A Generic Foreground Calibration Algorithm For ADCs with Nonlinear Impairments

Armia Salib, *Student Member, IEEE,* Mark F. Flanagan, *Senior Member, IEEE,* and Barry Cardiff, *Member, IEEE*

***Abstract*—This paper presents a generic foreground calibration algorithm which estimates and corrects memoryless nonlinear impairments in both single channel and time-interleaved analog-to-digital converters (TIADCs), and which is capable of correcting for amplifier nonlinearity, comparator offsets and capacitance mismatch for each channel. It operates by generating, and then using, a Look-Up Table which maps raw ADC output decision vectors to linearised output. For TIADCs, the algorithm also uses information gained during the calibration phase to estimate timing and gain mismatches among the sub-ADCs. The problem of selecting an appropriate timing reference so as to relax the requirements on the time skew correction circuitry is statistically analysed, as is the corresponding impact on manufacturing yield. Accordingly, a new method is proposed having superior performance; for example, in the case of an 8 sub-ADC TIADC system, the proposed scheme reduces the time skew correction requirement by 44% compared to conventional methods. The architecture is instrumented with some additional circuitry to facilitate built-in self-test (BIST), allowing manufacturing test time and cost reductions. Implementation aspects are discussed, and several complexity reduction techniques are presented along with synthesis results from a Verilog implementation of the calibration engine.**

***Index Terms*—ADC, transfer function, foreground, calibration, nonlinearity, TIADC, BIST, bandwidth mismatch.**

I. Introduction

MULTIPLE nonlinear impairments exist in ADCs, which can limit their performance. Many calibration techniques have been proposed to address some specific nonlinearity source via post-processing, e.g., [1]–[10]. However, since an ADC may suffer from multiple nonlinear effects simultaneously, it may be necessary to implement multiple calibration algorithms in a complete solution, which can increase the complexity; these can also introduce performance uncertainties, as the algorithms may have a deleterious impact upon each other.

Sample-and-hold (S/H) nonlinearity is an example of a nonlinear impairment that affects many ADC architectures. A model for the S/H input-output relationship is [1]

$$y_{\text{out}} = \sum_{i=1}^{\infty} c_i x_{\text{in}}^i, \tag{1}$$

where c_i are the nonlinearity coefficients. In [1], only c_1 and c_2 are estimated, and their effect is compensated through post-processing.

In SAR ADCs, capacitor mismatch is an example of an architecture-dependent impairment; it results from variations in the fabrication process [3]. For its calibration, [2]–[4] suggest using a linear combination, where the comparator decisions are linearly combined with weights proportional to the capacitance (called *ideal weights*). In [3] and [4], these weights are evaluated via histogram measurements.

The work in [11] targets nonlinearity calibration in pipeline ADCs. This calibration requires a sinusoidal input with curve fitting techniques employed to evaluate the error in each output sample. This error is used to update the linear combining weights via a least-mean-square (LMS) adaptation technique. However, to evaluate the input signal parameters accurately for curve fitting, a large number of samples needs to be processed; this requires a large memory size.

As with many calibration algorithms, the above examples compute a set of weights that are then used at run-time (often via a linear combination) to produce the eventual calibrated output. While these can be effective for some specific nonlinearity, a more flexible approach is to use a *look-Up Table* (LUT) which can, in principle, correct for any memoryless nonlinearity.

An LUT is used in [12] to map each ADC raw output to its associated integral nonlinearity (INL) error. Using a sinusoidal input, the INL errors manifest themselves as harmonic frequency components. In [12], the INL errors are evaluated by measuring the amplitude of those harmonic components; however, this requires matrix manipulations, which complicates the implementation for real-time operation.

The above techniques can be applied to correct for nonlinear effects within a single ADC or indeed within each of the sub-ADCs of a time-interleaved ADC (TIADC) architecture. However, there are mismatches among the sub-ADCs that contribute to an additional form of nonlinearity in the aggregated output. The main sources of these mismatches are offset, gain and time skew. These mismatches need to be estimated and compensated for, as they cause spurious components to occur in the aggregated output, thus limiting the overall performance.

As is the case for a single channel ADC, the calibration process for these TIADC-specific mismatches can be either background or foreground. In background calibration, the mismatches are estimated during normal operation by utilizing some hardware redundancy and/or by taking advantage of some assumed statistical signal properties.

This publication has emanated from research supported in part by S3group, Dublin, Ireland, and in part by a research grant from Science Foundation Ireland (SFI), and is co-funded under the European Regional Development Fund under Grant Number 13/RC/2077.

The authors are with the School of Electrical and Electronic Engineering, University College Dublin, Dublin, Ireland (e-mail: armia.salibfarag@ucdconnect.ie; mark.flanagan@ieee.org; barry.cardiff@ucd.ie).

In [5], the background estimation of time skew mismatches is performed by applying a low-complexity Fast Fourier Transform (FFT) on the output samples; however, this method still requires intensive computations and large memory sizes. The cross-correlation between the different sub-ADC outputs is used in [13]–[17], where the output of a bank of correlators is processed to obtain the time skew values.

Hardware redundancy is employed in [6] and [7], wherein an extra-slow reference ADC is utilized that periodically samples simultaneously with each of the sub-ADCs, enabling estimation of the error associated with that sub-ADC output. The correction side is adapted such that the measured error is minimized. A similar idea is proposed in [8], where an extra comparator is used instead, and an adaptation process is used to maximize the correlation between the comparator's output and that of the sub-ADC's. One problem with this approach is that, since the extra ADC/comparator does not sample every clock cycle, the sub-ADC is loaded differently when being calibrated, causing its behaviour to change during calibration and thus leading to inaccuracies. Also, the periodic changing of the input load can introduce spurs into the ADC output [9].

In [9] and [10], both additional hardware redundancy and signal statistics are utilized in time skew calibration. Here the authors aim to minimize the variance of the sub-ADCs' output corresponding to a certain input voltage window. An extra-fast flash ADC or comparator is used to mark the samples within that voltage window. However, the relationship between the variance and the time skew becomes very weak for a random input signal, which prevents accurate estimation. Also, these algorithms cannot decide the adaptation direction, and consequently continuous small changes are applied on the correction side, making the convergence speed very slow for high precision time skew correction. This problem also exists in the algorithm proposed in [8].

In general, background calibration algorithms offer the ability to track slow variations in voltage and temperature (PVT variations). However, their initial convergence can be slow and, since they are running continuously, they deplete the allowable power consumption budget. It is also possible that these algorithms may converge to incorrect values in cases where the signal statistical assumptions are not satisfied [16], making them unsuitable for general-purpose ADC applications.

For faster convergence and accurate estimation, foreground calibration algorithms may be considered. In these algorithms, a calibration time slot is allocated where a known input signal is applied to the TIADC system. After estimation, the ADC switches to the normal operation mode, where the samples are corrected using the estimated values previously obtained. In general, foreground calibration techniques cannot track PVT variations; however, this can be addressed by occasional recalibration.

A foreground calibration algorithm is used in [18] to estimate the time skew mismatch in a TIADC system, where a linear "ramp" or triangular input signal with known characteristics is injected into the TIADC system; however, in this case noise cannot be removed from the injected signal by filtering as this also affects the linearity of the triangular input [19]. Also, in [20] and [21] the time skew and gain mismatches are obtained through processing off-chip in the frequency domain the output samples of a known input signal.

The estimation of mismatches requires the existence of a reference. Conventionally, this reference can be selected to be

1) one of the main sub-ADCs, e.g., [13], [14], [17], [22], [23]; or
2) an extra sub-ADC or a comparator, e.g., [6]–[9], [18].

However, this pre-selection for the reference, especially for the time skew, leads to tightening of the requirements on the correction side, as we illustrate in Subsection IV-B.

The correction for the offset and gain mismatch is easily achieved digitally using addition and multiplication operations. However, the correction for the time skew mismatch can be done either by analog delay lines before each sub-ADC, e.g., [7], [14], or by digital interpolation after the sub-ADCs (using the aggregated output), e.g., [15], [17]. In the primary application considered in this paper, we use analog delay lines.

In this paper, a unified calibration technique is proposed which works in the existence of several sources of nonlinearity. A sinusoidal input with a known frequency is used during the calibration phase, where both the sampling clock and the input signal are generated using the same signal generator to guarantee their synchronization. Knowing the input signal frequency obviates the complexity of the curve fitting required in [11]. The algorithm populates a LUT which stores the most likely input corresponding to each raw output. The use of this LUT saves post-processing computational power and reduces latency, at the cost of additional storage.

In addition to calibrating the nonlinearity in each individual ADC, the proposed mechanism can be used to estimate the mismatch in TIADCs. The foreground time skew estimation enables proper selection for the reference timing that reduces the complexity on the correction side.

Also, the proposed calibration mechanism can be employed as a built-in self-test to evaluate the ADC performance. With minor extra circuitry, the SNDR for the ADC output can be evaluated without the need for processing the samples externally. This obviates the need to transfer a large amount of data to an external computation unit, which speeds up the production test procedure to be carried out for each fabricated chip, reducing the overall manufacturing cost.

This paper is organized as follows, Section II describes the proposed calibration algorithm for a single channel ADC, and its implementation is illustrated in Section III. Section IV extends the proposed algorithm to estimate the mismatches among the sub-ADCs of the TIADC architecture. In Section V, we study the behaviour of the proposed algorithm in the presence of multiple nonlinearity sources, and in particular with bandwidth mismatch. The exploitation of the algorithm as a built-in self-test is described in Section VI. Simulation results are presented in Section VII targeting both a single channel ADC and a TIADC system. Finally, conclusions are drawn in Section VIII.

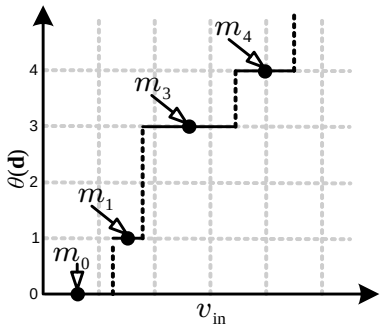

Fig. 1: An illustrative example of a transfer function, modified from [24].

II. Proposed calibration algorithm

In this algorithm, we consider a generic ADC that makes N binary decisions, which form its raw output. This raw output is denoted by an $N \times 1$ vector $\mathbf{d}$, where the n^{th} decision is denoted by d_n, and d_0 is the least significant decision. We define the following function to map any possible combination for $\mathbf{d}$ to a unique integer,

$$\theta(\mathbf{d}) \triangleq \sum_{n=0}^{N-1} d_n 2^n. \tag{2}$$

Without considering any impairments, the *nominal* ADC output can be calculated according to

$$y \triangleq \sum_{n=0}^{N-1} d_n w_n, \tag{3}$$

where w_n is the nominal weight for the n^{th} binary decision, e.g., $w_n = 2^n$ for radix-2 SAR ADCs. However, due to circuit impairments, the output of (3) can be an inaccurate representation of the analog input.

The ADC system can be modeled as a memoryless system whose current output is a function of its current input, and it can be described by a *transfer function*. In the absence of internal noise, the transfer function for an ADC between its analog input v_{in} and digital output $\theta(\mathbf{d})$ looks like a staircase as illustrated in Figure 1. For a given output $\mathbf{d}$, the analog input can be modeled as a sum of

1) a deterministic value, $m_{\theta(\mathbf{d})}$, that represents the average input corresponding to the raw ADC output $\mathbf{d}$ as shown in Figure 1; and
2) a random quantization noise having a uniform distribution with *zero mean*, and whose range can vary due to the existence of circuit impairments.

The objective of the algorithm is to estimate $m_{\theta(\mathbf{d})}$ for all $\mathbf{d}$. Figure 2 shows a block diagram for the *calibration engine* used in calibrating the nonlinearity of a single ADC channel, where the estimated values for $m_{\theta(\mathbf{d})}$ are populated in the LUT. During the calibration, the LUT output $\tilde{y}[k]$ is ignored, and the output $\tilde{\psi}$ can be used to estimate the mismatch among the sub-ADCs for TIADC systems as described in Section IV. After calibration, the ADC may switch to normal operation, where the LUT maps the raw ADC output at the k^{th} time

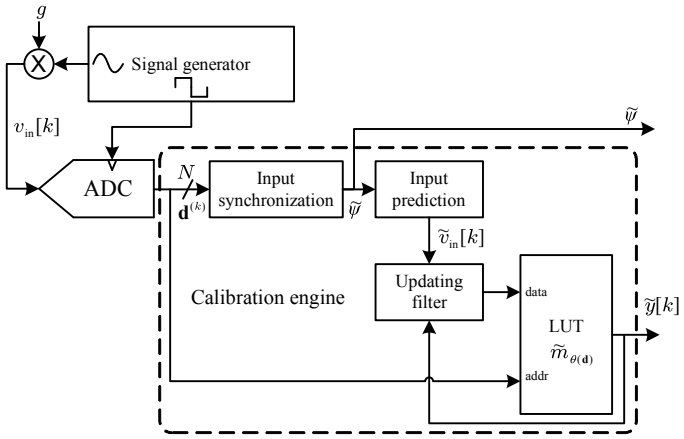

Fig. 2: Block diagram for the proposed calibration engine, modified from [24].

index, denoted by $\mathbf{d}^{(k)}$, to its corresponding calibrated output, $\tilde{y}[k]$, an estimate of $m_{\theta(\mathbf{d}^{(k)})}$.

The proposed calibration algorithm can be divided into two stages that may run in parallel:

1) *Input signal synchronization*: in this stage, the input signal is characterized in order to compute an approximate digital copy $\tilde{v}_{\text{in}}[k]$ of the analog input $v_{\text{in}}[k]$ at any time index k; this stage is outlined in Subsection II-A.
2) *Building the look-up table*: using the output of the first stage, the average input corresponding to each output level is evaluated; Subsection II-B describes this stage.

A. Input signal synchronization

In the proposed calibration algorithm, a sinusoidal input signal with a known frequency is used. The frequency is selected to be $\frac{a}{K} f_{\text{s}}$ where f_{s} is the sampling frequency, K is a power of 2, a is an integer such that a and K are relatively prime, and $0 < a < K/2$. This analog input can be described as follows,

$$v_{\text{in}}[k] = A\, g \cos\Big(\frac{2\pi a}{K} k + \phi_0\Big) \tag{4}$$

$$= \frac{A}{2} g \Big(e^{j(\frac{2\pi a}{K} k + \phi_0)} + e^{-j(\frac{2\pi a}{K} k + \phi_0)} \Big), \tag{5}$$

where A is the input amplitude, ϕ_0 is the initial phase and g is the ADC internal gain used to convert the input voltage into a digital output. The nominal value for g is denoted by $\bar{g}$, e.g., $\bar{g} = 1024/V_{\text{swing}}$ for a 10-bit ADC, where V_{swing} is the input voltage swing. The input amplitude A is configured such that the test signal covers most of the input swing, which increases the number of raw output levels exercised during the calibration. In order to create a digital copy of this analog input, both its initial phase and amplitude need to be evaluated.

Let ψ denote the result of the following average over K samples,

$$\psi = \frac{1}{K} \sum_{k=0}^{K-1} e^{-j\frac{2\pi a}{K} k} v_{\text{in}}[k] \tag{6}$$

$$= \frac{A}{2} g\, e^{j\phi_0}, \tag{7}$$

where (7) holds because the averaging covers an integer number of cycles of the sinusoidal input. The magnitude and the phase of ψ contain information about the input signal amplitude and initial phase respectively, which facilitates the input signal prediction at any time instant. Note that (6) resembles the calculations needed for the a^{th} component of the Discrete Fourier Transform (DFT) of size K.

However, (6) may not be used directly to find ψ, since we do not know $v_{\text{in}}[k]$. Therefore, we substitute $v_{\text{in}}[k]$ with the output of the nominal combining in (3); the corresponding estimate of ψ may then be written as

$$\psi \approx \tilde{\psi} \triangleq \frac{1}{K}\sum_{k=0}^{K-1}\Big(e^{-j\frac{2\pi a}{K}k}\underbrace{\sum_{n=0}^{N-1} w_n d_n^{(k)}}_{\text{nominal combining}}\Big). \tag{8}$$

The averaging in (8) relaxes the impact of both the noise and nonlinear impairments, which makes $\tilde{\psi}$ an accurate approximation to ψ. Also, this averaging disregards any offset in the output of (3).

B. Building the look-up table

Using (4), (7) and the output of (8), the analog input can be predicted at any time index k according to

$$v_{\text{in}}[k] \approx \tilde{v}_{\text{in}}[k] = 2\Re\big(\tilde{\psi}e^{j\frac{2\pi a}{K}k}\big), \tag{9}$$

where $\Re(X)$ denotes the real part of a complex number X. Note that for the input signal used, $\tilde{v}_{\text{in}}[k] = \tilde{v}_{\text{in}}[k \pmod K]$.

After obtaining $\tilde{\psi}$ from (8), another F samples are observed, indexed by k where $0 \le k < F$. These samples are used to measure the average predicted input $\tilde{v}_{\text{in}}$ corresponding to each raw output $\mathbf{d}$. We define a set $\mathcal{S}^{(\mathbf{d})}$ for *each* possible $\mathbf{d}$, containing the time indices when the observed ADC output happens to be $\mathbf{d}$, i.e., $\mathcal{S}^{(\mathbf{d})} = \{k \mid \mathbf{d}^{(k)} = \mathbf{d}, 0 \le k < F\}$. Using this set, $\bar{m}_{\theta(\mathbf{d})}$, an estimate of $m_{\theta(\mathbf{d})}$, can be evaluated according to

$$\bar{m}_{\theta(\mathbf{d})} \triangleq \frac{\sum_{s\in\mathcal{S}^{(\mathbf{d})}} \tilde{v}_{\text{in}}[s]}{|\mathcal{S}^{(\mathbf{d})}|}, \tag{10}$$

where $|\mathcal{S}^{(\mathbf{d})}|$ is the cardinality of $\mathcal{S}^{(\mathbf{d})}$.

III. Implementation

A suitable hardware realization for the input synchronization block (8) and the prediction block (9) is depicted in Figure 3, where each of the two depicted multipliers can be implemented as half a *complex* multiplier. A coordinate rotation digital computer (CORDIC) is used to calculate $\beta[k] = e^{j\frac{2\pi a}{K}k}$ which is a common term in both (8) and (9).

The averaging in (8) produces an updated $\tilde{\psi}$ every K samples, allowing tracking of any small changes to the input. After obtaining the first $\tilde{\psi}$, both stages (described in Subsections II-A and II-B) can run in parallel to estimate the input voltage $\tilde{v}_{\text{in}}[k]$, which is used to update the LUT content as shown in Figure 4.

The LUT contains 2^N entries each with width $N+b$, where b is the bit-width allocated for the fractional part. The i^{th} entry

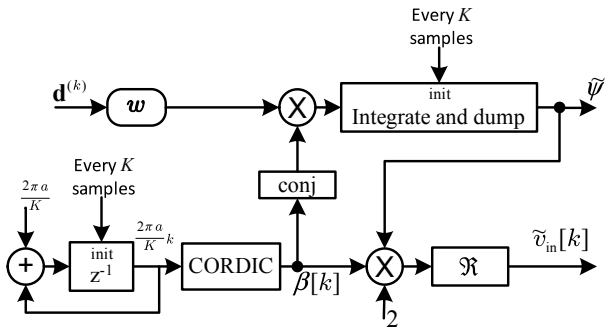

Fig. 3: Suggested implementation for (8) and (9), modified from [24].

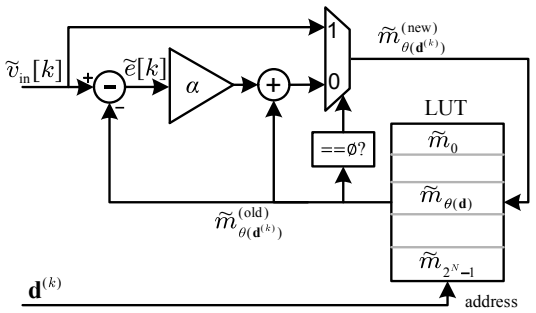

Fig. 4: LMS adapting filter used for the updating the LUT entries, modified from [24].

in the LUT stores $\tilde{m}_i$, an estimate for $\bar{m}_{\theta(\mathbf{d})}|_{\theta(\mathbf{d})=i}$. At the start of the calibration process, all entries are initialized to an invalid value $\varnothing$, this is in order to mark any non-updated entry during the calibration run.

For the k^{th} input in the second calibration stage, the difference between $\tilde{v}_{\text{in}}[k]$ and the LUT entry content $\tilde{m}_{\theta(\mathbf{d}^{(k)})}$ can be evaluated,

$$\tilde{e}[k] = \tilde{m}_{\theta(\mathbf{d}^{(k)})} - \tilde{v}_{\text{in}}[k], \tag{11}$$

where $\tilde{e}[k]$ resembles the instantaneous error associated with the content of the LUT entry with index $\theta(\mathbf{d}^{(k)})$. Using LMS adaptation, this entry can be updated according to:

$$\tilde{m}_{\theta(\mathbf{d}^{(k)})} \mapsfrom \begin{cases} \tilde{v}_{\text{in}}[k], & \text{when } \tilde{m}_{\theta(\mathbf{d}^{(k)})} = \varnothing \\ \tilde{m}_{\theta(\mathbf{d}^{(k)})} - \alpha\tilde{e}[k], & \text{otherwise,} \end{cases} \tag{12}$$

where $0 < \alpha < 1$ is the adaptation step size. The special case when $\tilde{m}_{\theta(\mathbf{d}^{(k)})} = \varnothing$ in (12) is used to enhance the convergence speed, and it occurs only once for each entry. This adaptation process acts as an averaging process similar to (10), but it is suitable for run-time implementation. Note that the adaptation process for each entry in the LUT is independent.

Upon the completion of the second calibration stage, there may exist LUT entries which have not been updated, i.e., they remain $\varnothing$. This can happen because only a limited number of samples is observed. Those entries are filled using linear interpolation with the aid of the nearest updated entries.

IV. Mismatch calibration in TIADC

The description in the previous sections covers the calibration needed for a single channel ADC. In this section, we

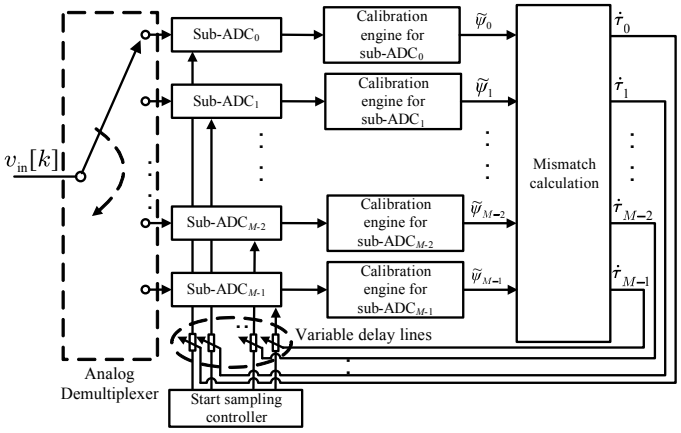

Fig. 5: TIADC mismatch calibration block diagram.

extend the application of this algorithm to support mismatch estimation in a TIADC that consists of M sub-ADCs as per Figure 5. The m^{th} sub-ADC ($0 \leq m \leq M-1$) is equipped with a variable delay line whose configuration is denoted by $\dot{\tau}_m$.

The proposed calibration algorithm in Section II is sufficient to neutralize the effect of offset mismatch since $\tilde{v}_{\text{in}}$ has zero mean; however, other types of mismatch remain and require estimation.

A. Mismatch estimation in TIADC system

Figure 5 shows the block diagram for the proposed mismatch calibration technique, where M calibration engines are running in parallel. Each engine is connected to the output of one of the sub-ADCs; these engines share the same CORDIC block and initialization controller. The output of the 'Input synchronization' block inside the m^{th} calibration engine is denoted by $\tilde{\psi}_m$.

Considering the gain and time skew mismatch in the m^{th} sub-ADC, we can rewrite (4) as

$$v_{\text{in}}^{(m)}[k] = A\, g_m \cos\Big(\frac{2\pi a}{K}\big(k + \frac{m+\tau_m}{M}\big) + \phi_0\Big), \qquad (13)$$

where g_m is the internal gain of the m^{th} sub-ADC, and τ_m is the time skew associated with the m^{th} sub-ADC normalized to the TIADC sampling time. We assume that τ_m has a Gaussian distribution with mean zero and standard deviation σ_τ.

Using (13) and doing a similar analysis as before, we can write the output of the 'input synchronization' block for the m^{th} calibration engine as

$$\tilde{\psi}_m \approx \frac{g_m A}{2} e^{j\left(\phi_0 + \frac{2\pi a (m+\tau_m)}{M\,K}\right)}, \qquad (14)$$

where the phase and magnitude of $\tilde{\psi}_m$ are proportional to τ_m and g_m respectively. All of the $\tilde{\psi}_m$ are calculated using $M\,K$ samples, and these $\tilde{\psi}_m$ values are used to estimate the gain and time skew mismatches using the calculations given below. Note that the estimations of $\tilde{\psi}_m$ are independent from the LUT updating process.

Knowing the input signal amplitude A, we evaluate $\tilde{g}_m$, an estimate for g_m, according to

$$\tilde{g}_m = \frac{2|\tilde{\psi}_m|}{A}. \qquad (15)$$

The gain is compensated after populating the LUT for all sub-ADCs by multiplying the LUT elements by $\bar{g}/\tilde{g}_m$ where $\bar{g}$ is the nominal internal gain for the ADC.

Initially, we consider the first sub-ADC with index 0 as a timing reference, and the relative time skew mismatch can be written as

$$\tau_m - \tau_0 = \frac{M\,K}{2\pi a}\big(\angle\tilde{\psi}_m - \angle\tilde{\psi}_0 - \frac{2\pi a\, m}{M\,K}\big), \qquad (16)$$

where $\angle X$ denotes the phase angle of the complex number X.

The amount of correction applied to the variable delay line connected to the m^{th} sub-ADC can be generalized to take the following form

$$\dot{\tau}_m = \tau_m - r, \qquad (17)$$

where r is an arbitrarily selected timing reference. However, we need to select r such that the requirements on the correction side are relaxed.

The choice of r impacts the range of $\dot{\tau}_m$ that the correction mechanism must support, which in turn will affect the device manufacturing yield. In the following subsections, we examine the impact on yield for various choices of r.

B. Timing reference selection

Each sub-ADC is equipped with a correction mechanism, in this case, a variable delay line as shown in Figure 5. These delay lines can be configured to correct delays within the range $\pm D$. We define η as the target yield, i.e., all $\dot{\tau}_m$ values must be within the range of the delay lines for at least a fraction η of the fabricated ADCs. For a certain ADC chip, if any of these $\dot{\tau}_m$ lie outside the correctable range, those values will be saturated, leaving uncompensated time skews that cause performance degradation, and we consider this chip to be corrupted in this case.

Assuming that r is selected to be equal to 0, we need $|\dot{\tau}_m| < D\ \forall m \in \{0, ..., M-1\}$ for at least η of the chips to satisfy the target yield. From this, we can set the following constraint on D:

$$D \geq \sqrt{2}\sigma_\tau \text{erf}^{-1}\big(\eta^{\frac{1}{M}}\big), \qquad (18)$$

where $\text{erf}^{-1}(.)$ is the inverse of the error function. Note that choosing $r = 0$ is impractical since there is no unique solution for all $\dot{\tau}_m$ using the $M-1$ measurements obtained from (16).

Many estimation algorithms choose the timing of the first sub-ADC as a reference, e.g., [13], [14], [17], [22], [23]. However by (17), choosing $r = \tau_0$ doubles the variance of $\dot{\tau}_m\ \forall m \in \{1, ..., M-1\}$, i.e.,

$$E(\dot{\tau}_m^2) = \sigma_{\dot{\tau}}^2 = 2\sigma_\tau^2, \qquad (19)$$

and hence the constraint on D can be written as follows:

$$D \geq 2\sigma_\tau \text{erf}^{-1}\big(\eta^{\frac{1}{M-1}}\big). \tag{20}$$

Other algorithms employ an extra reference ADC or comparator to act as a timing reference, e.g., [6]–[9], [18]. The variance of the $\dot{\tau}_m$ $\forall m \in \{0, ..., M-1\}$ is the same as in (19); however, this case yields a tighter design requirement on D, since M constraints need to be satisfied by the values $\dot{\tau}_m$. The constraint on D can be expressed as

$$D \geq 2\sigma_\tau \text{erf}^{-1}\big(\eta^{\frac{1}{M}}\big). \tag{21}$$

Note that to find (20) and (21), we assumed that all $\dot{\tau}_m$ are statistically independent.

C. Proposed timing reference

The proposed time skew estimation technique allows obtaining all the M relative time skew values directly from (16) simultaneously, which facilitates the adjusting of the timing reference such that the constraints on the correction side are reduced. Since we need to reduce D, we suggest to choose r to equal the mid-range among all τ_m,

$$r = \frac{\max_m(\tau_m) + \min_m(\tau_m)}{2}, \tag{22}$$

and the delay lines are correspondingly configured to (from (17))

$$\dot{\tau}_m = (\tau_m - \tau_0) - \frac{\max_m(\tau_m - \tau_0) + \min_m(\tau_m - \tau_0)}{2}, \tag{23}$$

where $\tau_m - \tau_0$ $\forall m \in \{0, ..., M-1\}$ are obtained using (16).

Using (23), the maximum delay line configuration value for a given TIADC can be expressed as

$$\dot{\tau}_{\max} = \max_m(\dot{\tau}_m) = \frac{\max_m(\tau_m) - \min_m(\tau_m)}{2}, \tag{24}$$

and the minimum delay line configuration value equals $-\dot{\tau}_{\max}$. This choice of r minimizes the maximum value of $\dot{\tau}_m$, $\dot{\tau}_{\max}$, for a given set of τ_m (c.f. [25]), thus relaxing the requirements on D.

There is no closed form for the probability distribution function (PDF) of $\dot{\tau}_{\max}$ [26]; however, using [27], the PDF of $\max_m(\tau_m)$ (and also of $-\min_m(\tau_m)$) can be approximated to a Gamma distribution, $\max_m(\tau_m) - c_\text{g} \sim \text{Gamma}(k_\text{g}, \theta_\text{g})$, where the position parameter c_g, shape parameter k_g and scale parameter θ_g can be written as

$$c_\text{g} = \big(2.8989 \ln(\log_2(M)) - 4.4291\big)\sigma_\tau, \tag{25}$$

$$k_\text{g} = 4\Big(\frac{\Phi^{-1}(0.5264^{\frac{1}{M}}) - c_\text{g}/\sigma_\tau}{\Phi^{-1}(0.8832^{\frac{1}{M}}) - \Phi^{-1}(0.2142^{\frac{1}{M}})}\Big)^2, \tag{26}$$

$$\theta_\text{g} = \frac{\big(\Phi^{-1}(0.8832^{\frac{1}{M}}) - \Phi^{-1}(0.2142^{\frac{1}{M}})\big)^2}{4\left(\Phi^{-1}(0.5264^{\frac{1}{M}}) - c_\text{g}/\sigma_\tau\right)}\sigma_\tau, \tag{27}$$

and where $\Phi^{-1}(.)$ is the inverse of the Gaussian cumulative distribution function (CDF).

Since M is relatively large, we can assume that $\max_m(\tau_m)$ and $-\min_m(\tau_m)$ are independent, and hence by (24), $\dot{\tau}_{\max}$ has a Gamma distribution with position parameter c_g, shape parameter $2\,k_\text{g}$ and scale parameter $\theta_\text{g}/2$;

$$\dot{\tau}_{\max} - c_\text{g} \sim \text{Gamma}(2\,k_\text{g}, \frac{\theta_\text{g}}{2}). \tag{28}$$

Figure 6 depicts the CDF for $\dot{\tau}_{\max}$ at different M using Monte Carlo simulations and using the approximated distribution in (28) where $\sigma_\tau = 0.01$. It can be observed that this approximation gives an accurate estimation of the CDF for $M > 4$.

Using the approximation in (28), we can set the following constraint on D to satisfy the target yield when the proposed timing reference is used:

$$D \geq \frac{\theta_g}{2}\gamma^{-1}\big(\eta, 2k_g\big) + c_\text{g}, \tag{29}$$

where $\gamma^{-1}(.,.)$ is the inverse lower incomplete gamma function.

As an example, we target a TIADC system with $M = 8$ sub-ADCs suffering from time skew with a standard deviation $\sigma_\tau = 0.01$, and the target yield is $\eta = 98\%$.

Figure 7 shows the measured yield versus the supported delay line half range D at different timing reference configurations; the yield is measured using Monte Carlo simulations for $1,000,000$ time skew sets. The figure depicts also the predicted relationship between η and D using the suggested mathematical models. With $r = 0$, the predicted results from (18) coincide with the simulation. However, (20) and (21) failed to predict accurately the relationship at low yield due to ignoring the dependency among the $\dot{\tau}_m$.

Table I lists the required D to satisfy the target yield for different configurations of r. It can be noticed that selecting r as proposed in (22) helps in reducing the constraint on D by 44% compared to conventional methods (to $D = 0.0236$). Using this value of D, the yield is limited to 60% for the algorithms that use $r = \tau_0$.

It is worth mentioning here that minimizing the value of $\dot{\tau}_{\max}$ helps also to improve the performance in applications that employ digital correction for time skew. Those techniques exploit approximations to simplify the reconstruction of time skew error free samples. The accuracy of those approximations is usually degraded for large correction values.

TABLE I: ESTIMATED REQUIRED D TO SATISFY THE TARGET YIELD.

Selected r	Using the mathematical models	Using Monte Carlo simulation
$r = 0$	0.0302	0.0303
$r = \tau_0$	0.0421	0.0416
With extra ADC	0.0427	0.0421
With proposed r	0.0239	0.0236

V. MISMATCH ESTIMATION INDEPENDENCE

For algorithms that target a specific mismatch calibration, it is commonly assumed that the processed samples are free from errors associated with other mismatch types. For example, under this assumption it is required to use offset, bandwidth and gain mismatch-free samples in techniques which estimate

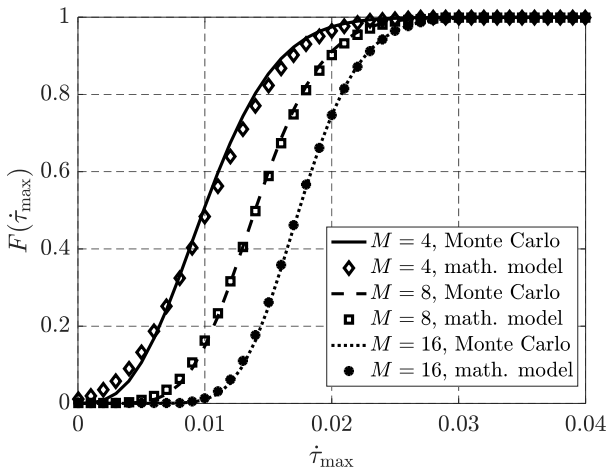

Fig. 6: The CDF for $\dot{\tau}_{\max}$ at different M with $\sigma_\tau = 0.01$.

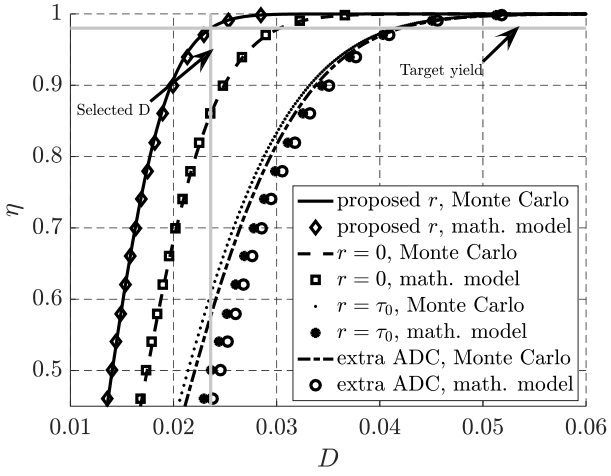

Fig. 7: Estimated yield versus support delay line half range at different timing reference configurations.

the time skew via direct processing of the sample values; this is the case for a wide variety of algorithms, e.g., [6], [7], [13], [14], [17], [18], [22], [23]. However, this is not the case for the algorithm proposed in this paper.

The averaging process used to estimate $\tilde{\psi}_m$ in (8) relaxes the effect of static nonlinearity, and removes the *absolute* offset for each sub-ADC. This makes any processing on $\tilde{\psi}_m$ independent of both offset mismatch and static nonlinearity.

The existence of gain and time skew manifests itself in the magnitude and the phase of $\tilde{\psi}_m$ independently, as seen in (14); this allows extracting the information for both of them *simultaneously* using (15) and (16).

Bandwidth mismatch calibration is not covered in this work[1]; however, in many algorithms available in the literature, its existence may mislead the estimation of the gain and time skew mismatches leading to further performance degradation. This happens because bandwidth mismatch introduces *frequency dependent* gain $g_{\mathrm{B},m}(f_{\mathrm{in}})$ and *nonlinear* phase shift $\theta_{\mathrm{B},m}(f_{\mathrm{in}})$ as follows [28]:

$$g_{\mathrm{B},m}(f_{\mathrm{in}}) = \frac{1}{\sqrt{1+(f_{\mathrm{in}}/B)^2}}, \tag{30}$$

$$\theta_{\mathrm{B},m}(f_{\mathrm{in}}) = -\arctan\left(\frac{f_{\mathrm{in}}}{B+\Delta_m}\right), \tag{31}$$

where f_{in} is the input frequency, B is the 3-dB bandwidth of the RC circuit and Δ_m is the bandwidth mismatch.

Using Taylor series, $\theta_{\mathrm{B},m}(f_{\mathrm{in}})$ can be approximated to

$$\theta_{\mathrm{B},m}(f_{\mathrm{in}}) \approx -\arctan\left(\frac{f_{\mathrm{in}}}{B} - \frac{\Delta_m f_{\mathrm{in}}}{B^2}\right) \tag{32}$$

$$\approx -\arctan\left(\frac{f_{\mathrm{in}}}{B}\right) + \underbrace{\frac{\Delta_m f_{\mathrm{in}}}{B^2}\left[\frac{1}{1+(f_{\mathrm{in}}/B)^2}\right]}_{\text{phase due to BW mismatch}} \tag{33}$$

$$\approx -\arctan\left(\frac{f_{\mathrm{in}}}{B}\right) + \frac{\Delta_m f_{\mathrm{in}}}{B^2}, \tag{34}$$

where the approximations that lead to (32) and (33) are viable because Δ_m is considered small. The approximation that leads to (34) is *weak* since f_{in}/B is not usually small enough to make the term in the square brackets in (33) approximately equal to 1; however, it guarantees that the error in the final approximation is less than the phase mismatch introduced due to bandwidth mismatch for $f_{\mathrm{in}} < B$.

It can be noticed from (33) and (34) that the effect of the phase shift introduced by the bandwidth mismatch can be approximated to a *linear* phase shift that can be treated as a time skew mismatch. Compensating this linear phase using the delay lines reduces the amount of the phase error introduced by bandwidth mismatch for $f_{\mathrm{in}} < B$, which offers a *partial* correction for the bandwidth mismatch.

In principle, the estimation for the linear phase component introduced via bandwidth mismatch can be carried out using normal time estimation process; however, many of the available techniques are sensitive to gain mismatch, e.g., [7], [13], [14], [17], [18], [22], [23] – these techniques are affected by the frequency dependent gain in (30), which leads to an incorrect time skew estimation that worsens the performance. The use of a background gain calibration can mask this problem for a narrowband input where $g_{\mathrm{B},m}(f_{\mathrm{in}})$ can be compensated; however, it does not help for wideband signals.

Unlike those techniques, the proposed algorithm estimates the time skew via processing the phase of the input signal without an impact from gain mismatch. This allows successful estimation of the time skew values that include the linear components introduced by the bandwidth mismatch.

VI. Built-in self-test (BIST)

The estimation of $\tilde{e}[k]$ in (11) enables exploiting the proposed mechanism as a built-in self-test after the calibration process without adding much complexity to the proposed circuit, where we can 1) measure the signal-to-noise-and-distortion ratio (SNDR) for the ADC output, and 2) detect the comparator metastability. Those two applications are illustrated in the following subsections.

[1]It is possible to exploit the proposed calibration mechanism to measure the bandwidth mismatch via applying two sinusoids with different frequencies, i.e., with different values of a. Comparing $\tilde{\psi}_m$ obtained for each case helps in identifying the bandwidth mismatch.

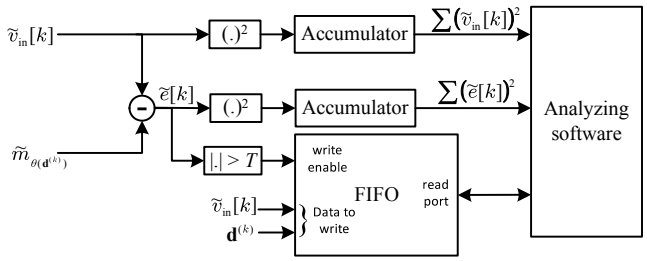

Fig. 8: Suggested in-circuit mechanism to estimate the SNDR and the comparator metastability.

A. Measuring SNDR

To measure the SNDR, we assume that the input signal is $\tilde{v}_{\text{in}}[k]$, and the noise is the estimated $\tilde{e}[k]$. With these assumptions, we can measure the power of both the noise and the signal as shown in Figure 8, and the SNDR can be measured according to

$$SNDR = \frac{\sum_{k=0}^{L-1}(\tilde{v}_{\text{in}}[k])^2}{\sum_{k=0}^{L-1}(\tilde{e}[k])^2}, \tag{35}$$

where L is the number of considered samples.

B. Comparator metastability detection

Minimizing the comparator metastability rate is another design criterion that needs to be fulfilled. This metastability occurs when the comparator decisions fail to produce a binary output [29], leading to a large error on the samples which experience such behaviour. This consequence is exploited in [29]–[32] to detect those events. In [30] and [31], a very low frequency input sinusoid is used, where the expected difference between two successive digital outputs is less than 1; the comparator metastability is then detected when the difference is greater than 1. A similar idea is used in [32] and [29], where a higher input frequency can be used; however, the output is hard decimated such that the expected difference between two successive digital samples after decimation is kept less than 1.

In the proposed calibration algorithm, we estimate a value for the error associated with each sample, $\tilde{e}[k]$, and the metastability condition is detected when $|\tilde{e}[k]| > T$, where T is a selected threshold. Figure 8 depicts a simple mechanism to detect and handle this event. On detecting comparator metastability, both $\tilde{v}_{\text{in}}[k]$ and $\mathbf{d}^{(k)}$ are pushed into a first-in-first-out (FIFO) block, the output of the FIFO is read by a slow-running analyzing software that can calculate the comparator metastability rate, and identify systematic errors.

VII. Results

In this section, we verify the proposed algorithm's performance using Matlab simulations targeting a 10-bit differential radix-2 SAR ADC (note however that the proposed algorithm is not limited to this architecture). The ADC unit capacitance suffers from mismatch having a Gaussian distribution with a standard deviation equal to 10% of its nominal value, which is large enough to produce missing codes, large differential nonlinearity (DNL) and a non-monotonic transfer function. To model a realistic ADC, comparator noise is added to limit the ENOB to around 9 bits. Unless otherwise specified, we used the following configurations, $\alpha = 2^{-3}$, $K = 2^{12}$ and $F = 61440$; the total number of processed samples per single sub-ADC is $K+F = 2^{16}$. For a single channel ADC $a = 409$ is used, and $a = 1433$ is used for the tests that target a TIADC system. The final ADC output is truncated to $N = 10$ bits, while the LUT values are stored as $N + b = 15$ bits.

To demonstrate that the calibration values do not correlate with a specific input frequency, the frequency of the test signal used to evaluate the performance after calibration is randomly selected with a uniform distribution up to the Nyquist rate.

The following subsections present the results targeting a single channel ADC and time-interleaved ADC. Subsection VII-C reports the results from synthesizing the hardware implementation of the proposed calibration engine.

A. Using a single channel ADC

A Monte Carlo simulation is used to verify the proposed algorithm's performance in compensating the capacitor mismatch problem, where 5000 different capacitor sets are used. Figure 9 shows the ENOB distributions when 1) a linear combination with weights proportional to the capacitance values (ideal weights) is used to obtain the output, 2) no calibration is used, and 3) the proposed calibration algorithm is used. The measured average ENOB are 8.97, 7.90 and 8.99 bits for the aforementioned configurations respectively. Note that we used different ranges for the ENOB axis in the results depicted in Figure 9.

Using one of the described tests, the moving average over 128 measurements for the error signal $|\tilde{e}|$ is depicted in Figure 10, where the non-updated LUT entries are replaced by their nominal values to evaluate $\tilde{e}$ only for demonstration purposes. In this simulation, the system converges within $20,000$ samples.

The SNDR after calibration is evaluated using (35) and using the `sinad` Matlab function; the distribution of the difference between the two results is depicted in Figure 11 showing an error that is limited to below 0.3dB.

In SAR ADCs, a settling time τ_{s} is allocated after each capacitor switch. This permits the voltage presented to the comparator to settle with an RC time constant τ_{rc}. If τ_{s} is insufficient, errors may occur [3]. Increasing the sampling frequency (i.e., reducing $\tau_{\text{s}}/\tau_{\text{rc}}$) exacerbates this issue. Since the foreground calibration is done using the same sampling frequency as in normal operation, the LUT values $\tilde{m}_{\theta(\mathbf{d})}$ are tailored to the appropriate value of $\tau_{\text{s}}/\tau_{\text{rc}}$, which aids in providing a better performance compared to the linear combination based approach. However, the proposed algorithm is not able to avoid the problem of missing codes which occurs for sufficiently small values of $\tau_{\text{s}}/\tau_{\text{rc}}$.

Figure 12 compares the measured ENOB obtained using the proposed algorithm with that obtained using the linear combination with ideal weights. It can be observed that the degradation of the ENOB due to reducing $\tau_{\text{s}}/\tau_{\text{rc}}$ is smaller

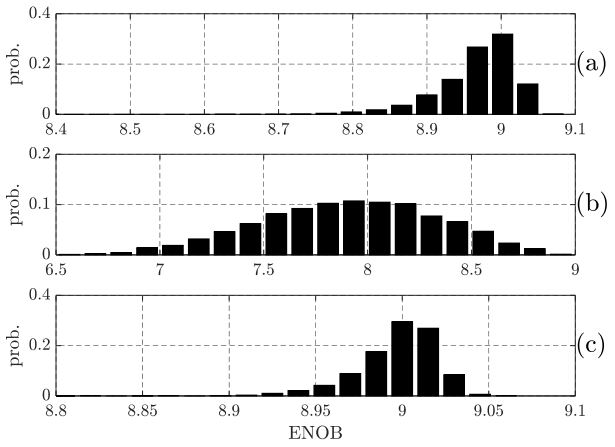

Fig. 9: ENOB distribution for SAR ADC with (a) linear combination using ideal weights, (b) without calibration, and (c) proposed calibration algorithm, modified from [24].

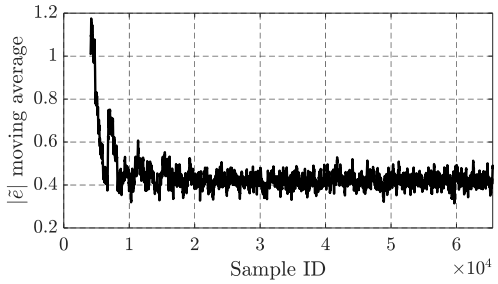

Fig. 10: Evolution of the error signal $|\tilde{e}|$ during the calibration process.

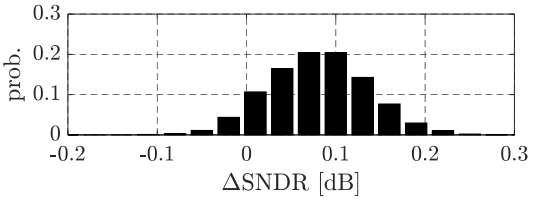

Fig. 11: Distribution of the error in the estimated SNDR using (35).

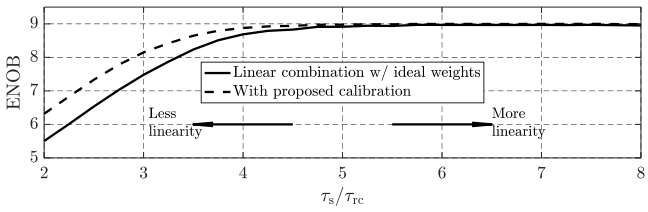

Fig. 12: Measured ENOB with varying τ_s/τ_{rc} [24].

when the proposed algorithm is used. This allows an increase in the sampling frequency with a minor loss in performance.

Figure 13 shows the measured ENOB on changing the S/H nonlinearity coefficients c_2 and c_3 in (1). It can be observed that the performance after calibration has a far greater immunity to the nonlinearity of the S/H.

B. Using time-interleaved ADC

Here, the same ADC configurations are used to form a TIADC system with $M = 8$ sub-ADCs and aggregated

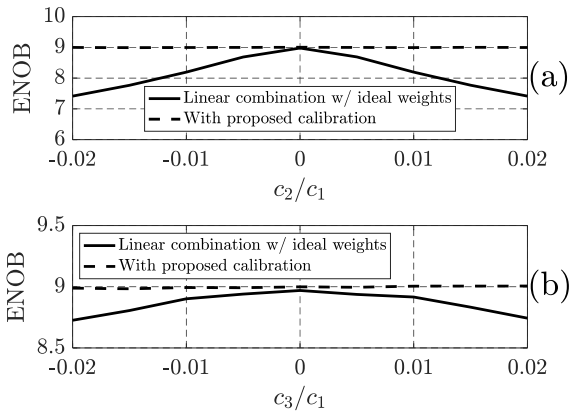

Fig. 13: Measured ENOB with varying S/H nonlinearity coefficients, modified from [24].

sampling rate F_s. The system suffers from time skew mismatch having a Gaussian distribution with standard deviation $\sigma_\tau = 0.01$.

In the first test, we target time skew calibration only. Here, we use $K = 2^{10}$, and $M\,K = 2^{13}$ samples are processed to evaluate all $\tilde{\psi}_m$ needed for the estimation. Figure 14 depicts the average measured SNDR and SFDR before and after calibration at different testing input frequency; each point in the figure is a result of averaging the evaluated performance over 25 tests. Without calibration, continuous performance degradation occurs on increasing the input frequency. On the other hand, the evaluated SNDR after calibration is maintained around 56dB which corresponds to 9 bits ENOB. The measured SNDR around the Nyquist frequency is 55.7dB. The observed SFDR degradation is due to the remaining uncompensated time skew, which can be minimized by increasing K and enhancing the delay line resolution.

The root mean square (RMS) value of the uncompensated time skew (or time skew *residue*) can be calculated as

$$\varepsilon = \sqrt{\frac{1}{M}\sum_{m=0}^{M-1}\left(\tau_m - \dot{\tau}_m - \left[\frac{\sum_{m=0}^{M-1}\tau_m - \dot{\tau}_m}{M}\right]\right)^2}, \quad (36)$$

where the term in the square brackets is used to compensate the global delay. Figure 15 shows the evaluated distribution for ε before and after calibration. On average, ε is reduced from 9.3×10^{-3} to 1.7×10^{-4}, which is an indication for successful time skew estimation.

Also, Figure 16 compares the distribution of the maximum configured $\dot{\tau}_m$ for each test considered in the results depicted in Figure 14 and the approximated distribution suggested in (28). In those tests, $\max_m(|\dot{\tau}_m|) < D = 0.0236$ in 791 out of 800 tests which conforms with the target yield $\eta = 98\%$.

In addition to time skew mismatch, the previous test was repeated where the TIADC system suffers from capacitor mismatch. Further offset and gain errors for each sub-ADC are assigned, where the standard deviation for the applied offset and gain error are 0.5% of the full input signal swing and 1% of the nominal gain respectively. The full proposed calibration algorithm is carried out for both TIADC mismatches and non-

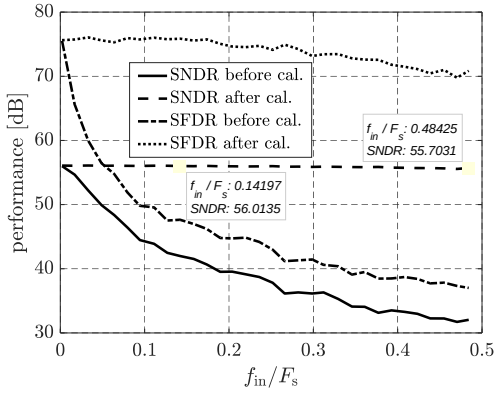

Fig. 14: Average measured SNDR and SFDR before and after calibration at different input frequency.

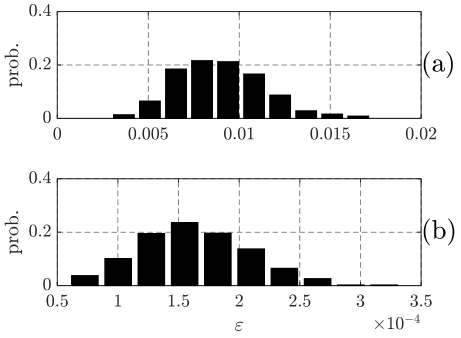

Fig. 15: Distribution of the RMS of the time skew residue (a) without calibration where all $\dot{\tau}_m = 0$, and (b) $\dot{\tau}_m$ is evaluated using the proposed calibration algorithm.

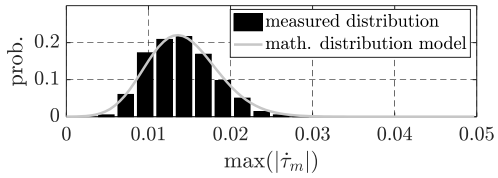

Fig. 16: Distribution of the maximum configured $\dot{\tau}_m$ in each test.

linearities of the sub-ADCs, where $K = 2^{12}$. The total number of samples used for the full calibration is $M(K + F) = 2^{19}$; only the final $M\,K$ samples are used to estimate the gain and time skew. Figure 17 shows the evaluated distribution for the ENOB before and after calibration. The ENOB is improved from 5.47 to 8.98 bits on average.

In the following tests, we study the effect of the existence of bandwidth mismatch on the proposed algorithm, the sample and hold circuit is modeled as in [28] with nominal bandwidth $B = F_{\rm s}$, the bandwidth suffers from mismatch having a standard deviation equal to 0.5% of its nominal value. The results of the following four test setups are presented in Figure 18, where each point in the figure is obtained by measuring the average ENOB over 25 tests:

1) *Without any mismatch:* The noticed performance degradation on increasing $f_{\rm in}/F_{\rm s}$ is due to the limited bandwidth of the RC circuit that shrinks the input amplitude.
2) *With bandwidth mismatch only, before calibration:* This mismatch causes more performance degradation as $f_{\rm in}$ moves toward B; the system performance is worsened by 0.6 bits around $f_{\rm in} = F_{\rm s}/2$.
3) *With bandwidth mismatch only, after calibration:* It can be noticed that, although the bandwidth mismatch calibration is not supported directly, time skew estimation and correction facilitate its partial calibration enabling performance improvement; there is 0.2 bit improvement at $f_{\rm in} = F_{\rm s}/2$ compared to the previous test setup. However, a minor performance degradation is noticeable at low frequencies due to skewing the estimated values by the introduced frequency dependent gain and the nonlinear phase mismatches.
4) *With bandwidth and other mismatches, after calibration:* In this test the TIADC system suffers from offset, gain, time skew, bandwidth and capacitor mismatches. The performance after calibration is maintained similar to the results obtained in the previous test setup.

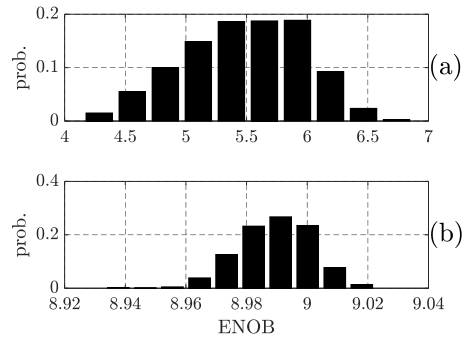

Fig. 17: Distribution of the measured ENOB for a TIADC system suffers from offset, gain, time skew and capacitor mismatch (a) before calibration, (b) after calibration.

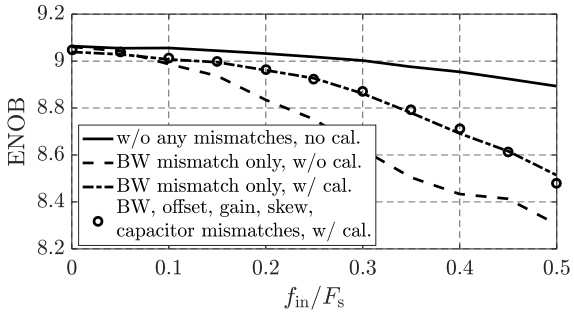

Fig. 18: Average ENOB at different $f_{\rm in}$ with and without calibration on the existence of bandwidth mismatch, $B = F_{\rm s}$.

C. Hardware implementation and comparison of estimation techniques

A Verilog model was implemented for the calibration engine using the specifications reported at the beginning of this section. To save area, we decided to implement the LUT as a single port memory, where the LUT is updated only using the even-indexed samples. The hardware design was verified against a fixed-point Matlab model to be bit-accurate using the Cadence Incisive simulator. The design was synthesized using the Cadence Genus synthesis tool targeting a 250MHz

TABLE II: AREA UTILIZATION AND POWER BREAKDOWN FOR THE CALIBRATION ENGINE.

	Area	Area%	Power	Power%
Memory	$5,250\mu m^2$	**39**%	635μW	23%
Cordic	$3,325\mu m^2$	25%	941μW	**33**%
Multipliers	$3,141\mu m^2$	23%	705μW	25%
Miscellaneous	$1,672\mu m^2$	13%	548μW	19%
Total	$13,388\mu m^2$	100%	$2,829\mu$W	100%

clock using a TSMC 28nm HPM process. The design occupies an area of $13,388\mu m^2$ which is dominated by the single port memory[2] as shown in the area utilization breakdown in Table II. Note that the required memory size increases exponentially with the ADC resolution, which makes the proposed calibration technique unsuitable for high resolution ADCs.

A gate level simulation was successfully run, all internal signal waveforms were dumped into a Value-Change Dump (VCD) file, which is used to collect the switching activity for each net in the design, allowing an accurate power estimation. During the foreground calibration, the design consumes 2.8mW operating on a 250MHz clock. Table II reports the power breakdown. In normal operation mode, only the LUT is active which consumes 0.59mW.

Finally, Table III provides a summary comparison of the proposed algorithm with other state-of-the-art algorithms [2], [7]–[9], [11], [15].

VIII. CONCLUSION

In this paper, a generic foreground calibration technique has been presented for high speed ADCs with low to medium resolution, which calibrates various nonlinearity sources for a single channel ADC and TIADC. With the use of a sinusoidal input, the calibrated output corresponding to each raw output is evaluated and stored in a LUT. This technique obviates the calibration post-processing, substituting it with a memory read access. Various design simplifications were introduced to facilitate a real-time hardware implementation. In addition to calibrating the nonlinearity of each sub-ADC in a TIADC system, the algorithm can be used to estimate the mismatches in this system independently, and it also offers a partial calibration for bandwidth mismatch. We proposed to choose the timing reference for the estimated time skews such that the mid-range of the estimated time skews is zero; this choice reduces the requirements on the correction side by 44% compared to conventional methods. In addition to calibration, the proposed mechanism can be used as a built-in self-test to detect the comparator metastability and to evaluate the SNDR performance without transferring a large amount of data outside the ADC. A SAR ADC model was used to verify the algorithm's performance. Compared to the linear combining approach using ideal weights, the algorithm showed superior capacitor mismatch calibration, increased tolerance to settling time reduction and significant improvements in the presence of high order nonlinear terms. The area utilization and power consumption for the calibration engine were reported, which demonstrates the feasibility of the proposed algorithm implementation.

[2]The area and power estimations for the memory were collected manually using the target technology datasheet.

REFERENCES

[1] C. Grace, P. Hurst, and S. Lewis, "A 12-bit 80-MSample/s pipelined ADC with bootstrapped digital calibration," *IEEE J. Solid-State Circuits*, vol. 40, no. 5, pp. 1038–1046, May 2005.

[2] W. Liu, P. Huang, and Y. Chiu, "A 12-bit, 45-MS/s, 3-mW redundant successive-approximation-register analog-to-digital converter with digital calibration," *IEEE J. Solid-State Circuits*, vol. 46, no. 11, pp. 2661–2672, Nov. 2011.

[3] A. Salib, M. F. Flanagan, and B. Cardiff, "Blind SAR ADC capacitor mismatch calibration," in *Proc. IEEE Int. Midwest Symp. on Circuits and Syst. (MWSCAS)*, Aug. 2017, pp. 587–590.

[4] A. H. Chang, H.-S. Lee, and D. Boning, "A 12b 50MS/s 2.1mW SAR ADC with redundancy and digital background calibration," in *Proc. Eur. Solid State Circuits Conf. (ESSCIRC)*, Sep. 2013, pp. 109–112.

[5] S. Liu, N. Lyu, J. Cui, and Y. Zou, "Improved blind timing skew estimation based on spectrum sparsity and ApFFT in time-interleaved ADCs," *IEEE Trans. Instrum. Meas.*, pp. 1–14, 2018.

[6] A. Mrassy and M. Dessouky, "Channel mismatch background calibration for pipelined time interleaved ADCs," in *Proc. IEEE Int. Conf. on Electron., Circuits and Syst. (ICECS)*, Dec. 2012, pp. 609–612.

[7] Y.-Z. Lin, C.-H. Tsai, S.-C. Tsou, and C.-H. Lu, "A 8.2-mW 10-b 1.6-GS/s 4x and TI SAR ADC with fast reference charge neutralization and background timing-skew calibration in 16-nm CMOS," in *IEEE Proc. Symp. on VLSI Circuits (VLSI)*, Jun. 2016, pp. 1–2.

[8] M. El-Chammas and B. Murmann, "A 12-GS/s 81-mW 5-bit time-interleaved flash ADC with background timing skew calibration," *IEEE J. Solid-State Circuits*, vol. 46, no. 4, pp. 838–847, Apr. 2011.

[9] J. Song, K. Ragab, X. Tang, and N. Sun, "A 10-b 800-MS/s time-interleaved SAR ADC with fast variance-based timing-skew calibration," *IEEE J. Solid-State Circuits*, vol. 52, no. 10, pp. 2563–2575, Oct. 2017.

[10] S. Lee, A. P. Chandrakasan, and H.-S. Lee, "A 1 GS/s 10b 18.9 mW time-interleaved SAR ADC with background timing skew calibration," *IEEE J. Solid-State Circuits*, vol. 49, no. 12, pp. 2846–2856, Dec. 2014.

[11] A. J. Gins, E. J. Peralas, and A. Rueda, "Black-box calibration for ADCs with hard nonlinear errors using a novel INL-based additive code: A pipeline ADC case study," *IEEE Trans. on Circuits and Syst. I, Regular Papers*, vol. 64, no. 7, pp. 1718–1729, Jul. 2017.

[12] V. Kerzrho, V. Fresnaud, D. Dallet, S. Bernard, and L. Bossuet, "Fast digital post-processing technique for integral nonlinearity correction of analog-to-digital converters: validation on a 12-bit folding-and-interpolating analog-to-digital converter," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 3, pp. 768–775, Mar. 2011.

[13] C.-Y. Lin, Y.-H. Wei, and T.-C. Lee, "A 10b 2.6GS/s time-interleaved SAR ADC with background timing-skew calibration," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Jun. 2016, pp. 468–469.

[14] H. Wei, P. Zhang, B. D. Sahoo, and B. Razavi, "An 8 bit 4 GS/s 120 mW CMOS ADC," *IEEE J. Solid-State Circuits*, vol. 49, no. 8, pp. 1751–1761, Aug. 2014.

[15] N. L. Dortz, J.-P. Blanc, T. Simon, S. Verhaeren, E. Rouat, P. Urard, S. L. Tual, D. Goguet, C. Lelandais-Perrault, and P. Benabes, "A 1.62GS/s time-interleaved SAR ADC with digital background mismatch calibration achieving interleaving spurs below 70dBFS," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2014, pp. 386–388.

[16] A. Salib, B. Cardiff, and M. F. Flanagan, "A low-complexity correlation-based time skew estimation technique for time-interleaved SAR ADCs," in *Proc. IEEE Int. Symp. on Circuits and Syst. (ISCAS)*, May 2017, pp. 1–4.

[17] S. Jamal, D. Fu, N.-J. Chang, P. Hurst, and S. Lewis, "A 10-b 120-msample/s time-interleaved analog-to-digital converter with digital background calibration," *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 1618–1627, Dec. 2002.

[18] E. Iroaga, B. Murmann, and L. Nathawad, "A background correction technique for timing errors in time-interleaved analog-to-digital converters," in *Proc. IEEE Int. Symp. on Circuits and Syst. (ISCAS)*, May 2005, pp. 5557–5560.

TABLE III: COMPARISON WITH THE STATE-OF-THE-ART TECHNIQUES FOR TIME SKEW AND NONLINEARITY CALIBRATION.

	TCAS-I **This work**	JSSC 2011 [8]	ISSCC 2014 [15]	VLSI 2016 [7]	JSSC 2017 [9]	JSSC 2011 [2]	TCAS-I 2017 [11]
SNDR [dB] @Nyquist	55.7	25.1	48	50.3	48	70.1	56.1
Foreground	Yes	No	No	No	No	No	Yes
Input requirements	Sinusoidal	Large difference between successive samples, avoid limitations listed in [16]				Covering most of the input swing	Sinusoidal
Hardware overhead	Signal gen., Cordic, $2M$ mul., ($2M$ mul., M LUT)[a]	Reference comparator	Bank of M correlators	Reference ADC	Window detector	Adaptive filter, analog adder, double sampling	Signal gen., adaptive filter, LUT
Time skew cal.	Yes	Yes	Yes	Yes	Yes	No	No
M	8	8	12	4	4	–	–
N# sample for time skew est.	8K	$2,000$M	N/A	N/A	6.8M[b]	–	–
Nonlinear cal.	Yes	No	No	No	No	Yes	Yes
N# sample for nonlinear cal.	20K	–	–	–	–	40K	131K

[a] Used for ADC nonlinear calibration only.
[b] Using a sinusoidal input.

[19] W. Kester, *The Data Conversion Handbook*. Elsevier, 2005.
[20] K. Poulton, R. Neff, A. Muto, W. Liu, A. Burstein, and M. Heshami, "A 4 Gsample/s 8b ADC in 0.35 μ m CMOS," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, vol. 1, 2002, pp. 166–457.
[21] Y. M. Greshishchev, J. Aguirre, M. Besson, R. Gibbins, C. Falt, P. Flemke, N. Ben-Hamida, D. Pollex, P. Schvan, and S.-C. Wang, "A 40GS/s 6b ADC in 65nm CMOS," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2010, pp. 390–391.
[22] S. Chen, L. Wang, H. Zhang, R. Murugesu, D. Dunwell, and A. C. Carusone, "All-digital calibration of timing mismatch error in time-interleaved analog-to-digital converters," *IEEE Trans. on Very Large Scale Integration Syst.*, vol. 25, no. 9, pp. 2552–2560, Sep. 2017.
[23] C.-Y. Lin, Y.-H. Wei, and T.-C. Lee, "A 10-bit 2.6-GS/s time-interleaved SAR ADC with a digital-mixing timing-skew calibration technique," *IEEE J. Solid-State Circuits*, vol. 53, no. 5, pp. 1508–1517, May 2018.
[24] A. Salib, M. F. Flanagan, and B. Cardiff, "A generic foreground calibration algorithm for ADCs with nonlinear impairments," in *Proc. IEEE Int. Symp. on Circuits and Syst. (ISCAS)*, May 2018, pp. 1–4.
[25] D. R. Shier and C. J. Witzgall, "Norm approximation problems and norm statistics," *J. of Research of the National Bureau of Standards*, vol. 83, no. 1, Jun. 1978.
[26] E. J. Gumbel, "The distribution of the range," *The Annals of Mathematical Statistics*, vol. 18, no. 3, pp. 384 – 412, 1947.
[27] C.-C. Chen and C. W. Tyler, "Accurate approximation to the extreme order statistics of gaussian samples," *Communications in Statistics - Simulation and Computation*, vol. 28, no. 1, 1999.
[28] N. Kurosawa, H. Kobayashi, K. Maruyama, H. Sugawara, and K. Kobayashi, "Explicit analysis of channel mismatch effects in time-interleaved ADC systems," *IEEE Trans. on Circuits and Syst. I, Fundamental Theory and Applications*, vol. 48, no. 3, pp. 261–271, Mar. 2001.
[29] S. Guhados, P. J. Hurst, and S. H. Lewis, "A pipelined ADC with metastability error rate $< 2^{-15}$," *IEEE J. Solid-State Circuits*, vol. 47, no. 9, pp. 2119–2128, Sep. 2012.
[30] S. Hashemi and B. Razavi, "Analysis of metastability in pipelined ADCs," *IEEE J. Solid-State Circuits*, vol. 49, no. 5, pp. 1198–1209, May 2014.
[31] C.-H. Chan, Y. Zhu, S.-W. Sin, B. Murmann, S.-P. U, and R. P. Martins, "Metastablility in SAR ADCs," *IEEE Trans. on Circuits and Syst. II, Express Briefs*, vol. 64, no. 2, pp. 111–115, Feb. 2016.
[32] C. W. Mangelsdorf, "Improving bandwidth and error rate in flash converters," in *Proc. Symp. on VLSI Circuits (VLSI)*, May 1989, pp. 53–54.

Armia Salib (S'18) received his B.Sc. degree with honors from Alexandria University, Egypt, and an M.Sc. degree from Ain Shams University. He has gained a valuable industry experience as a digital design engineer at SySDSoft, Intel and Axxcelera. He is the recipient of SySDSoft's Employee of the Year award in 2010. Since 2015, he has been working toward the Ph.D. degree at University College Dublin, Ireland, with a focus on ADC digital calibration.

Mark F. Flanagan (M'03-SM'10) received the B.E. and Ph.D. degrees in electronic engineering from University College Dublin, Ireland, in 1998 and 2005, respectively. From 1998 to 1999, he was a Project Engineer with Parthus Technologies Ltd. From 2006 to 2008, he held post-doctoral research fellowships with the University of Zurich, Switzerland, the University of Bologna, Italy, and The University of Edinburgh, U.K. In 2008, he was appointed an SFI Stokes Lecturer in electronic engineering with University College Dublin, where he is currently an Associate Professor. In 2014, he was a Visiting Senior Scientist with the German Aerospace Center, Institute of Communications and Navigation, under a DLR-DAAD Fellowship. His research interests include information theory, wireless communications, and signal processing. He has served on the technical program committees of several IEEE international conferences. He is currently a Senior Editor of the IEEE COMMUNICATIONS LETTERS.

Barry Cardiff (M'06) received the B.Eng., M.Eng.Sc., and Ph.D. degrees in electronic engineering from University College Dublin, Ireland, in 1992, 1995 and 2011, respectively. He was a Senior Design Engineer / Systems Architect for Nokia from 1993 to 2001, moving to Silicon & Software Systems (S3 group) thereafter as a Systems Architect in their R&D division focused on wireless communications and digitally assisted circuit design. Since 2013 he is an Assistant Professor at University College Dublin, Ireland. His research interests are in digitally assisted circuit design and signal processing for wireless and optical communication systems. He holds several US patents related to wireless communication.