final speed and get every individual sample at the final rate by adding two of those samples. Now we can calibrate that as a single stream, as every sample contains an error from A and one from B.

Interestingly enough, this last method can actually result in lower noise. Please see below for more detailed explanation.

1. Introduction

DDC devices use traditionally two elements, say A and B, in ping-pong fashion to manage to integrate the input without losing any of the detector charge/current. One of the elements, say A, is used to take one sample, and the next sample in the same channel is taken with B. Then back to A, and so on.

As not two elements can be perfectly matched, this results in slight differences between consecutive samples of the same channel taken by each of those circuits, even if ideally (with noise removed) they should be the same. The main differences will manifest as an offset and a gain error:

- Offset. The output of the DDC may not be the ideal value (4095 in the 20b versions) even when there is no input current into that channel (input completely disconnected/floating). This is expected for any device out there, DDC or not (classic offset non-ideality on amplifiers). Nevertheless, in the DDCs, per the above explanation, that non-ideality may translate into different values for the A and the B sides. The offset error for any input as well as the mismatch between the A and B sides are specified in the datasheet.
- Gain. Along the same lines, the feedback capacitor, which basically sets the full-scale range, will not be the same between both sides, A and B. This will result on a gain difference between the two samples. I.e., a different result for the same input signal level depending on which one of the two integrators was used. Again, this can be found in the datasheet.

The following describes couple of relatively straightforward techniques to deal with this mismatch.

2. Correction techniques

There are two classical approaches to this problem:

1. Calibration
2. Sampling at 2x the intended sampling rate

2.1 Calibration

This approach is relatively straightforward and simply leverages the classical correction algorithms used, anyhow, to correct even single integrator systems. In this case, we will simply repeat that approach twice, once for the A side and once for B.

More in detail, let's assume that there is a single element doing the sampling. Users will traditionally measure offset and gain errors on that element and correct for them during operation. This is traditionally done by taking many samples with no signal (to measure offset) or, with a large signal close (say 90%) to the full-scale (to measure gain), averaging those samples (with or without signal) and recording the difference of those two averages to the ideal value. In a 20b DDC, the ideal values are 4095 (for no input) and ~90%*2^20 (for "full-scale" measurement).

Note on offset: usually in CT, offset measurement is taken right before the measurement of interest (for instance, the beginning of the scan, before the x-ray tube turns on) so that any future deviations to this value can be assumed to come from the x-ray signal.

Note on gain: traditionally in CT, any measurement is actually referred to a reference value measuring the output of the x-ray tube, at the same time. This way, gain errors due to variation on output intensity on the x-ray tube can also be removed together with the gain error on the device, the scintillator, etc.

In the end, once those two errors are measured for every channel, they are stored in a table. This is then used to apply a correction to the sampled data from every channel before combining it with the data from the other channels to get an image.

Back to the dual integrator case, all what we need to do here is to consider that there is not one offset and gain error per channel, but two, one for every side A or B. So, as the data is collected, the data should be separated on A and B sides. That is straightforward as data comes relative to CONV level (high or low). Then the two streams, A and B, for every channel are treated separately, i.e., offset and gain are calibrated and corrected like if they were two independent channels (like it was explained in the single integrator case). So, the calibration table would certainly be 2x the size of the previous case, but in current systems today this should represent no problem as memory/storage is widely available at very low cost. After correction, data can be combined in the same channel and with any other channel for image formation.

2.2 Sampling at 2x the intended sampling rate

In this second/alternative approach we actually treat A and B sides as two half samples of the final sample we want to take and combine them to get one sample. For instance, say that we want to integrate for 1ms, then we actually take the samples every 500us (CONV frequency equal to 1KHz) and add every two samples to get the one at 1ms.

As both samples (one from each side) are used to compute one single final sample, the errors on every final sample include the same original errors. No need to store two separate offsets and two separate gain errors. The offset and gain calibration are done like explained above for the single channel, i.e., like if the converter only had a single element to do the integration.

Obviously on the negative side there is the issue of having to sample at 2x the rate, which may result in higher power consumption. Nevertheless, this difference may be small or actually negligible depending on the converter (no or negligible scaling on power with sample rate).

Bonus points… On the flip side, besides saving memory, there is an interesting fact about many of the DDCs that may actually help obtain even better performance of the DDC by operating it this way: the noise in two continuous samples is correlated in a way that adding two consecutive samples results actually in the same or even lower noise than individual samples (instead of increasing by sqrt(2)). To illustrate this, let's apply this technique to the DDC264. We capture 512 samples for side A of every channel and 512 for the B side, in Range 0 (12.5pF) with input disconnected and 30pF internal cap (special test mode), and analyze the resulting noise (expressed in 20bit codes rms):

1. 1ms CONV. We compute the rms for every channel A or B side separately. In average we get **38.54 ppm of FSR** (0.48fCrms).
2. 250us CONV. Compute the rms for every channel A or B side separately. **38.38.** Basically the same as before. As we know, noise does not change much with integration time.
3. 250us CONV but now we get a new sample by adding two consecutive channel samples, i.e., an A followed by B. I.e., like integrating for 500us. Theoretically the resulting noise of the new sample should be the previous times sqrt(2) (if the samples are uncorrelated). Nevertheless, we get **39.5.** Almost no change!
4. 250us CONV but now we get a new sample by adding two consecutive A samples. I.e., like integrating for 500us (but not really as we are skipping the B). That should break any correlation between the two and indeed does. We get **54** which is very close to 38.38*SQRT(2) = **54.28** proving the classic theory for uncorrelated.

So, indeed, A and B are correlated in the DDC264. Knowing all this, it would be interesting to analyze what the noise would be if we chose half the range when we go at 2x the speed. For that we took some new data at 500us. Then we compute the STDEV (fCrms) for each of these groups:

STDEV in fCrms	One SIDE	A+B	Sqrt(2)*One side	A1+A2
Range 0 (12.5pC)	0.48	0.49	0.68	0.67
Range 1 (50pC)	0.56	0.59	0.79	0.8
Range 2 (100pC)	0.76	0.87	1.07	1.12
Range 3 (150pC)	0.9	1.16	1.27	1.27

Where "One SIDE" represents samples of either the A side or the B side (the result is the same), "A+B" is the sample resulting from adding two consecutive samples, one from A and one from B, and A1+A2 is the sample resulting from adding two consecutive samples on the A side (like A1+A2, A3+A4…). Assuming the samples are uncorrelated (which should be as they are separated by a sample from the other integrator), the last should be sqrt(2)*stdev of one side, and in fact they are as the table shows.

Notice that the one side column roughly matches the DS numbers (3rd column below for Cdet=30pF):

Range 0: 12.5 pC	0.2	0.25	0.38
Range 1: 50 pC	0.32	0.37	0.53
Range 2: 100 pC	0.51	0.55	0.71
Range 3: 150 pC	0.72	0.75	0.9

This experiment actually shows that instead of running at a given sample rate on say, Range 2 (100pC), we can run at 2x that speed with Range 1 (50pC) and add both samples. The first approach would yield an rms of 0.76fCrms, while the second would yield 0.59fCrms. Again, notice that I am not looking at the speed the data was taken to determine the rms as we have already shown that there is no rms variation with integration time.

3. Conclusion

We presented two methods to correct for A and B mismatches, one doubling the memory and the second doubling the speed, both yielding good results.

Question 2. About the integrate time, will short integrate time add more high frequency noise? Will this kind of noise have a big influence on the accuracy?

Basically in charge terms, different integration times do not affect the final noise (see the specific DDC datasheet). Nevertheless, they certainly do change the response bandwidth and if there are external noise sources (like the resistor on app note SBAA034 example or just external interferers), there could be different results. See app note SBAA034 for an explanation on the gated integrator (DDC input circuit) bandwidth response. Still:

1. If the board is designed properly to reduce external noise sources, one should see no difference.
2. If one chooses to sample faster a signal that is slow, one could still apply digital filtering to remove the external higher frequency (off-signal) noise sources…

App Note SBAA034: https://www.ti.com/lit/pdf/sbaa034