Calibration Techniques for Optimizing Performance of High-Speed ADCs

Ewout Martens, Nereo Markulic, Jorge Lagos Benites, and Jan Craninckx

imec, Leuven, Belgium

Abstract – This paper discusses recent developments in digitally assisted analog-to-digital converters. Foreground and background calibration techniques can be leveraged to realize energy-efficient high-speed ADCs with high accuracies. Trade-offs between different approaches are elaborated exemplified by details of some recently proposed techniques for pipeline, SAR and pipelined-SAR ADCs.

I. Introduction

Recent advances in ADC design focus on improving sampling speed and/or conversion accuracy while keeping power and area small. To realize such high-performance designs in advanced technologies, several calibration techniques are adopted to overcome various technology imperfections like mismatches, nonlinearities and parasitics. For example, rather than increasing the power of a residue amplifier to improve its linearity, a better solution might be to save power in a nonlinear amplifier and correct it via digital calibration. Indeed, digital processing tends to become more efficient with technology scaling whereas improvements in analog properties are limited. This paper discusses a framework with general properties of digital-intensive calibration techniques for ADCs and illustrates them with some recent examples.

Calibration techniques for ADCs can be classified according to various criteria [1][2]. In Fig. 1, an overview is shown of different dimensions that characterize these techniques. Application-driven demands dictate which of these dimensions must be prioritized to implement a cost-efficient and reliable solution.

II. Error types

A first dimension deals with the kind of errors to correct. For Nyquist-ADC architectures like flash, SAR, pipeline, and pipelined-SAR, the errors typically targeted by digital calibration are listed per type of building block in Fig. 2. Within a channel, offset, gain and non-linearity are most common. Once the signal is sampled, any error becomes independent of the frequency of the input signal making it a *static* error. Hence, the input-output relation for the *n*-th sample is described by a simple albeit nonlinear function:

$$v_{out}[n] = f(v_{in}[n]). \quad (1)$$

Memory effects occur due to, e.g., incomplete settling or reset making the relationship a bit more complicated but still static:

$$v_{out}[n] = f(v_{in}[n], v_{in}[n-1], ...). \quad (2)$$

The input buffer and sampling process have a frequency-dependent behavior and hence introduce *dynamic* errors. The technique used for calibration needs to reflect the same properties to detect and correct the resulting errors.

High-speed ADCs use multiple interleaved channels to realize high sampling speeds, but the mismatches between the channels lead to various spurs [3] (*IL* = number of channels and f_{CK} = the overall clock frequency of the ADC):

- *offset spurs* at $k \cdot f_{CK} / IL$ independent of the input signal;
- *gain spurs* at $k \cdot f_{CK} / IL \pm f_{in}$ with levels independent of the frequency of the input signal;
- *skew spurs* at $k \cdot f_{CK} / IL \pm f_{in}$ due to a different phase shift per channel increasing with the frequency of the input signal;

	input buffer	sampler	reference buffer	DAC	comparator	residue amplifier
within channel	non-linearity	non-linearity	offset settling gain	weights gain	offset	offset gain non-linearity
interleaved	offset gain bandwidth	time skew bandwidth	gain	gain	offset	offset gain

dynamic & static ⟵ ⟶ static

Fig. 2. Main errors corrected by digital calibration in ADCs.

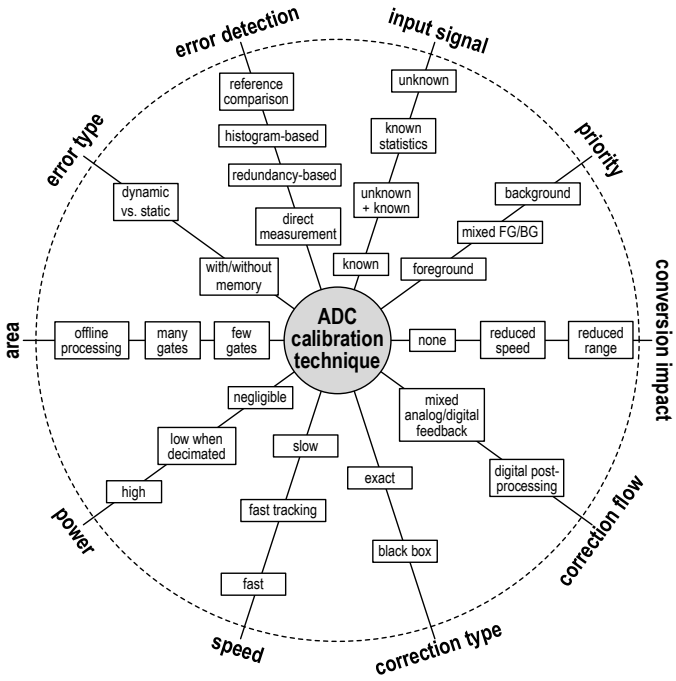


Fig. 1. Different dimensions of calibration techniques for ADCs.

- *bandwidth spurs* at $k \cdot f_{CK} / IL \pm f_{in}$ due to different amplitude and phase transfer per channel.

Furthermore, as illustrated in Fig. 3, when there is distortion in the system, extra spurs arise due to the intermodulation of the already existing spurs. The bottom row in Fig. 2 indicates the type of spurs to expect per block when the channels contain different copies of them. For instance, different input buffers per channel [4] might result in offset, gain and/or bandwidth spurs.

III. Error detection

Once the errors to calibrate have been identified, a detection mechanism for the errors must be designed. As indicated in Fig. 1, there are different approaches:

- A *direct measurement* of the error looks at some quantity inside the ADC and translates that to an error magnitude.
- With *redundancy* in the system, looking at the raw output bits reveals when the redundancy is activated which is an indication of some error happening.
- Errors can be detected by observing the distribution of the output signals in the form of a *histogram*. Its mean value and width, for instance, are measures for the offset and gain of comparators and residue amplifiers.
- The outputs can be *compared* to a *reference* conversion to detect errors. For instance, split-ADC techniques [5] use an explicit reference ADC, and when calibrating interleaving spurs, any channel can be chosen as reference.

An example of a direct measurement with a pipeline ADC is shown in Fig. 4 [7]. A closed-loop residue amplifier exhibits errors due to the finite open-loop gain of its core amplifier. This effect can be quantized

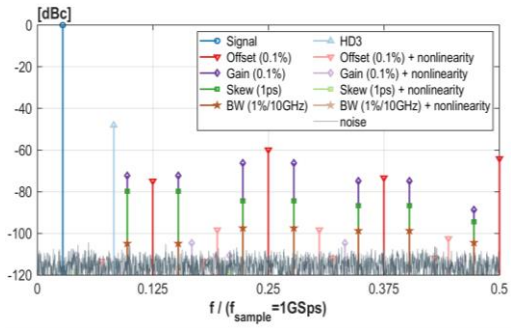
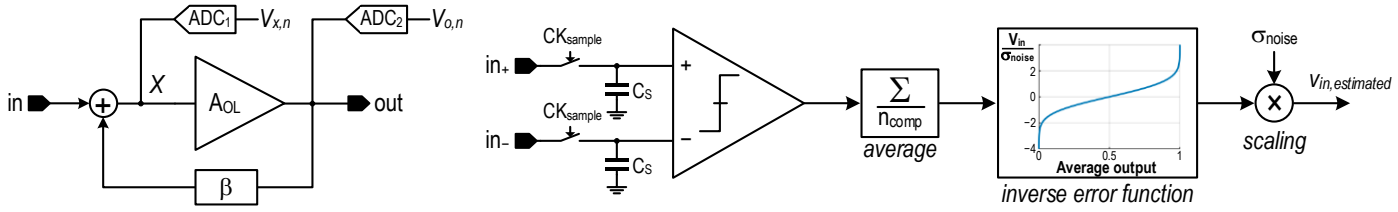


Fig. 3. Example of spurs arising due to 8x interleaving.



(a) Direct measurement of errors in residue amplifier. *(b) Stochastic ADC approach for direct measurement of errors in ADC.*

Fig. 4. Direct measurement of errors in a pipeline ADC using a stochastic ADC [7].

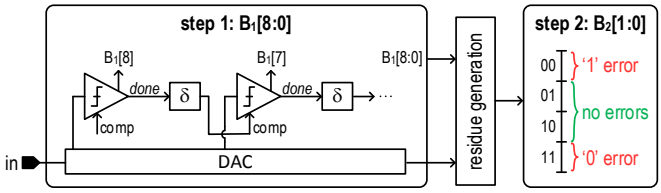


Fig. 5. Use of redundancy in a SAR-based 2-step ADC to detect errors of the comparators [10].

by monitoring the voltage $V_{x,n}$ at the virtual node input X of the closed-loop amplifier as shown in Fig. 4(a). By digitizing per sample n both this voltage $V_{x,n}$ and the output voltage $V_{o,n}$ of the amplifier, an rms estimation of the open-loop gain is derived:

$$\hat{A}_{OL} = \frac{\sum V_{o,n}^2}{\sum V_{o,n} V_{x,n}}, \tag{3}$$

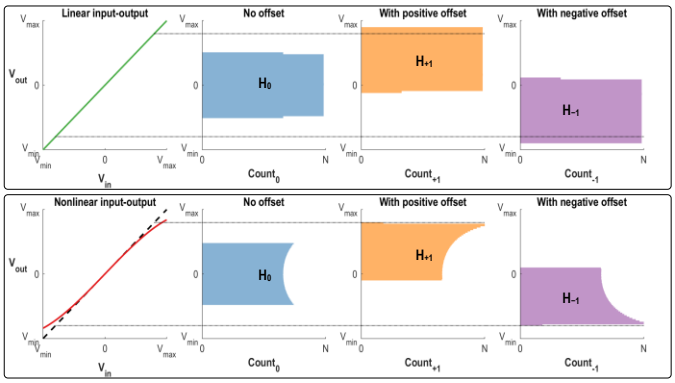
which can be used to compute other metrics, like the distortion in the ADC as its Signal-to-Distortion Ratio SDR:

$$\widehat{\mathrm{SDR}}_{\mathrm{dB}} = 10 \log_{10} \left[\frac{\beta^2 \mathrm{Var}(V_{o,n})}{\mathrm{Var}(V_{x,n} - V_{DC} - V_{o,n}/\hat{A}_{OL})} \right]. \tag{4}$$

This technique requires 2 auxiliary ADCs to measure the voltages directly. Whereas for $V_{o,n}$ the back-end of the ADC is readily used, for $V_{x,n}$ there is not an inherent solution available. A complete ADC for this purpose (e.g., a 13-bit ADC in [6]) is usually a cost too high. Since the error is usually small, a more elegant solution with a *stochastic ADC* [8] can be adopted.

This concept is shown in Fig. 4(b). The signal to be observed is sampled onto a small capacitor and a comparator is fired multiple times. Due to its noise (including the kT/C noise of the small sampling capacitor), the average output scales with the input voltage. An inverse error function and a scaling with the noise of the comparator can be used to get results in volts. Instead of getting multiple outputs from a single sampled input, values over multiple inputs can be combined based on the assumption that samples with the same output value $V_{o,n}$ will also have similar voltages $V_{x,n}$ [7]. As shown in Fig. 4(b), the input range of the stochastic ADC is limited to about $\pm 3\sigma_{\mathrm{noise}}$. It can be extended by using multiple comparators with built-in offsets. Such a stochastic ADC is also useful to monitor internal signals in the ADC when put in a "scope-on-chip" configuration [9]. By firing the comparator at multiple fixed time points, an estimation of the signal can be made in these time points and an internal waveform is revealed.

Fig. 5 shows an example of how errors can be detected using the redundancy in the system [10]. Here, a conversion happens in 2 steps with a 1-bit redundancy between the 2 steps. The first step is a SAR-like algorithm with a comparator-based controller using a different comparator per comparison. Offsets between the comparators results in nonlinearities. After the first phase, a residue is generated which is further digitized. When there are no errors, the second step produces codes that only fall within half of its input range. So, when an output falls outside its normal operation range, an error should have happened. When the error is on the upper side of the normal interval, the last comparator that returned a '1' made an error since the SAR algorithm produces then inputs for the subsequent comparators that are so negative that a '0' result is guaranteed from then onwards. Similarly, in case of a B_2 result in the lower half, the last comparator that returned a '0' is to be blamed.



NonLinearity = $\sum (H_0 \cdot | H_{+1} - H_0 | + H_0 \cdot | H_{-1} - H_0 |)$

Fig. 6. Measurement of the nonlinearity of a residue amplifier with a histogram-based detection technique [11].

Obviously, averaging is needed to account for errors due to for instance noise.

Histograms can also be used to derive nonlinearity errors in the ADC as illustrated in Fig. 6 [11]. For a 2-stage pipelined-SAR ADC, the nonlinearity of the residue amplifier is to be calibrated. To find a measure for the nonlinearity, an intentional positive or negative offset can be added to the input signal of the amplifier. Assuming a signal with a uniformly distributed amplitude at the input of the amplifier, the histogram of the converted output of the amplifier is flat with or without offset. However, as shown in the second row of Fig. 6, subjected to a nonlinear input-output amplifier characteristics, the histogram gets distorted differently depending on the sign of the applied offset. The equation in Fig. 6 then gives an estimation of the nonlinearity based on the number of points in the different bins of the histograms. With non-uniform inputs, only samples within a limited range should be used to obtain reliable nonlinearity estimation.

To calibrate the nonlinearity of the input buffer, the modified split-ADC technique of Fig. 7 can be adopted. The input is converted twice: once with the main ADC and once with an auxiliary ADC that operates on a 4x attenuated version of the input signal. Using a reduced input signal linearizes the auxiliary input buffer. Further, since main interests are the nonlinearities which are more prominent with large input signals, small input signals and noise are not a big concern, and the auxiliary buffer is realized as a scaled-down version of the normal input buffer to save power. The difference between the

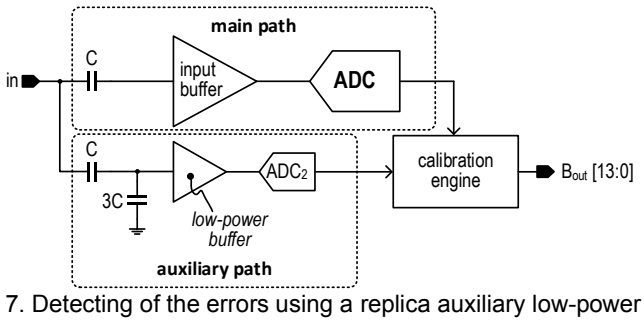


Fig. 7. Detecting of the errors using a replica auxiliary low-power ADC with reduced input signal [12].

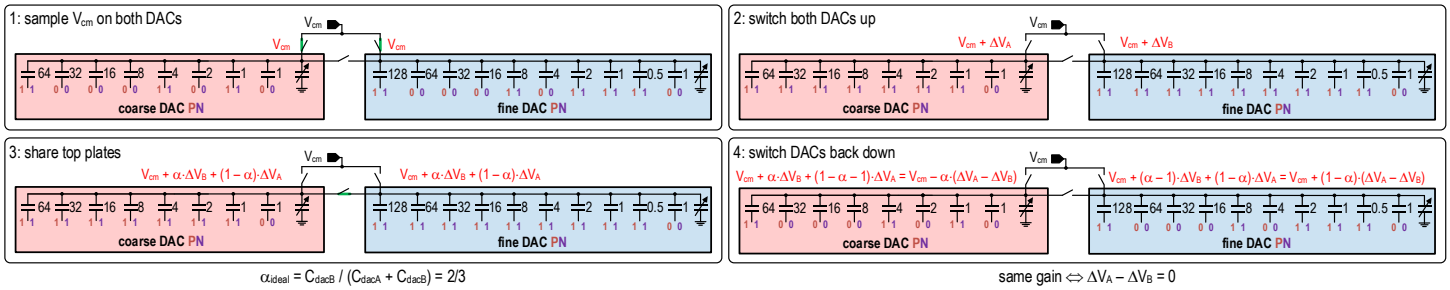


Fig. 8. Generation of the error signal with a zero input signal and internally generated reference signals [14].

outputs of the 2 paths for large input signals then gives a measure for the nonlinearity of the main path.

IV. Input signal

During the calibration, the input signal of the ADC can be completely known (e.g., a sine wave in a test environment), or completely unknown except for its range. Usually, however, at least some statistics of the input signal are known (e.g., for communication or biomedical signals) which can be exploited by the calibration technique. For instance, a known mean value simplifies offset calibration, and a known frequency range might help in correcting interleaving errors. Frequently, a dither signal is injected in the ADC [13] resulting in a combination of a known and unknown input signal.

The most well-known signal is the absence of any input signal. Parts of the ADC are then steered during a calibration sequence to generate an error signal that can be used by the calibration algorithm. An example is shown in Fig. 8 for the calibration of gain errors between two DACs in a SAR-based ADC [14] where the fine DAC is ideally twice the size of the coarse DAC. First, the common-mode voltage is sampled on both DACs with most of the bottom plates at a low voltage (0). Once sampled, the bottom plates of both DACs are switched up (1) to generate the maximum signal which ideally is the same for both DACs ($\Delta V_A = \Delta V_B$). Next, the top plates are shared via a special switch between the two DACs. Finally, the bottom plates are put back in the start position so that the top plates are now equal to V_{cm} plus an error voltage depending on the gain mismatch between the DACs. This signal on the top plate can then be used as input signal for the conversion. Since the error is expected to be small, the first comparator can be used in the configuration of a stochastic ADC (see Fig. 4(b)) to quantify the error.

Various errors can be detected by injecting a pseudorandom bit stream into the ADC to get a combination of an unknown and a known input signal. Such dither signal is uncorrelated with the actual input signal of the ADC and its autocorrelation function is a constant value. Fig. 9 shows different ways to inject this signal in the system. A straightforward way is to just add it in front of the ADC. Since this requires converting the digital dither signal to a continuous-time signal with possible kick-back to the ADC driver, it is usually preferred to add it after sampling via a DAC, especially when the errors of the input buffer are not the target of the calibration technique. For instance, with a SAR-like ADC, an extra unit is added to inject a positive or negative signal [15]. An alternative approach is shown in Fig. 9(c) [16]. Rather than changing the signal that the comparators see, the offset of the comparators can be adjusted instead. This technique avoids touching the sampled value at all, but it requires the availability of a comparator whose offset can be dynamically altered.

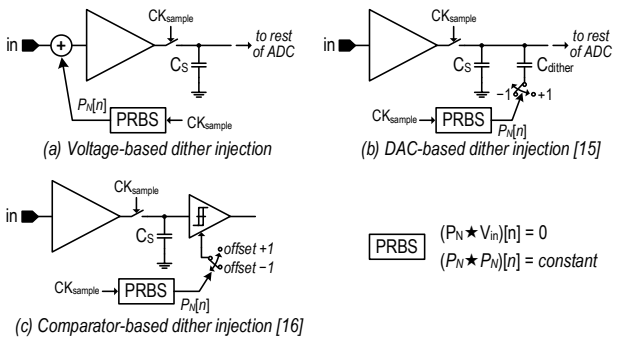


Fig. 9. Adding a pseudo-random bit stream (PRBS) dither signals to the ADC.

Both wideband and narrowband dither signals can be injected as shown in Fig. 10 [11]. In the first case, the signals which are much larger than the dither signal cause a large amount of noise in the error detection resulting in slow convergence algorithms. With the narrowband dither signal, the energy corresponding to the error is tracked more reliably and the algorithm that minimizes the error converges faster to the optimal value. Obviously, it requires extra bandpass filters. Knowledge of the signal frequencies helps in finding a free spot for the narrowband dither signal. Alternatively, the power in frequency bands can be monitored to find such a free spot.

Fig. 11 shows a digital implementation to suppress the offset and gain spurs in an interleaved ADC assuming the statistics of the input signal regarding mean and rms values are the same within each channel, i.e., after sub-sampling and frequency folding. the offset is corrected in the digital domain while the gain is tuned in the analog domain via a programmable capacitor bank [17]. For the offset, the difference between the average values is subtracted. The gain can be tuned in steps of 2^{-13} and its update value is set by the ratio of the powers of the signals in the different channels with respect to the reference channel 0. To simplify implementation in digital gates and avoid costly multipliers, the operations have been approximated.

V. Priority

The priority of the calibration technique determines whether the ADC can operate normally during calibration implying a background or online calibration, or whether it needs to be taken offline for a foreground calibration. Foreground calibration is suited for errors or correction mechanisms that don't fluctuate with temperature and that are not sensitive to local voltage supplies. Typical cases are capacitor mismatches and process variations. In other cases, it is often still desired to run a calibration sequence when starting up the ADC to find process-variable settings for the nominal temperature and supply. Background calibration then just needs to track for instance the usually slow temperature variations.

For instance, in a large system, a DAC might be available that can generate a sine wave at the input of the ADC during foreground calibration. In such case, the output of each channel $i = 1,...,IL$ can be fit onto a sine wave of a frequency of about f_0 obtaining least-square estimations of the amplitude A_i, phase φ_i, offset C_i and frequency deviation Δf [18]:

$$s_i(t) = \hat{A}_i \cos[2\pi(f_0 + \Delta f)t + \hat{\varphi}_i - 2(i-1)\pi/IL] + \hat{C}_i. \quad (5)$$

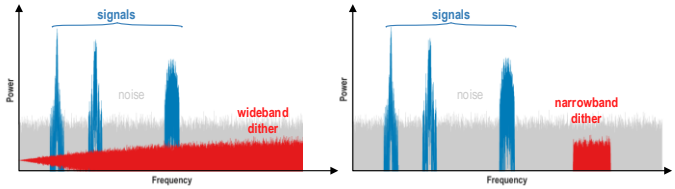


Fig. 10. Different spectra for wideband and narrowband dither injection [11].

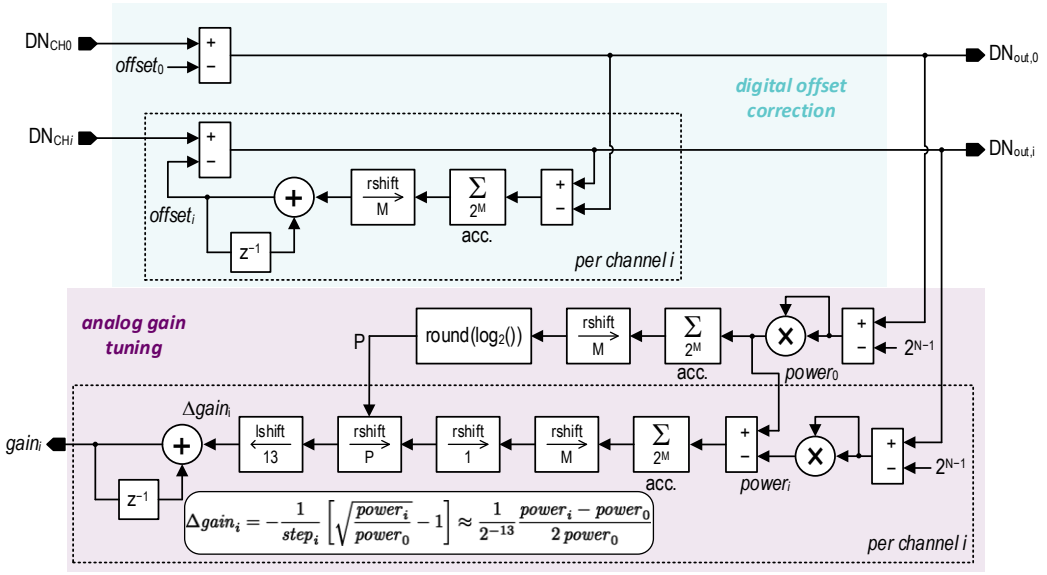


Fig. 11. Example of digital operations for calibration of the offset and gain mismatch between channels of interleaved ADC.

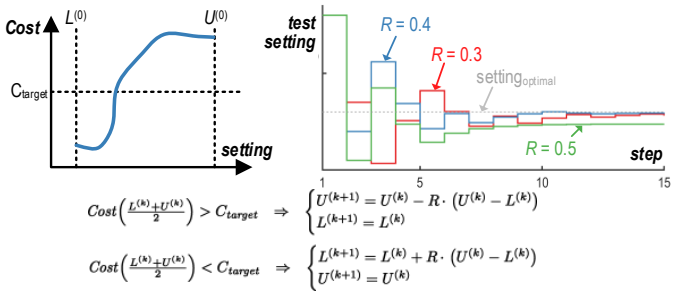


Fig. 12. Cost function and (sub-)binary search updates and curve.

All channels sample a signal of the same frequency, hence the frequency deviation Δf should be common for all channels. It is non-zero when the exact frequency of either the sine wave or the sampling clock is not known. This exact value is needed, for example, to accurate estimate the time skew per channel:

$$\hat{t}_{skew,i} = (\varphi_i - \varphi_{ref})/(f_0 + \Delta f). \quad (6)$$

Using for instance a binary search (see Fig. 12), the optimal setting is then easily found within the available settings on the range [$L^{(0)}$,$U^{(0)}$]. With R = 0.5, the search is binary but due to errors like noise or non-monotonicity, the final result can deviate from the optimal setting. With a reduction factor R < 0.5, some redundancy is built into the calibration leading to more accurate results at the cost of extra steps in the process as shown in the right of Fig. 12.

Rather than taking the ADC completely offline, it is often sufficient to allocate some time for extracting errors during the normal operation. For example, after the conversion, the offset of a comparator could be evaluated. Another example is shown in Fig. 13 [17]. In the first stage of a pipelined-SAR ADC, the conversion is done with a 6-bit coarse quantizer and the residue is generated by switching a high-accuracy DAC with uses as reference a charge reservoir C_{ref}. The output of the DAC is then amplified to the second stage. The charge consumed by these operations result in a signal-dependent drop of *vref*. To linearize the ADC, this drop is equalized over codes by adding the appropriate amount of extra capacitance C_{aux} with initial zero charge to the reference node. After the amplification, time is allocated to fire a comparator that compares the final value *vref* with the target value. This information is then averaged per SAR code and after collecting a fixed number of samples, the setting for C_{aux} per code is updated by an increment or decrement of the setting. As shown in Fig. 13, the time it takes to reach the optimal value starting from a common zero value depends on the SAR code. If not desired, the start value per curve can be set by taking the ADC completely offline resulting in a mixed foreground/background technique. When

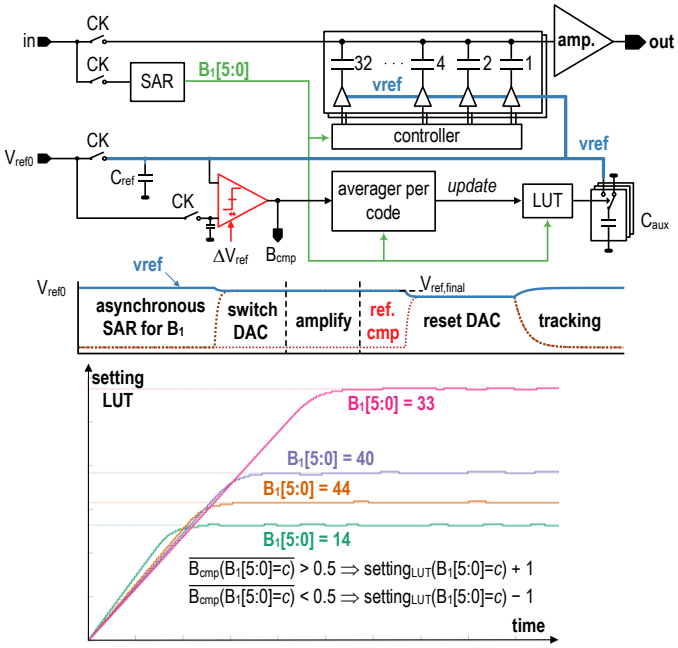


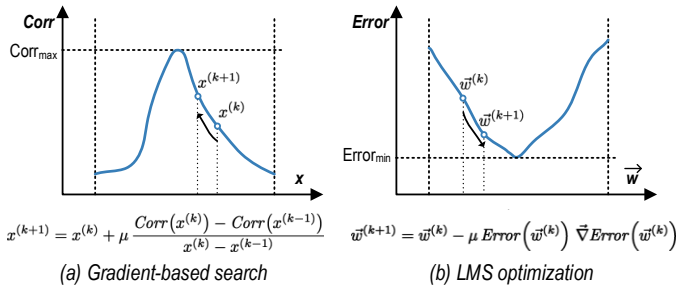
Fig. 13. Extraction of error on the reference voltage per SAR code and correction via increment/decrement background tracking [17].

the optimal value is reached, it keeps on toggling around the optimal setting. This simple procedure only works well when the mapping of the setting onto the tuned value (the value of C_{aux} in Fig. 13) is a monotonic curve: otherwise, it might get stuck in a local optimum.

Examples of more advanced techniques are shown in Fig. 14. In the first one [19], the gradient is approximated using the last 2 points, and the direction of the steepest ascent is used to maximize the cost function. This cost function, for instance, is the correlation between the outputs of 2 channels of an interleaved ADC and *x* corresponds to the skew correction between them.

A general illustration of the popular LMS algorithm is shown in Fig. 14(b). The current setting is updated by subtracting the product of the error and its gradient from it to minimize the expected value of the least mean square error. For example, the error can be the difference between the output value and a reference (e.g., a reference channel) with the output value of the ADC just a linear function of the output bits B_j so that the gradient of the error is readily computed [20]:

$$Error = \sum w_j^{(k)} B_j - \sum w_{j,ref} B_{j,ref}. \quad (7)$$



(a) Gradient-based search (b) LMS optimization

Fig. 14. Maximizing correlation via a gradient-based search (left) and minimizing error via an LMS optimization (right).

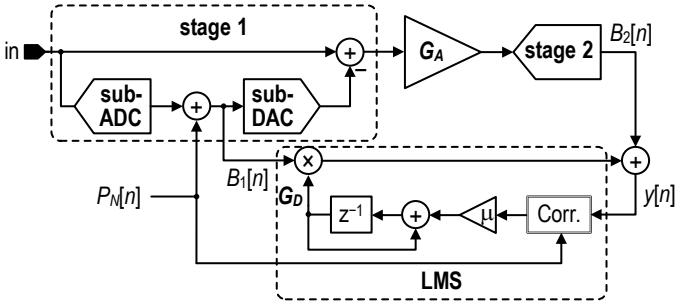


Fig. 15. Example of calibration of the gain of a residue amplifier in a 2-stage pipelined ADC using LMS optimization [11].

The expression for the error quickly becomes more complicated and approximation of the gradient is needed. For example, the skew error of a channel in an interleaved ADC can be estimated based on the samples in the previous and next channels [21]:

$$Error = \left| \frac{1}{\#points} \sum \left(|x_k[n] - x_{prev}[n]| - |x_{next}[n] - x_k[n]| \right) \right|, \quad (8)$$

and the LMS update equation then becomes

$$w^{(k+1)} = w^{(k)} - \mu \, Error^{(k)} \frac{\text{sign}(Error^{(k)} - Error^{(k-1)})}{\text{sign}(w^{(k)} - w^{(k-1)})}. \quad (9)$$

Another application is the calibration of the gain of a 2-stage pipelined ADC [11]. With a dither signal $P_N[n]$ injected into the first stage, the outputs of the 2 stages can be combined:

$$y[n] = a\, v_{in}[n] - G_A\,(B_1[n] + P_N[n]) + G_D\,(B_1[n] + P_N[n]), \quad (10)$$

and the error for the calibration process can be defined as the correlation between the output and dither signals:

$$Error = (y \star P_N)[n]. \quad (11)$$

Since the correlation is a linear operation and the dither signal is uncorrelated with the signals $v_{in}[n]$ and $B_1[n]$ (see Fig. 9), the gradient is just constant and the signal flow graph of the calibration technique becomes as shown in Fig. 15.

VI. Conversion impact

Ideally, the background calibration only improves the accuracy of the analog-to-digital conversion. However, there is often some impact on other aspects, most notably the conversion speed and its input range, both by the error detection and correction mechanisms. For example, after the actual conversion some time can be allocated to detect an error like the offset of comparators as shown in Fig. 16 [22], or some other internal voltage like the reference voltage (see Fig. 13). This detection then reduces the time available for conversion.

The correction mechanism can also decrease the conversion speed, usually by adding load and making the building blocks slower. In Fig. 17, different techniques are shown to compensate offset of a comparator in the analog domain with different impact on its speed.

(a) Programmable capacitors give an imbalance that counteracts the offset [23]. However, albeit beneficial for reducing the noise, the extra loading makes the comparator slower. Also, there might be a different loading for different offset compensation settings.

(b) An extra input pair injects a current opposite to the imbalance current caused by the mismatch of the main input pair [24]. The extra load is only that of the extra input pair, but an extra DAC is needed. The additional input pair also introduces additional mismatch.

(c) Technologies like FD-SOI allow to exploit the body-bias to compensate offset [25]. The impact on the speed of the comparator and hence the conversion speed is small.

(d) A split-source comparator [26] allows to program a voltage into the sources of the input pair to realize a threshold. Hence, it is readily re-used to compensate the offset of the input pair with no significant impact on the speed.

(e) Rather than changing anything in the comparator, the ADC architecture might offer an easy way to add a programmable offset. For example, in a SAR, extra units can be added to the sampling DAC which are switched before the conversion starts, or together with the MSB switching when some redundancy is available [27].

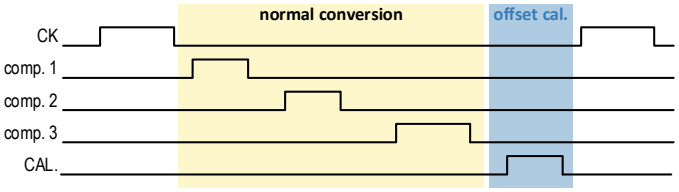
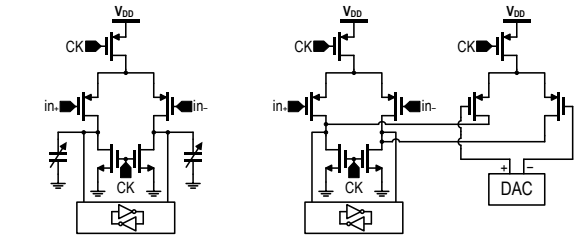


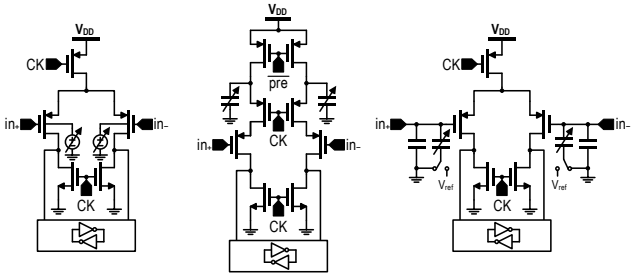
Fig. 16. Example of a typical timing diagram with conversion time sacrificed for comparator offset error detection [22].

The way the gain of the DAC in a SAR-based ADC is compensated can have an impact on the maximum input range of the ADC. For example, Fig. 18(a) shows a compensation of the DAC gain via programmable capacitive units. During the sampling process (CK = 1) their bottom plates are connected to a fixed voltage (ground or supply) while during conversion (CK = 0), they are either still connected to the fixed voltage (lowering the gain) or connected to each other (increasing the gain) with a bottom plate voltage almost in the middle between ground and supply, avoiding any risk of going above or below the supply or ground respectively [17]. However, it adds always extra load to the top plate resulting in a smaller input range of the ADC.

An alternative is depicted in Fig. 18(b). The reference voltage is filtered via an RC filter with small R and a DC current is sent over this resistor to induce a fix positive or negative voltage drop [27]. There is no extra loading of the top plate and hence no significant impact on the input range of the converter. The filter also helps in reducing high-frequency coupling between different channels of a time-interleaved ADC.

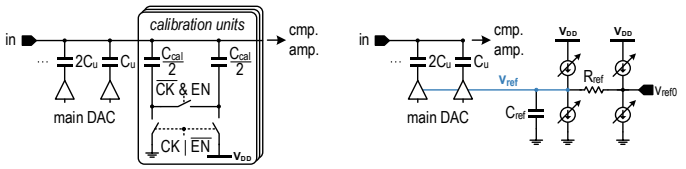


(a) Capacitive offset correction [23] (b) Correction via second input pair [24]



(c) Body-bias correction [25] (d) Source correction [26] (e) Correction via sampling DAC [27]

Fig. 17. Different ways to correct offset of a comparator with different impact on the speed of the comparator.



(a) Capacitive gain correction [17] (b) Reference gain correction [27]

Fig. 18. Different ways to correct gain of a DAC in SAR-based ADC with different impact on the input range.

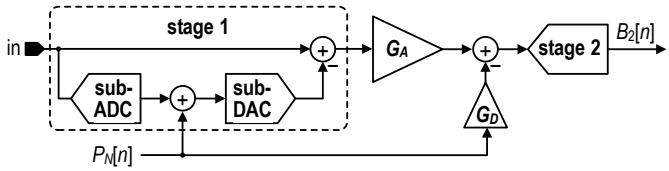


Fig. 19. Removing dither after the residue amplifier in a pipelined-SAR ADC [15].

Dither injected in the ADC (see Fig. 9) also consumes part of the input range and/or its robustness to errors by using part of the redundancy. An easy way to counteract this effect is to remove the dither again. Fig. 19 shows that in a pipelined-SAR ADC this can be done right after the amplification by switching the DAC in the second stage appropriately before the conversion in this stage starts [15].

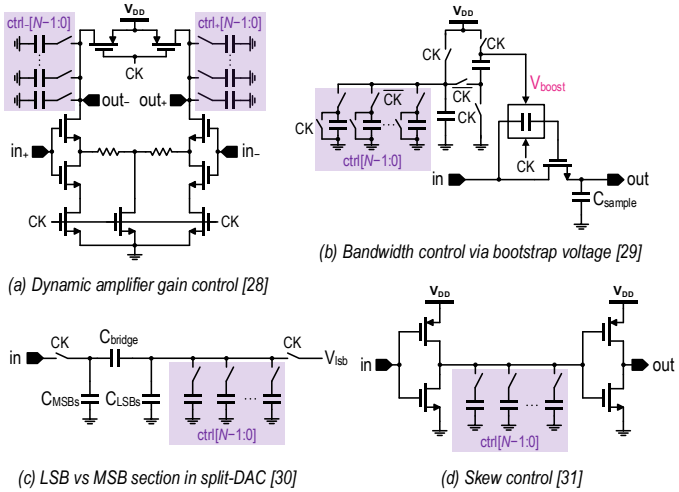
VII. Correction flow

Another dimension of a digital calibration technique is the way the correction happens. In a feedback system, the digital setting is translated to some analog quantity within the ADC resulting in a mixed analog-digital feedback system. On the other hand, the raw digital bits could just be corrected in a feedforwarding digital post-processing fashion.

Examples of feedback for comparators and DACs are shown in Fig. 17 and 18, respectively. Capacitor banks are a popular way to provide the feedback from the digital to the analog domain. Fig. 20 shows some examples for different ADC building blocks:

(a) a dynamic residue amplifier [28] whose gain is set by $g_m T_{amp}/C_L$, so tuning C_L allows to control the gain;
(b) a bootstrapped sampling circuitry whose bandwidth is controlled by tuning the bootstrap voltage V_{boost} [29] via a charge pump programmed via a capacitor bank;
(c) a DAC with an MSB and LSB section whose mismatch between the 2 sections due to an incorrect value of C_{bridge} is tuned via a capacitor bank on the LSB section [30];
(d) a conventional delay line used to tune skew mismatches in a time-interleaved ADC [31].

Complete digital correction in post-processing is usual an attractive solution as the internals of the ADC don't need to be changed.



(a) Dynamic amplifier gain control [28] (b) Bandwidth control via bootstrap voltage [29]

(c) LSB vs MSB section in split-DAC [30] (d) Skew control [31]

Fig. 20. Mixed analog/digital feedback via capacitor banks in different building blocks of high-speed ADCs.

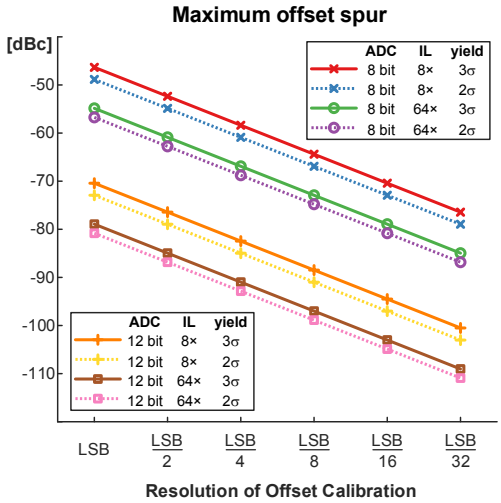


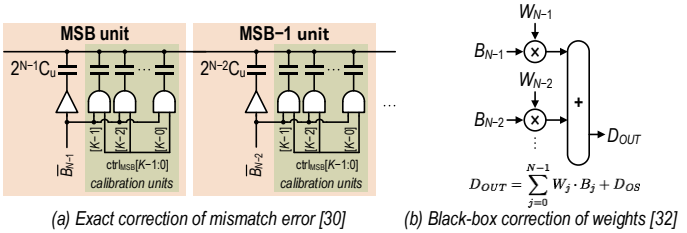
Fig. 21. Effect of the resolution of digital correction on the level of the maximum offset spur in an interleaved ADC.

However, even for a rather simple problem like the offset mismatch between channels, also the resolution of the digital correction has an impact on the overall ADC performance. In Fig. 21, for instance, the maximum spur level due to offset is plotted versus the resolution of the digital offset compensation for different interleaved ADCs (8x and 64x interleaved, with 8- and 12-bit resolution, and with a 2σ or 3σ yield). When high SFDR is required, and the ADC has only a small number of channels, a basic offset compensation with a resolution of 1 LSB is usually not sufficient. The more channels, the less likely all errors add up to cause a large spur, but there are obviously more spurs. Correction via analog-digital feedback makes achieving lower resolution usually easier: the capacitors of the calibration bank, for instance, can just be made smaller, or the effective capacitances are reduced by connecting them via a bridge capacitor.

VIII. Correction type

The correction applied by the calibration technique usually has an exact link with the cause of the error detected in the ADC. For example, the mismatch of the DAC in a SAR-like ADC result in nonlinearities in the input-output transfer function and these errors can be corrected by adding extra small calibration capacitors to the units that toggle together with the main units [29] as shown in Fig. 22(a). Note that since only capacitance can be *added*, the main unit needs to be designed with a systematic smaller value to leave some room for the calibration.

An alternative approach is a black-box correction type where no prior knowledge of the cause of the error is needed. For DAC capacitor mismatch, for instance, the outputs are corrected digitally by adjusting the different bit weights (see Fig. 22(b) [32]). This technique allows to correct any kind of error in the weights, not only due to the mismatch of the capacitors, but also due to, e.g., gain errors or settling errors. Hence, it comes with lots of flexibility. However, care must be taken to not overcorrect errors with weights that are, for instance, specific to the signals used when determining them. Hence, some background tracking mechanism is indispensable. Indeed, whereas matching errors of capacitors are independent of voltage and temperature variations, errors like settling errors do vary and hence the black-box weights need to be updated accordingly.



(a) Exact correction of mismatch error [30] (b) Black-box correction of weights [32]

Fig. 22. Correction of errors due to capacitor mismatch using exact correction and using a black-box approach.

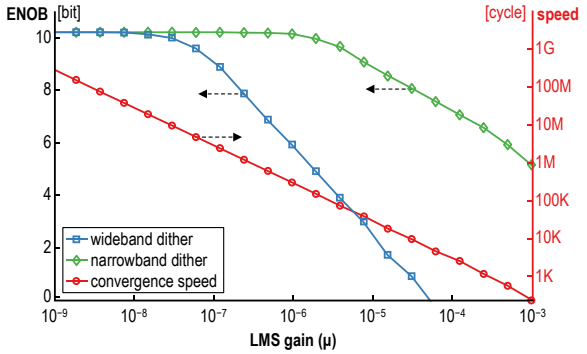


Fig. 23. Convergence speed of LMS gain calibration with wideband and narrowband dither [11].

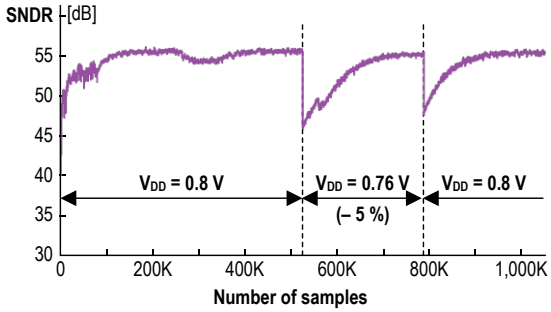


Fig. 24. Tracking changes in supply voltage [10].

IX. Speed

For the sub-binary search of Fig. 12, the maximum number of steps to complete the algorithm is given by

$$\#steps = -\frac{\log_2(\#settings)}{\log_2(1-R)}. \quad (12)$$

So, when such an algorithm is used, the number of steps required to finish the calibration is usually small. However, other aspects play a role in the total time required to complete the calibration of an ADC. First, the more noise there is in the system, the more averaging is required for accurate estimations of the errors. Second, parallelism allows to reduce the total calibration time significantly. For example, in a SAR-like ADC with comparator-based controller and hence a different comparator per comparison, each sample of the ADC ideally returns during for instance offset calibration relevant bits for all comparators instead of from only one [17]. When there are *IL* channels in a time-interleaved ADC, parallelism reduces the time accordingly. For example, instead of estimating the skew error using (8) sequentially, it is more time-efficient to parallelize it [33].

The convergence speed of the gradient- and LMS-based algorithms shown in Fig. 14 depends on the value of μ. A small value gives slow convergence while a large value results in fast convergence at the cost of often a sub-optimal final value. This is shown in Fig. 23 which corresponds to the calibration of the gain of the residue amplifier in a 2-stage pipelined-SAR ADC [11] using the LMS algorithm of Fig. 15 with the different dither injections of Fig. 10. Due to the better tracking of the error with the wideband dither, it converges faster to the optimal value. Also, the trade-off between the speed and accuracy for different values of μ is plotted. When there are multiple parameters to optimize like multiple distortion coefficients, it is beneficial to use a multi-layer LMS implementation [12] where the parameters corresponding to smaller errors (like the higher-order coefficients) are only optimized in later phases of the calibration.

Often, the calibration routine needs only to be able to track changes fast while starting up from a random initial could be slower. An example is shown in Fig. 24 for the redundancy-based calibration of Fig. 5 in a ping-pong SAR ADC [10]. After the initial calibration, the supply voltage is dropped with 5% after which the SNDR is restored relatively quickly, and again when the supply voltage is increased again to its original value.

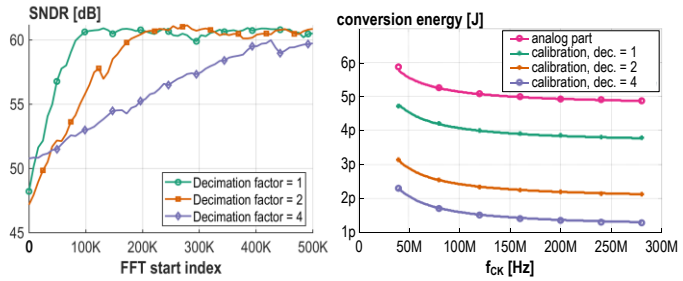


Fig. 25. Example of impact of decimation factor on the speed and power of the background calibration [15].

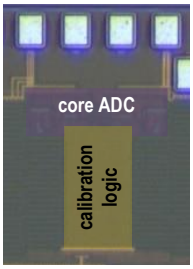
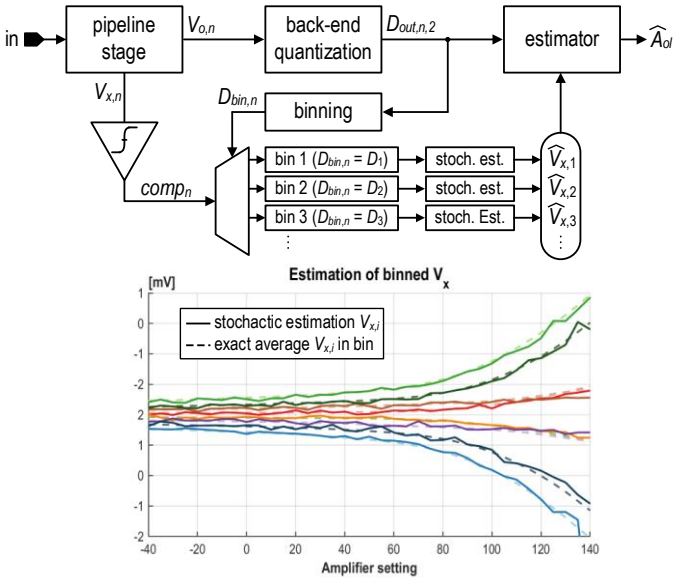


Fig. 26. Example of relative area of 2x TI pipelined-SAR ADC with on-chip foreground and background calibration logic [15].



X. Power

Three key contributors can be distinguished regarding the power of a digital calibration technique. First, the power overhead to detect the error during the conversion. Second, the energy required to compute the correction. Finally, the actual application of this correction. Since the correction needs to be applied for all samples, it might result in a non-negligible contribution, especially with digital correction for high-speed ADCs [13].

Decimation of both the error detection and correction update is an easy way to limit the power consumption, especially when only tracking variations is desired. Other aspects like the convergence speed are then sacrificed as shown in Fig. 25 [15]. Here, a time-interleaved pipelined-SAR is calibrated in the background which runs at a decimated clock to save power (less convergence energy) at the cost of a slower convergence (higher FFT start index to reach optimal performance).

XI. Area

The area occupied by the calibration logic could be rather significant. An example is shown in Fig. 26 [15] where the digital calibration logic in a 28nm technology takes up more area than the core ADC. To reduce the area, parts of it could be shared at the cost of lower calibration speeds. When the ADC is part of a large system or in a test environment, a general processor might be available so that the

calibration could run for a large part off-chip, especially when the correction flow is mainly a feedforward process in the digital domain.

Approximation of the operations as shown in Fig. 11 also reduces the required area. Operations per ADC code (e.g., a look-up table as in [17]) can be made more area-efficient by *binning* techniques. An example is shown in Fig. 27 where the open loop gain A_{ol} of an amplifier is to be estimated using (3) [7]. Instead of keeping the $V_{x,n}$ values per output code, the output codes are put in 8 bins spread across the range. The mean value of the $V_{x,n}$ values per bin is then estimated using the stochastic ADC of Fig. 4. As shown by the simulation results at the bottom of Fig. 27, this approach yields a good estimation of the exact value which can be used to simplify the logic and hence the area to estimate A_{ol}.

XII. Conclusions

Calibration is a popular technique to enhance the performance of ADCs. Various dimensions determine the success of the different algorithms. The type of errors to correct, the detection mechanism of these errors and the correction flow and type can have a significant impact on the conversion properties like speed and range. Techniques suited for random input signals and running continuously in the background give lots of flexibility, but they come at the cost of complexity translated in an increased power and area of the calibration algorithm itself. Different algorithms have been developed to increase the convergence speed of the calibration in a trade-off with one of the other dimensions. Careful design of the calibration technique with attention to all these dimensions result in a very efficient and powerful way to advance the state-of-the-art of ADCs.

References:

[1] B. Murmann, "Digitally Assisted Data Converted Design," *European Conf. Solid-State Circuits (ESSCIRC)*, Sep. 2013.

[2] B. Verbruggen, "Digitally Assisted Analog to Digital Converters," *Advances in Analog Circuit Design Workshop (AACD)*, Apr. 2014.

[3] B. Razavi, "Design Considerations for Interleaved ADCs," in *IEEE J. Solid-State Circuits*, Aug. 2013.

[4] M. Straayer *et al.*, "A 4GS/s Time-Interleaved RF ADC in 65nm CMOS with 4GHz Bandwidth," *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2016.

[5] J. McNeill *et al.*, ""Split ADC" Architecture for Deterministic Digital Background Calibration of a 16-bit 1-MS/s ADC," in *IEEE J. Solid-State Circuits*, Dec. 2005.

[6] A. M. A. Ali *et al*., "A 16-bit 250-MS/s IF Sampling Pipelined ADC with Background Calibration," in *IEEE J. Solid-State Circuits*, Dec. 2010.

[7] B. Hershberg *et al*., "A 4-GS/s 10-ENOB 75-mW Ringamp ADC in 16-nm CMOS With Background Monitoring of Distortion," in *IEEE J. Solid-State Circuits*, Aug. 2021.

[8] B. Verbruggen *et al.*, "A 60 dB SNDR 35 MS/s SAR ADC With Comparator-Noise-Based Stochastic Residue Estimation," in *IEEE J. Solid-State Circuits*, Sep. 2015.

[9] B. Hershberg *et al.*, "A 1-MS/s to 1GS/s Ringamp-Based Pipelined ADC With Fully Dynamic Reference Regulation and Stochastic Scope-on-Chip Background Monitoring in 16 nm," in *IEEE J. Solid-State Circuits*, Apr. 2021.

[10] K. Bunsen *et al.*, "A Redundancy-Based Background Calibration for Comparator Offset/Threshold and DAC Gain in a Ping-Pong SAR ADC," in *IEEE Trans. Circuits and Systems–II*, Feb. 2021.

[11] J. Lagos *et al.*, "A 10.1-ENOB, 6.2-fJ/conv.-step, 500-MS/s, Ringamp-Based Pipelined-SAR ADC With Background Calibration and Dynamic Reference Regulation in 16-nm CMOS," in *IEEE J. Solid-State Circuits*, Apr. 2022.

[12] L. Wei *et al.*, "An Auxiliary-Channel-Sharing Background Distortion and Gain Calibration Achieving >8dB SFDR Improvement over 4th Nyquist Zone in 1GS/s ADC," *IEEE Symp. VLSI Circuits*, Jun. 2021.

[13] A. M. A. Ali *et al.*, "A 12-b 18-GS/s RF Sampling ADC With an Integrated Wideband Track-and-Hold Amplifier and Background Calibration," in *IEEE J. Solid-State Circuits*, Dec. 2020.

[14] D. Dermit *et al.*, "A 1.67-GSps TI 10-Bit Ping-Pong SAR ADC With 51-dB SNDR in 16-nm FinFET," in *IEEE Solid-State Circuit Letters*, Mar. 2020.

[15] B. Verbruggen *et al.*, "A 2.1 mW 11b 410 MS/s Dynamic Pipelined SAR ADC with Background Calibration in 28nm Digital CMOS," *IEEE Symp. VLSI Circuits*, Jun. 2013.

[16] J. Li *et al.*, "Background Calibration Techniques for Multistage Pipelined ADCs With Digital Redundancy," in *IEEE Trans. Circuits and Systems–II*, Sep. 2003.

[17] E. Martens *et al.*, "A 69-dB SNDR 300-MS/s Two-Time Interleaved Pipelined SAR ADC in 16-m CMOS FinFET With Capacitive Reference Stabilization," in *IEEE J. Solid-State Circuits*, Apr. 2018.

[18] N. Giaquinto *et al.*, "Fast and Accurate ADC Testing Via an Enhance Sine Wave Fitting Algorithm," in *IEEE Trans. Instrumentation and Measurement*, Aug. 1997.

[19] M. El-Chammas *et al.*, "A 12-GS/s 81-mW 5-bit Time-Interleaved Flash ADC With Background Timing Skew Calibration," in *IEEE J. Solid-State Circuits*, Apr. 2011.

[20] D. Stepanović *et al.*, "A 2.8 GS/s 44.6 mW Time-Interleaved ADC Achieving 50.9 dB SNDR and 3 dB Effective Resolution Bandwidth of 1.5 GHz in 65 nm CMOS," in *IEEE J. Solid-State Circuits*, Apr. 2013.

[21] S. Chen *et al*., "All-Digital Calibration of Timing Mismatch Error in Time-Interleaved Analog-to-Digital Converts," in *IEEE Trans. VLSI Systems*, Sep. 2017.

[22] D. Li *et al*., "A 7b 2.6mW 900MS/s Nonbinary 2-then-3b/cycle SAR ADC with Background Offset Calibration," *IEEE Custom Integrated Circuits Conf. (CICC)*, Apr. 2019.

[23] V. Giannini *et al.*, "An 820μW 9b 40MS/s Noise-Tolerant Dynamic-SAR ADC in 90nm Digital CMOS," *IEEE Int. Solid-State Circuits Conf.*, Feb. 2008.

[24] Q. Fan *et al*., "A Time-Interleaved SAR ADC With Bypass-Based Opportunistic Adaptive Calibration," in *IEEE J. Solid-State Circuits*, Aug. 2020.

[25] E. Wittenhagen *et al.*, "Advanced Mixed Signal Concepts Exploiting the Strong Body-Bias Effect in CMOS 22FDX®," in *IEEE Trans. Circuits and Systems–I*, Jan. 2021.

[26] E. Martens *et al.*, "Wide-Tuning range programmable threshold comparator using capacitive source-voltage shifting," in *IET Electronics Letters*, Sep. 2018.

[27] E. Martens *et al.*, "A Compact 8-bit, 8GS/s 8xTI SAR ADC in 16nm with 45dB SNDR and 5GHz BW," *IEEE Symp. VLSI Circuits*, Jun. 2021.

[28] Z. Zheng *et al.*, "A 3.3-GS/s 6-b Fully Dynamic Pipelined ADC With Linearized Dynamic Amplifier," in *IEEE J. Solid-State Circuits*, Jun. 2022.

[29] B. Verbruggen *et al.*, "A 70dB SNDR 200 MS/s 2.3 mW dynamic pipelined SAR ADC in 28nm digital CMOS," *IEEE Symp. VLSI Circuits*, Jun. 2014.

[30] Y. Chen *et al.*, "Split Capacitor DAC Mismatch Calibration in Successive Approximation ADC," *IEEE Custom Integrated Circuits Conf. (CICC)*, Apr. 2019.

[31] Y.-H. Chung *et al.*, "A 38-mW 7-bit 5-GS/s Time-Interleaved SAR ADC with Background Skew Calibration," *IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Nov. 2018.

[32] A. T. Ramkaj *et al.*, "A 5-GS/s 158.6-mW 9.4-ENOB Passive-Sampling Time-Interleaved Three-Stage Pipelined-SAR ADC With Analog-Digital Corrections in 28-nm CMOS," in *IEEE J. Solid-State Circuits*, Jun. 2020.

[33] G. Bè et al., "A 900-MS/s SAR-Based Time-Interleaved ADC With a Fully Programmable Interleaving Factor and On-Chip Scalable Background Calibrations," in *IEEE Trans. Circuits and Systems–II*, Sep. 2022.