Interleaved Adc

Alfredo Tomasini

Introduction

The problem designing high speed pipeline ADC is related to the solution of few problems

1. getting high speed/gain amplifiers

2. optimize the usage of the available time during sampling and MDAC function.

In this I do not consider other aspect of the problem like correction for partial setting or gain error. We want to select the sampling clock to be such that the amplifier settling is correct for a give resolution.

Interleave ADC

A common technique for achieving high speed consist of using multiple ADC in parallel (aka interleave adc or parallel adc). The approach used for these adc is to run each adc at FS and multiplex the input signal among them to get an final digital output at P*FS, where P is the number of adc. On this subject there is a lot of literature [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11].

However, this technique require some kind of calibration in order to compensate for gain and offset error among the adc's. The problem of gain and offset mismatch among channels will cause DNL and INL problem. So far none of the previous papers tried to address the problem of having a close formula for the SFDR as function of the input pattern for offset and gain mismatch. The best reference on this subject is probably [5]

Offset error

The effect of the offset error is pretty intuitive. If each adc has an offset OF[i] for a constant input voltage the output of the adc will generate a non-uniform but predicable patter that repeats itself every FS/P, where FS is the main clock and FS/P is the sampling rate of each ADC (fig 1).

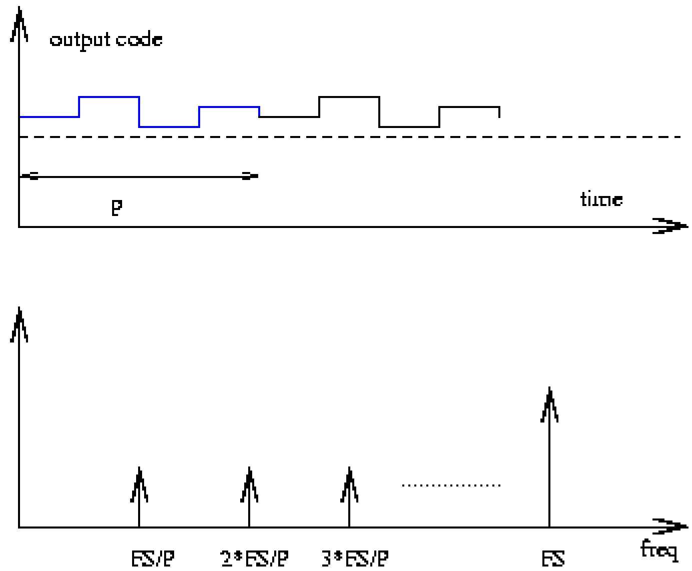


fig 1

The intention here is to get an estimated value of the spurs without having to go through a complex calculation the best will be to get an estimated value of the SFDR given a specific pattern. A possible help comes from the Fourier theory in particular Parseval's Theorem eq 1.

$$\int x(t)^2\,dt = \frac{1}{2\pi}\int_{-\infty}^{+\infty} x(\omega)^2\,d\omega \quad \text{eq 1}$$

This theorem link the energy content of the signal in the time domain and frequency domain leading us to a possible estimation of the SFDR. First let us refresh our mind with some ADC equation. For N bit resolution the RMS power associated with the full scale input sinusoidal signal is

$$\frac{2^{(2N)} LSB^2}{8} \quad \text{eq 1.1}$$

Where LBS = Full_scale/2^N. The power for each tone generated by the offset signal will be:

$$\frac{1}{2} FS/P \int_0^{P/FS} x'(t)^2\,dt \quad \text{eq 1.2}$$

where x'(t) = x(t)- avg(x(t)), P/FS is the length, in time, of the offset pattern, and the ½ takes into account that tones come in pairs each one carrying 50% of the energy (remember we want to know SFDR), by combining eq. 1.1 and eq. 1.2:

$$\text{SFDR} >= \frac{\dfrac{2^{(2N+1)} LSB^2}{8}}{FS/P \displaystyle\int_0^{P/FS} x(t)^2\,dt} \quad \text{eq 2}$$

By expressing x'(t) in LSB x"(t)=x'(t)/LBS we can simplify eq 2 into eq 2.1 (x" can be fraction of LSB).

$$\frac{2^{(2N-2)}}{FS/P \int_0^{P/FS} x''(t)^2\, dt} = \frac{P\, 2^{(2N-2)}}{\sum_0^{P-1} x''[n]^2} \qquad \text{eq 2.1}$$

For tones that are equally distributed there is an upper bound value for the best SFDR eq 2.2.

$$\frac{P\, 2^{(2N-2)}}{\sum_0^{P-1} x''[n]^2} \le SFDR \le \frac{P^2\, 2^{(2N-2)}}{\sum_0^{P-1} x''[n]^2} \qquad \text{eq 2.2}$$

Another way to look at this result is by dividing both numerator and denominator by P and notice that on the denominator we get the equation of the standard deviation of the offset.

$$\frac{2^{(2N-2)}}{\sigma^2_{offset}} \le SFDR \le \frac{P\, 2^{(2N-2)}}{\sigma^2_{offset}} \qquad \text{eq 2.3}$$

Example:
In this example we use a 4 channels adc, each adc is ideal 14 bit with offset uniformly distributed with standard deviation of 1,16 LBS.
Fig 2 shows the result of the simulation with a program that add the offset to the input signal. Fig 3 compares the simulated value with the one calculated from eq 2.3, the two results are in very good agreement.

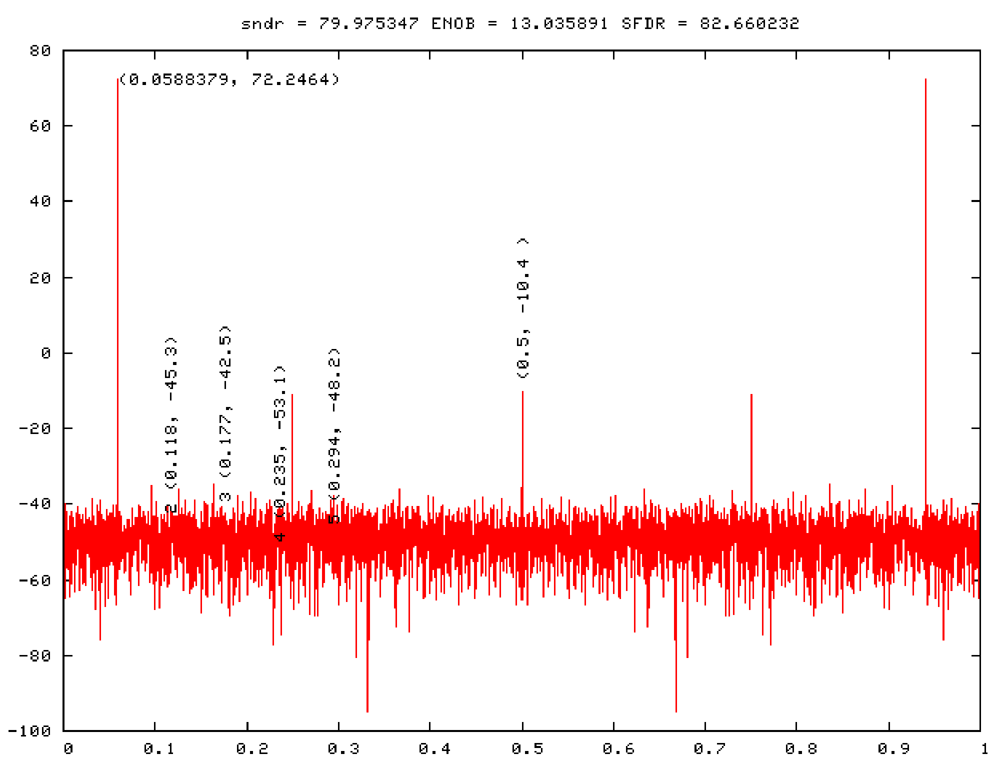


fig 2

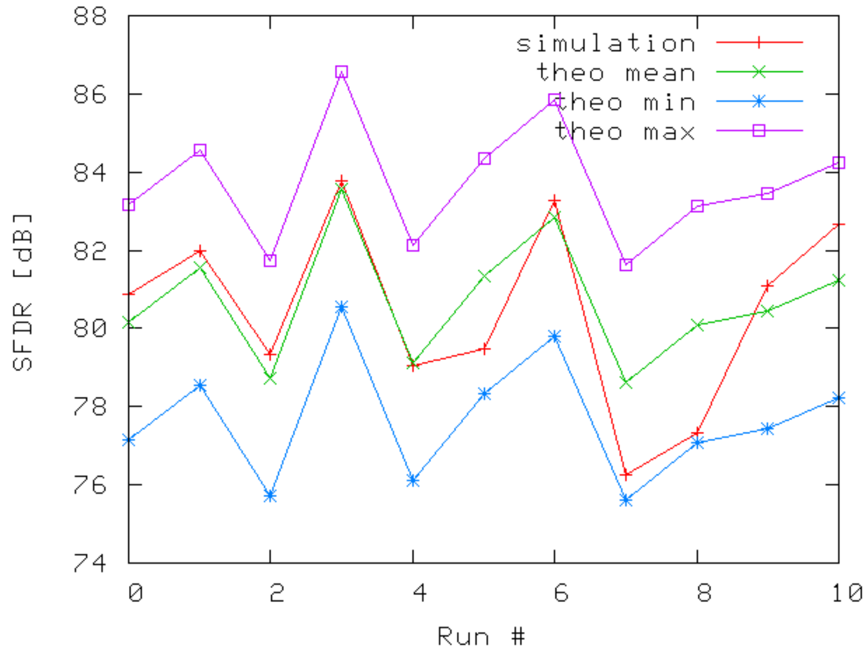


fig 3

Gain error

This error is the effect of different transfer function between different adc. This does not mean that each ADC has linearity problem it means that the conversion slope is slightly different among them (fig 4).

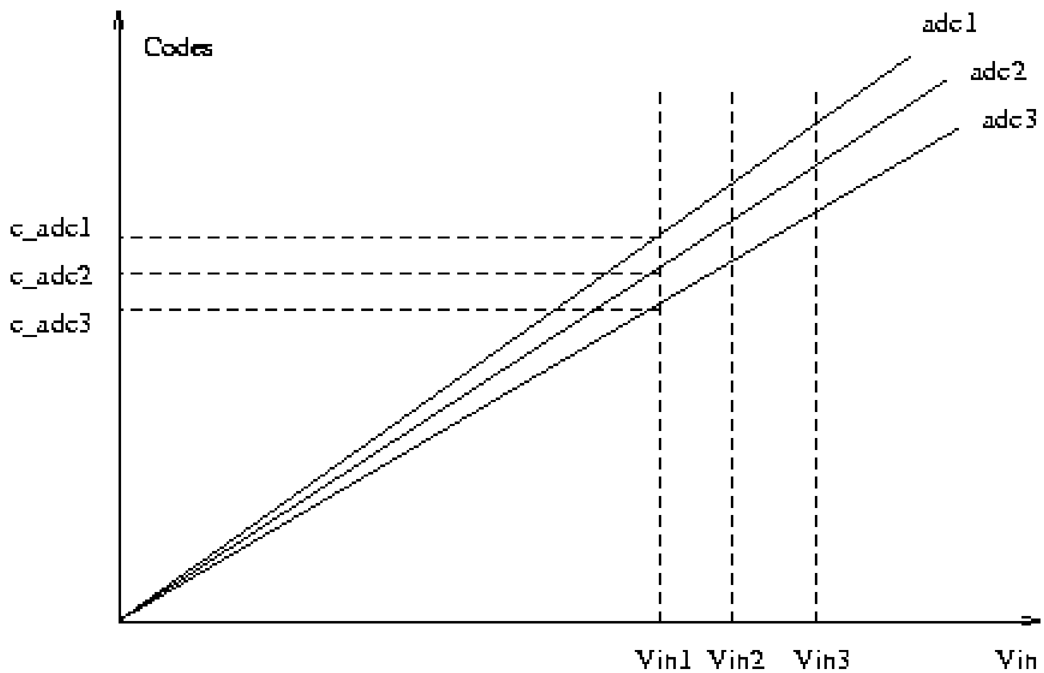


fig 4

Because of the various slopes S1,S2 and S3, the output value from each adc will be different for a given input Vin. If we call S[n] the slopes of adc n, the output code as function of the adc will be:

$$Code[n] = \langle Vin*S[n] \rangle$$

where S[n] = G[n]*(2^N)/Vrange = G[n]/LSB and <r> is the integer part of r. S[n} is periodic with period P (number of channels) and it can be expanded as Fourier series having tone at K*(P/FS) Hz or w=k w_p /2pi with k=1,2,3 ..... P-1 (eq. 3)

$$\hat{S}(\omega)=\sum c_n * \delta(\omega - n*\omega_p) \quad \text{eq 3}$$

If a input signal Vin(t)=Asin(w_i t), this will generate a two tones at frequency +-w_i. The spectrum at the output of the adc will be obtained by convolving the spectrum of the input signal Vin with the spectrum of eq 3.

$$\hat{code}=\int A\delta(\omega \pm \omega_i - x) * \sum c_n \delta(x - n\omega_p) dx = \sum A c_n \delta(\omega - n\omega_p \pm \omega_i) \quad \text{eq 4}$$

graphically (fig 5).

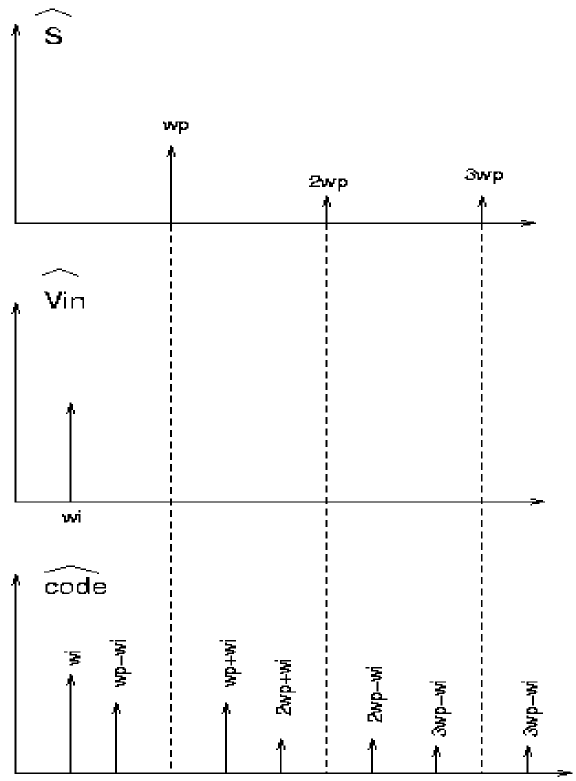
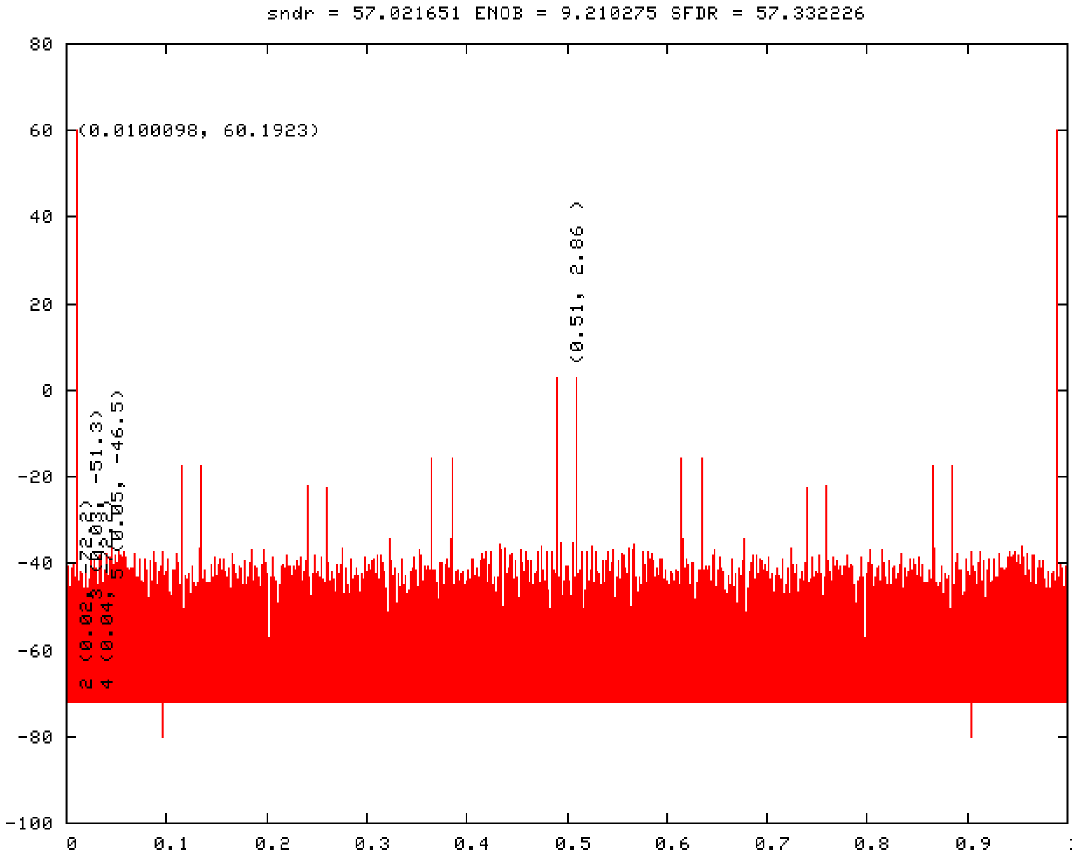


fig 5

Fig 6 shows a simulation with a combination of eight 12 bit adc having a random gain mismatch error in the range of +-0.2%.

fig 6



The output shows tones and it will be interesting to find and estimated value for the SFDR. By looking at the expression of S[n] as:

$$code = \frac{Vin}{Vrange} G[n] 2^N \quad \text{eq 5}$$

where G[n] is the gain for each channel. By removing the common gain we can write the variation of the output code in this way:

$$\Delta(LSB * code) = G'[n] \quad \text{eq 6}$$

where G'[n] = (1 - G[n]). The power spectrum associated with this code variations can be estimated with Parseval's theorem

$$P(G') = FS/P \int_0^{P/FS} (G'[t])^2 dt = \frac{1}{P} \sum_{n=0}^{P-1} (G'[n])^2 \quad \text{eq 6}$$

Equation 4 shows that the output modulated get multiplied by the amplitude A of the input signal by using the definition of LSB=2*A/2^N (remember A is the amplitude of a sinusoidal waveform) the new P(G') is:

$$P(G') = FS/P \int_0^{P/FS} (LBS\, 2^{N-1} G'[t])^2 dt = \frac{1}{P} \sum_{n=0}^{P-1} (LBS\, 2^{N-1} G'[n])^2 \quad \text{eq 7}$$

$$P(G')=2^{2N-2}\frac{LSB^2}{P}\sum_{n=0}^{P-1}(G'[n])^2 \quad \text{eq 8}$$

by combining eq 1.1 with eq 8 and considering that spurs energy is spit in pairs P(G') we need to divide by 2 to get the lowest value for the SFDR.

$$\text{SFDR} >= \frac{\dfrac{2^{(2N)}LSB^2}{8}}{2^{2N-2}\dfrac{LSB^2}{2P}\sum_{n=0}^{P-1}(G'[n])^2}=\frac{P}{\sum_{n=0}^{P-1}(G'[n])^2} \quad \text{eq 9}$$

An interesting observation is that the number of tones depends upon the number of pattern change in the gain mismatch, but it does not depends from the adc resolution.

Some examples:

Fig 7 shows the output for a sequence of 2 adc with gain error [0.0002,-0.0002], as expected we there 2 tones centered at FS/2. How the simulation works is explained in the next section.

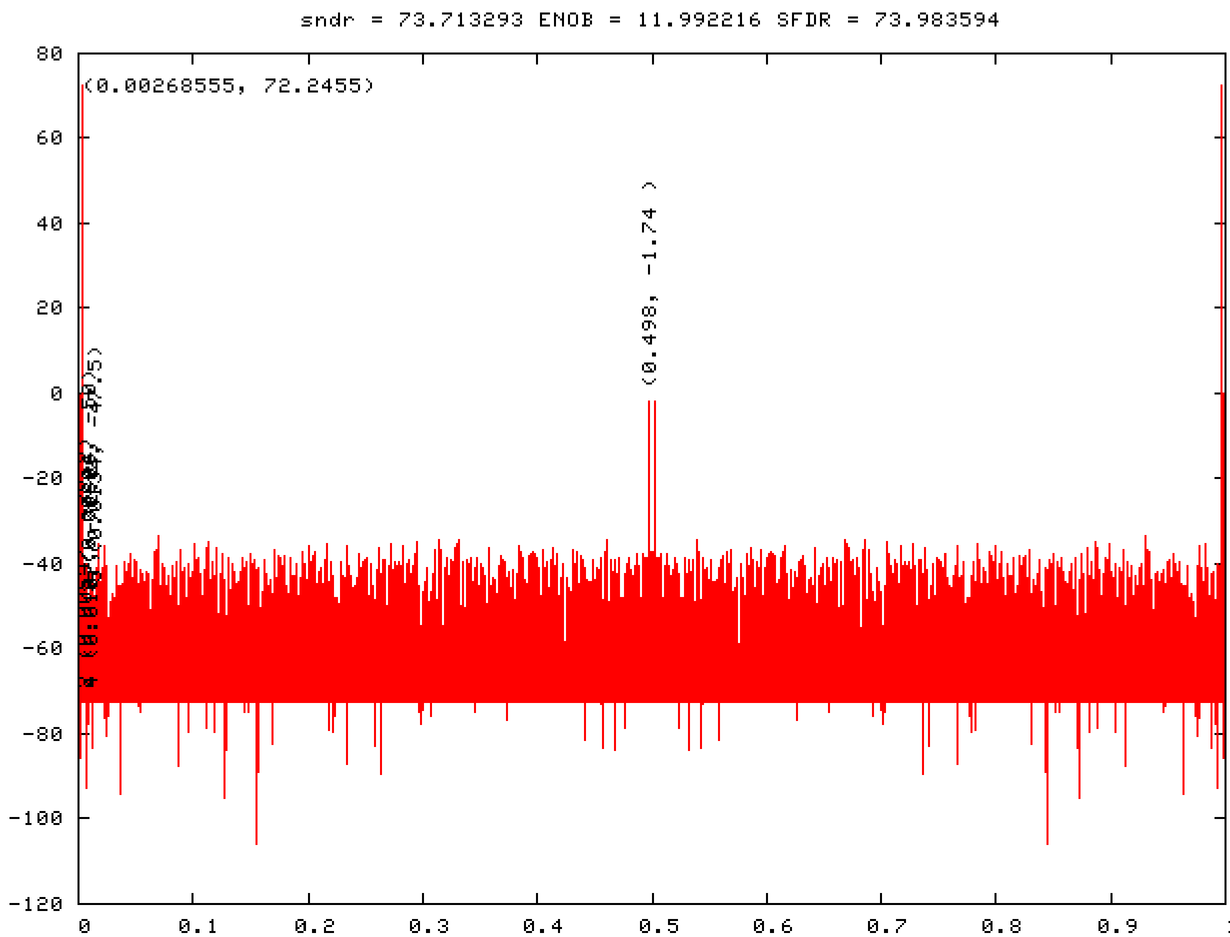


fig 7 : gain error pattern for 2 channels adc [0.0002,-0.0002]

The sequence [0.0002,-0.0002,0.0002,-0.0002] for a 4 channel will generate the same 2 tones too at FS/2 because it has the same energy of the 2 tones in fig 7

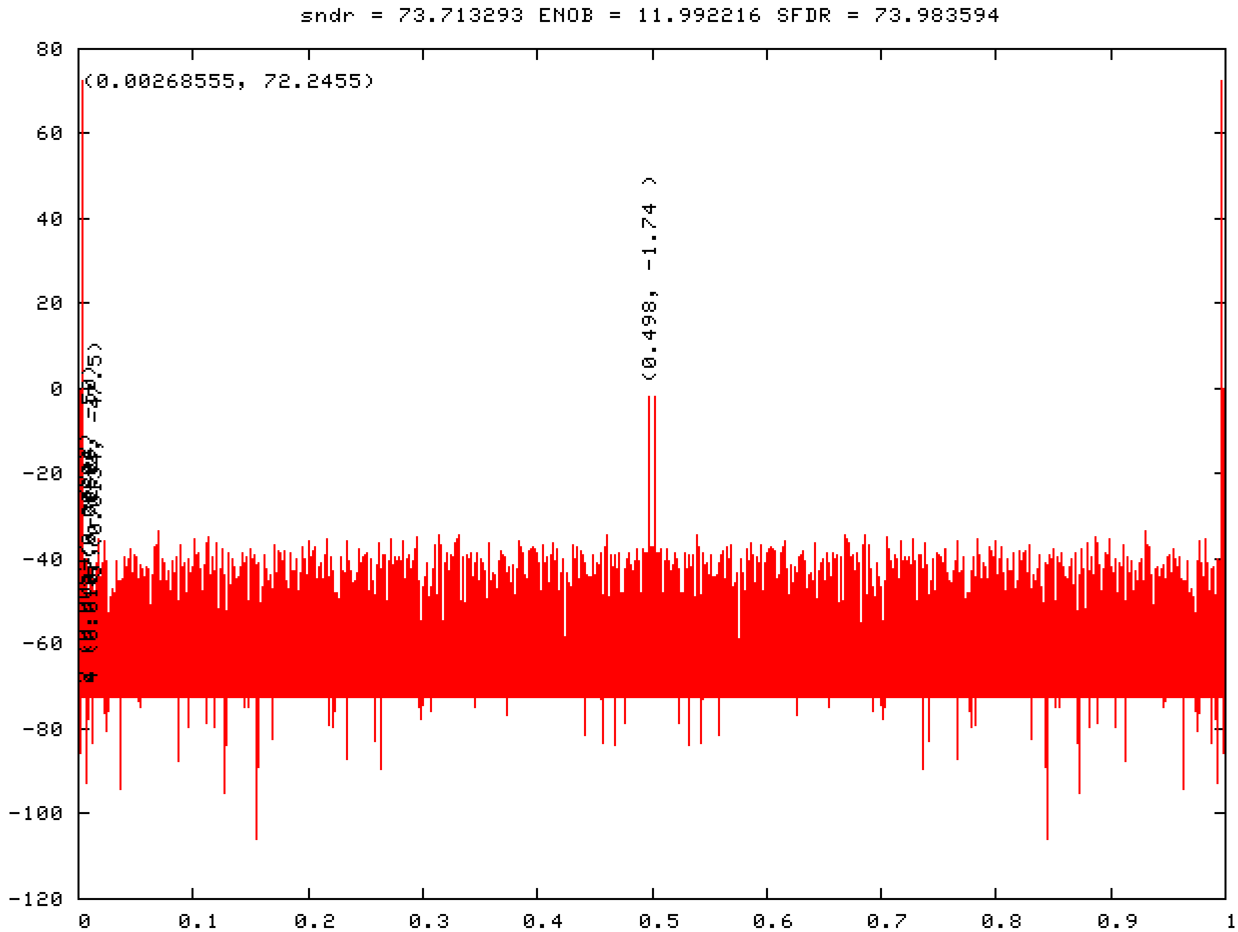


fig 8 : gain error pattern for 4 channels adc [0.0002,-0.0002,0.0002,-0.0002]

*Eq. 9 predicts SFDR = 10*log10(P/(P*(0.0002^2)))=73.979 dB in both cases which is correct because this pattern concentrate all the energy in two tones.*

Among all possible pattern with the same total energy, we can find the one that spread power equally between the P tones. This is the best case and the SFDR is going to be P time the value of Eq.9 and we can get an upper bound to the SFDR eq 10.

$$\frac{P}{\sum_{n=0}^{P-1}(G'[n])^2} \leq SFDR \leq \frac{P^2}{\sum_{n=0}^{P-1}(G'[n])^2} \quad \text{eq 10}$$

Simulation

The results of fig 7 and 8 comes from a direct approach by generating a verctor that the signal is traversing and send them to the ADC. The gains vector G' is first normalize to the max gain to get G" = G'/max(abs(G')), this prevent out of range on the following ADC.
The G" patter is applied to the signal samples in a repetitive sequence as shown in fig 9, the sequence is sent to the ideal ADC and process with FFT to get SNDR and SFDR.

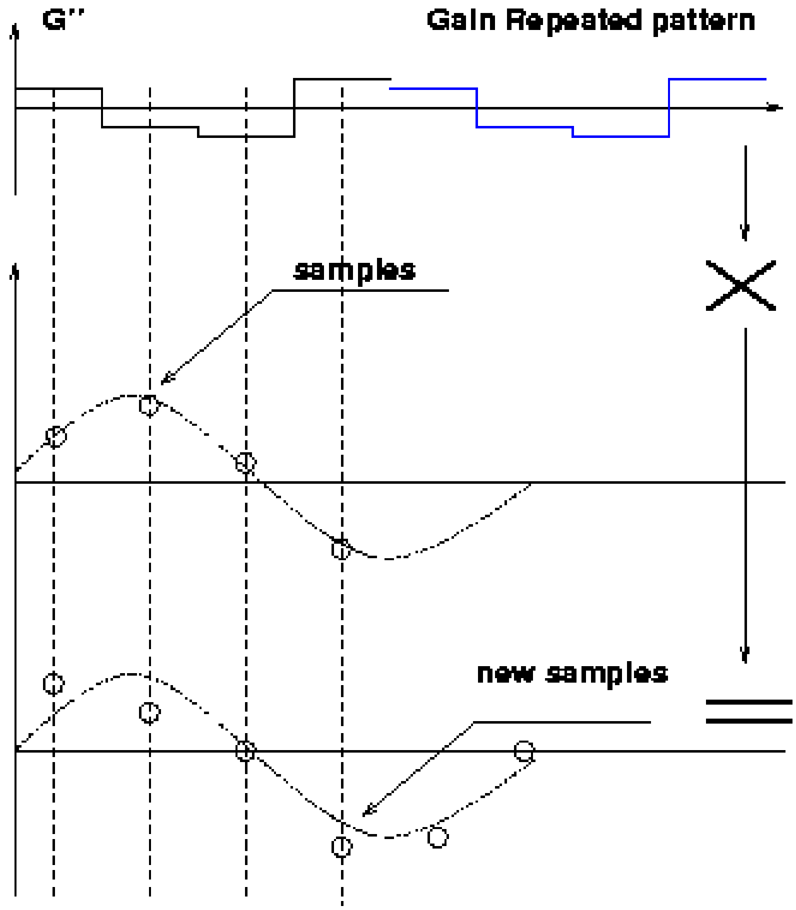


fig 9 : procedure to generate the samples (without normalization)

The simulator compute the SFDR give by eq 9 from the input vector G. In order to get G' we need to do some operation like calculating the mean and normalize it to avoid out of range.
Table 1 shows 10 simulations for a 4 channel adc with gain randomly selected within +-0.003%.

Table 1 SFDR for a 4 channels ADC

theoretical value		***simulated***
min	max	sim
82.64	88.66	84.76
80.53	86.55	85.23
81.24	87.26	85.8
81.75	87.77	84.94
81.29	87.31	84.67
82.05	88.07	84.97
81.8	87.82	82.15
86.29	92.32	88.97
80.94	86.96	85.13
97.33	103.35	101.42

Fig 10 shows the plot of the result in table 1

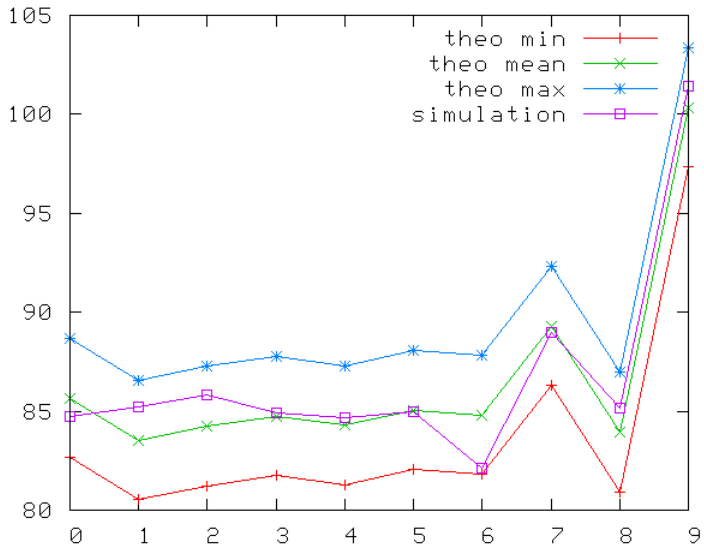


fig 10 ; 4 channel ADC with random sequence of gains mismatch

For the 2 channel ADC there are 2 tones only, in this situation there no best or worst case because there are not other tones where the spurs can land, the SFDR predicted by Eq 9 has to match exactly the simulated value. This is shown in fig 11 confirm, the SFDR plots from the simulation overlap the plot predicted by Eq 9.

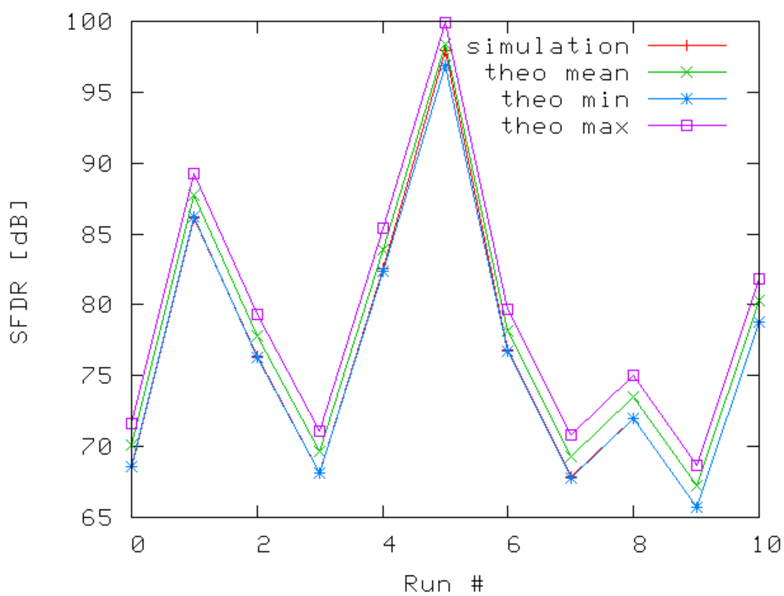


fig 11 : 2 channels ADC with random sequence of gains mismatch

We can write Eq 10 in a form that link the gain standard deviation to the SFDR, in fact:

$$\frac{1}{\frac{1}{P}\sum_{n=0}^{P-1}(G'[n])^2}=\frac{1}{\sigma_{gain}^2}$$

and

$$\frac{1}{\sigma_{gain}^2} \le SFDR \le \frac{P}{\sigma_{gain}^2} \quad \text{eq 11}$$

By equating the SFDR to the SNR = 3/2*2^2N, the gain variance must be bounded within the range of eq 12 in order to contribute to the SNDR no more than 3 dB.

$$\frac{1}{\sigma_{gain}^2} \le \frac{3}{2} 2^{2N} \le \frac{P}{\sigma_{gain}^2} \Rightarrow \sqrt{\frac{2}{3}} \frac{1}{2^N} \le \sigma_{gain} \le \sqrt{\frac{2P}{3}} \frac{1}{2^N} \quad \text{eq 12}$$

Another observation that comes out from eq 12 is that the gain variance can grow with the length of the pattern (number of channels) because the extra energy is spread across a larger number of tones. However the worst case is when the gain pattern is such that all the energy is concentrated in 2 tones. In that case the gain variation should be no more than what is estimated from Eq1 with P=1.
Typical number of channels range from 2 to 4 than sqrt(4/3) = 1.15 and sqrt(4/3) = 1.63, this is telling us that the gains variation has to be in the same order of a LSB to keep the SNDR within 3 dB from the ideal case.

Time error

The time error we want to consider here are not random error like jitter but systematic error at the sampling time that depends of the specific channel. The model assume that D(k) is the time error respect the nominal sampling time for the channel k. For a sinusoidal input waveform we can write:

$$\sin(\omega t) - \sin(\omega(t + d(k))) = \sin(\omega t) * (1 - \cos(\omega d(k))) - \cos(\omega t) \sin(\omega d(k))$$

A fair approximation is to assume that the error in time is such that w*d(k) << 1 and using Taylor to expand sin(wd(k)) we get:

$$\sin(\omega t) - \sin(\omega(t + d(k))) \approx (\omega d(k)) \cos(\omega t) \quad \text{eq 13}$$

This equation shows that a time error is equivalent to a gain error with amplitude E=w*d(k), then we can apply the result of eq 11 and get eq 14:

$$\frac{1}{\sigma_E^2} \le SFDR \le \frac{P}{\sigma_E^2} \quad \text{eq 14}$$

With eq 13 and 14 we had transformed an phase modulation into amplitude modulation. Well this is not a big error because eq 14 in combination with eq 12 is telling us that the standard deviation has to be in the range of 1/(2^N) in order to limit the effect of spurs within 3 db from the ideal SNDR. In other words w*d(k) cames to be really << 1 for N >=8.

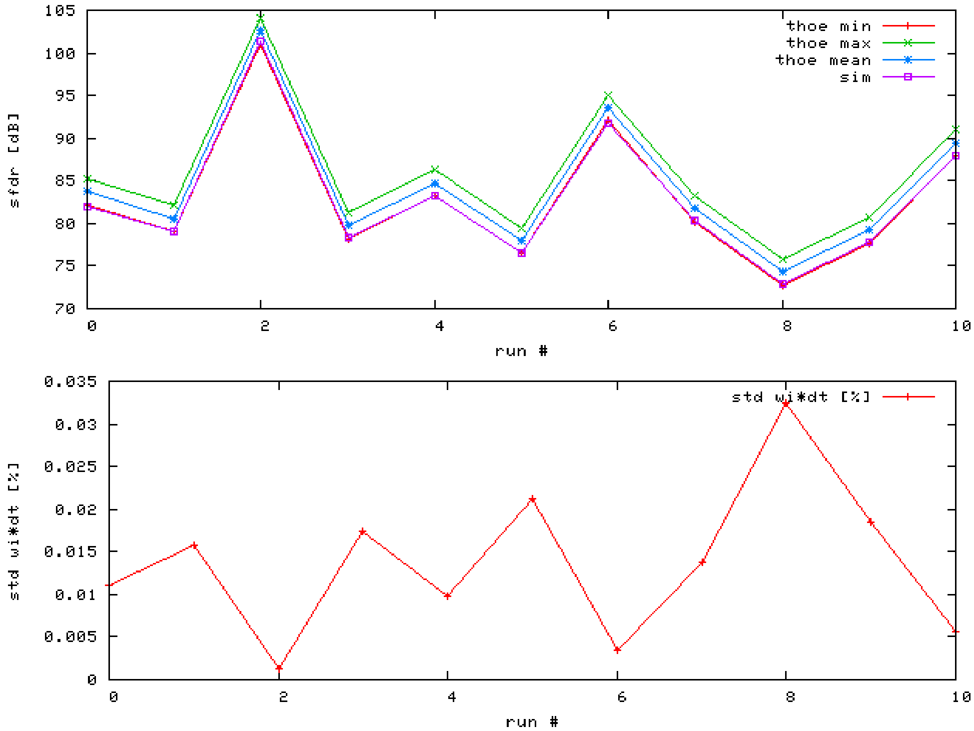


fig 12: 2 channels acd

Fig 12 shows compares the SFDR theoretically calculated with eq 14 and the simulation for a 2 channels 14 bit ADC. like in previous cases the SFDR for a two channels adc match all the time the minimum estimated from eq 14. The usual reason is because the are only tow identical tones in the spectrum. Top side of fig 14 shows the SFDR and the bottom is the SIGMA in % for the product w*d(t). For the SFDR the theoretical values and the simulation match very well.

Note: When the time error w*d(t) is getting very small the SFDR calculated with eq 14 is better than the real value, this because it falls into the quantization noise.

Fig 13 presents a set of simulations and theoretical SFDR for a 4 channels adc. Simulation and estimated value via eq 14 are in prefect agreement.

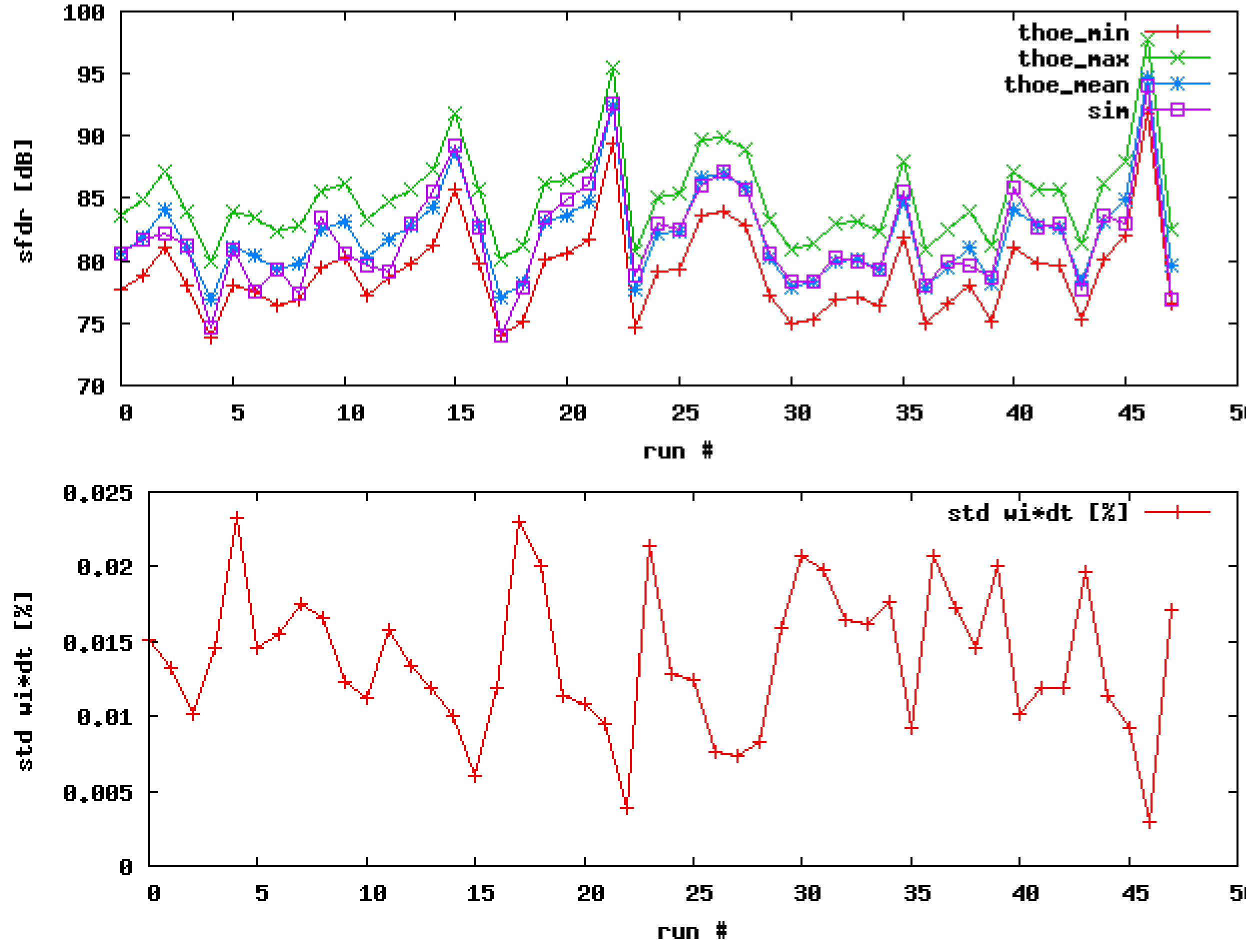


fig 13: 4 channels adc

Time errors generate spurs at the same frequency of the gain error and they are not distinguishable while the offset error is totally independents from these two. Another observation is that time error and gain error are opposite in phase, the first one is function of the slope of the signal the other is function of the amplitude. This characteristic is also well shows by eq 13, the input signal is a sinusoid but the output is a co-sinusoidal waveform.

Conclusion

In this documents we had analyze three different source of problems that effect parallel/interleave adc and provide a close form for the SFDR for each of them.
The first source of error is offset mismatch which creates spurs at K*FS/P and with SFDR bounded by eq 2.3, with P being the number of channels. The second source of errors is gain mismatch this cause spurs at frequencies (+-Fin + K*FS/P) the SFDR in this case is provided by eq 11.
last source of problem is systematic error in time between the channels, this error create spurs at the same spot of the gain error at it is highly function of the input frequency, in particular W*d(t). Eq 14 provides a way to estimate the SFDR in this case.

The theoretical aspect has been also check with against simulation in Octave. The code call adc_sim defines three main functions

```
adc_sim_offset(N,ofsamp,"comment")
adc_sim_gain(N,ogain,"comment")
adc_sim_time(N,otime,"comment")
```

N is the number of bit ofsamp, ogain, otime are vectors that represent the offset, gain and time mismatch between the channels, ofsamp is in LSB, the other two are % of the nominal value.
For instance something like this:

```
adc_sim_time(14,0.04*2*(0.5-rand(4)(2,:)))
```

will simulate and calculate the SFDR for a 4 channel 14 bit ADC with four random errors within +-0.04%, the output will be.

```
otime =

   -1.8838e-04    2.1657e-04   -6.3029e-05    7.0562e-05

norm_otime =

   -1.9731e-04    2.0764e-04   -7.1961e-05    6.1629e-05

# min_sfdr max_sfdr   mean_sfdr  std wi*dt [%]   SFDR sim   ENOB sim
76.429     82.45      79.439     0.017419        77.37      12.361
plot on /tmp/tplot.png data log into "/tmp/tlog.log"
```