Static Integral Nonlinearity Modeling and Calibration of Measured and Synthetic Pipeline Analog-Digital Converters

Samer Medawar, Boris Murmann, Peter Händel, Niclas Björsell, Magnus Jansson

Abstract

The integral nonlinearity (INL) modeling of pipeline analog-digital converters (ADCs) is investigated in this paper. The INL is divided into two distinct entities: a low code frequency component (LCF) and a high code frequency (HCF) component. Two static models are developed to represent the INL data. In both models, the LCF component is represented by a low-order polynomial. The HCF modeling is performed using two different basis functions: sinc and Gaussian. The structure of both HCF models is motivated by the pipeline architecture of the ADC under investigation. The model coefficients are estimated by applying the least-squares method to the measured INL data from two samples of a commercial pipeline ADC. The estimated HCF models are compared to each other and to previous models presented in the existing literature. In addition, the modeling methods are applied to synthetic HCF data generated by a pipeline ADC simulation model constructed in Matlab. The INL models are then used to calibrate the synthetic ADCs, and the improvements in spurious free dynamic range (SFDR) are compared to those obtained when the ADCs are compensated by the INL data. Furthermore, the capability of the HCF modeling to calibrate a given ADC is tested by using the HCF model to compensate a synthetically generated ADC output in which only the measured HCF sequence and noise are added to the quantization process. The results show that the developed HCF models can achieve virtually complete calibration of the considered ADC.

Index Terms

Pipeline ADC, integral nonlinearity, modeling, static calibration, synthetic data, Monte-Carlo simulations.

I. INTRODUCTION

The analog-digital converter (ADC) is a mixed-signal device that bridges the gap between the analog waveform and the digital signal processing (DSP) block in a digital system. Specifically, the role of an ADC is to quantize an analog signal into a digital sequence. Errors persist in the analog waveform

sampling and quantization process, leading to distorted ADC output sequences. There are various origins of errors inside an ADC, each of which contributes in its own way to ADC output distortions. The integral nonlinearity (INL) is one of the most indicative ADC characteristic, into which most of the ADC circuitry imperfections are translated. The INL is defined as the deviation of a practical quantizer from an ideal quantizer [1]; the INL describes the deviations of the actual ADC output relative to the output of an ideal quantizer.

The INL can be characterized (using single tone inputs) by the histogram test [1]. Therefore, for an N -bit ADC, an INL sequence of length $2^N - 1$ is obtained for every frequency f_m under test. If the frequency of the ADC input stimulus is known, the INL sequences can be used in a look-up table (LUT) fashion to post-correct the ADC in use. The compensation is performed as [2]

$$s_k = x[k] - Q i[m, k]. \quad (1)$$

where k is the ADC digital output code, s_k is the compensated ADC output, $x[k]$ is the original ADC digital output, Q is the ideal code bin width of the ADC and $i[m, k]$ is the INL data at frequency f_m and code k . INL modeling has been an active research topic because the given model can be used for ADC post-correction, as in (1). The INL model typically consists of few parameters compared to the INL data. Thus, a model-based compensation block will replace a LUT with a considerable number of entries that contain either the ADC error or the INL data. Additionally, INL modeling may be of interest in its own right.

A. State-of-the-art INL modeling

A dynamic INL model consisting of a single two-dimensional (input signal slope and output code k) polynomial was presented in [3]. In [4]–[8], a static (dependent solely on the code k) INL model was developed: a so-called low code frequency (LCF) component modeled by a low-order polynomial and a so-called high code frequency (HCF) component computed as the cumulative sum with respect to k of the differential nonlinearities (DNLs).

A dynamic INL model was presented and parameterized, and its coefficients were estimated in [9]. The model consisted of a static HCF component in terms of the ADC output code k and a dynamic LCF component that has an additional frequency f_m dependency. The LCF component was modeled by using polynomials with frequency-dependent coefficients. The HCF component was represented by a

set of piecewise linear segments centered around zero. This modeling method [9] was evaluated using measured data from two pipeline (Analog Devices AD9430) ADCs of the same type. Although this HCF model accurately represented the HCF data for one of the sample ADCs, it only demonstrated moderate performance in modeling the HCF data of the second sample ADC. Additionally, the INL model structure proposed in [9] cannot be identified in a straight-forward manner before the least-squares (LS) fit between the data and the model. A minimization tool was developed to choose the LCF polynomial order L (the polynomial is then subtracted from the INL data) that can render the HCF data centered around zero. The number P of HCF segments was computed using an algorithm that included a low pass filtering of the HCF data. Once the P number of HCF segments is obtained, a minimization criterion is used to optimize the widths and locations of the HCF segments.

A post-correction method was developed in [2] based on the INL model developed in [9]. It was found that superior static compensation results can be obtained for one of the investigated ADCs if a more representative static model (i.e., HCF modeling) was developed. This observation was subsequently confirmed in [10], where tangible improvements (over [2]) were obtained with a static correction when the actual HCF data were used instead of the HCF model in [9].

B. Goal of the paper

This work presents a methodology for modeling the (static) HCF component and the INL data in general. Previous works in the literature do not make use of the pipeline structure to derive the scheme of the HCF; these works use various ad-hoc approaches combined with complex pre-processing to derive the model structure before estimating the parameters of the given INL model. We propose suitable HCF model structures, beginning with the pipeline circuitry. In general, the INL modeling techniques developed in the existing literature are tested on and tailored to a specific ADC. We are interested in showing the potential and versatility of INL modeling in this paper by testing its ability to represent any INL data scheme.

C. Contribution of the paper

Two INL models are proposed in this paper. Both models present simpler means of obtaining their structure (than the model in [9]) and capturing more of the static INL information, i.e., the HCF component. Moreover, these models lead to superior representations of the INL data.

Fig. 1. A static description of a pipeline ADC in which the input s produces the digital output x through the output code k .

Because a more effective INL model will result in an enhanced post-correction (shown at the end of the paper), we analyze the modeling methodology in terms of its post-correction performance. A simulated model of a pipeline ADC, with the structure of the AD9430, is constructed in Matlab. Various sources of static errors are induced in the ADC model. Monte Carlo simulations are run to generate different static INL (HCF) sequences. The static INL modeling performance is evaluated by the ability of the HCF modeling developed in this paper to calibrate the ADC.

D. Organization of the paper

This paper is organized as follows: Sec. II presents the pipeline ADC diagram and imperfections, and Sec. III introduces the INL data estimation and recalls previous INL modeling approaches in the literature. The INL modeling techniques of this paper are parameterized, and the LS solution is derived in Sec. IV. The modeling results are shown in Sec. V, where the calibration results-based on synthetic data-are also presented. Sec. VI concludes the work.

II. THE PIPELINE ADC STRUCTURE AND IMPERFECTIONS

A. The Pipeline ADC Diagram

The pipeline ADC is a typical converter used in a telecommunication system, such as wireless base stations. A conventional pipeline ADC structure is shown in Fig. 1. The analog input s is fed to the ADC, where the signal goes through the subsequent S pipeline stages and the final flash ADC. Each pipeline ADC stage primarily consists of a sample-and-hold entity, a flash ADC, a digital-analog converter (DAC)

and a stage amplifier. In the first stage, the analog voltage is quantized by an N_1 -bit flash ADC, and the obtained code is directed to the output code register. The same obtained code is used to address an N_1 -bit DAC. The output of the DAC is subtracted from the sampled analog input voltage, and the residual is amplified by gain g_1 (as shown for the second stage, g_2 in Fig. 1) prior to being fed to the next stage. The process continues through the remaining $S - 1$ stages. The gained residual of stage S is fed into an N_3 -bit flash ADC, which resolves the remaining output code of the ADC. The N -bit output code k is constructed with the individual $(N_1 + (S - 1) \times N_2 + N_3)$ sub-codes provided by the S stages and the flash ADC. Each stage resolves the subsequent signal sample after each clock cycle, which explains the high throughput of the pipeline structure.

B. Circuitry imperfections

The origins of the HCF component of the INL are related to the stage gain errors, the flash comparator offsets, and the DAC level offsets [11]. These components can be modeled as Gaussian random variables with the mean taken as the desired value for the ideal design and a given variance. The standard deviation of the offsets of the flash comparators can be approximately 6%, while the gain error of each stage and the deviations of the DAC levels are significantly less than this value. In practice, all ADCs suffer from static HCF error. However, this HCF can be quite small in magnitude if the first stage gain is close to its desired value. The first stage encompasses the major part of the HCF error and defines the general structure of the HCF data. Due to the deviation in the first stage gain g_1 , the INL experiences either a positive or negative linear trend in each successive $2^N/2^{N_1}$ interval [11]. The slope of this trend depends on the gain error. A negative slope indicates a gain error less than the desired values and vice versa. The second stage gain error leads to another linear trend in each $2^N/2^{N_1+N_2}$ interval. The same process continues in later stages, but the resulting imperfections have a reduced effect on the overall performance. Therefore, the general structure of the HCF data (and consequently the HCF model) is primarily defined by the first stage error and only partially by the second. The flash offsets in each stage lead to the sliding (back and forth, depending on the polarity of the offsets) of the resulting linear trend, which results from this stage. The DAC level offsets are translated into the raising or lowering of the INL in the given interval against which this level is compared, i.e., a constant offset (positive or negative) is added in adjacent consecutive intervals of the INL. However, the effect of the offsets on the HCF structure and data decreases rapidly as the signal proceeds through the pipeline stages.

Fig. 2. A 12-bit pipeline ADC INL sequence measured for a 60 MHz input sine wave.

III. INTEGRAL NONLINEARITY ESTIMATION AND MODELING

A. Integral Nonlinearity

The INL is an ADC feature that labels its performance. For every ADC output code k , the INL is defined as the deviation of a practical ADC transition level from an ideal ADC transition level [1]:

$$i[k] = \frac{T_k - G \cdot T[k] - V_{os}}{Q} \quad [\text{LSB}] \quad (2)$$

where T_k and $T[k]$ are the ideal and practical quantizer transition levels, respectively. G and V_{os} are a gain and an offset term, respectively. The INL is generally expressed in terms of least significant bits [LSB]s. INL testing and characterization has been studied in [1], [12], [13], where the sine wave histogram test is presented along with other INL characterization procedures. The INL is characterized for pure sine wave inputs. The INL sequence of a 12-bit pipeline ADC (AD9430) for a 60 MHz input sine wave is shown in Fig. 2.

B. INL modeling preamble

Examining the INL sequence in Fig 2, one can discern two distinct entities: a smooth polynomial shaped carrier is present in addition to a fine structure with a quasi-repetitive scheme. The INL shows dependency on the test frequency, where the INL scheme is shown to markedly vary with respect to the frequency. In [14], it was demonstrated that the polynomial shaped carrier depends strictly on the input frequency, i.e., it is a dynamic feature. This polynomial scheme is related to the ADC input front-end circuit, which is sensitive to the signal frequency. As mentioned in the previous section, the HCF data experiences linear

trends in consecutive interval segments, primarily due to the first stage gain error. The work in [9] used this fact to develop an HCF model consisting of a set of adjacent, centered HCF segments. The segments were designed to have a single slope polarity, either positive or negative. The average INL model of [9] was as follows:

$$i_k = h_k + \ell_k = \underbrace{\eta_p k + \beta_p}_{\text{HCF model}} + \underbrace{\theta_0 + \theta_1 \bar{k} + \dots + \theta_L \bar{k}^L}_{\text{LCF model}} \quad (3)$$

where h_k is the HCF segment model comprising P segments (that is, $p = 1, \dots, P$) each with slope η_p and offset β_p . ℓ_k is the average LCF polynomial of order L with coefficients θ_l (for $l = 0, \dots, L$), and $\bar{k}$ is a normalized (between -1 and 1) version of k . However, due to the gain errors in the consecutive stages, which lead to additional linear trends with the different slope polarities (positive and negative) added to that of the DAC level offsets, the general scheme of the HCF in these intervals is not a well-defined (positive or negative) linear trend. It becomes a triangular form (or a form shaped like the Gaussian function in given consecutive intervals) with additional noise-like data. These structures can be observed in the INL data in Fig. 2. These characteristics of the HCF data led the HCF model in (3) to demonstrate a modest performance, necessitating the need for a new HCF modeling methodology, particularly one more capable of representing the HCF component. A low-order polynomial was reinstated to model the LCF in this work because it was shown to effectively represent the LCF data [9], [10], [14]. Two different models were developed to represent the HCF data $h[k]$.

IV. INL MODELING USING THE SINC AND GAUSSIAN BASIS FUNCTIONS

A. HCF modeling based on the sinc basis function

Inspired by the scheme of the HCF data and the pipeline ADC structure, an initial model composed of sinc functions, is proposed for the HCF model:

$$h_k = \alpha_1 \text{sinc}(\tilde{k} - c_1) + \alpha_2 \text{sinc}(\tilde{k} - c_2) + \dots + \alpha_P \text{sinc}(\tilde{k} - c_P) \quad (4)$$

where the c_p (for $p = 1, \dots, P$) are the center points of the sinc functions, and $\tilde{k}$ are a normalized version (which will be defined later) of the output code k . A sinc function with a zero center, i.e., $\sin(\pi x)/(\pi x)$, is shown in Fig. 3. The objective is to determine the necessary number of sinc functions and their respective center points. The solution lies in the circuitry of the pipeline ADC and the characteristics of the sinc function. The converter in the disposition (AD9430) is a $N = 12$ -bit pipeline ADC, i.e., it

Fig. 3. The sinc function.

has $K = 2^N = 4096$ different digital outputs. The first pipeline stage number of bits N_1 is equal to 4. As previously stated, it is not possible to assign a sole (positive or negative) trend to the HCF data. However, the INL data still demonstrate similar behaviors (triangular- or Gaussian-shaped patterns) in intervals measuring approximately $2^N/2^{N_1} = 256$ bits (where $N=12$ and $N_1=4$) in size due to the first pipeline stage gain error. Thus, given an N -bit ADC with an N_1 -bit first pipeline stage, one can select the size of each basis function interval as $2^N/2^{N_1}$. At this point, the basis function must be fit to each interval.

The sinc function is a form of $\sin(\pi x)$ that converges to zero and possesses a pseudo-period of length 1, as shown in Fig. 3. In each consecutive period between the zero crossings, we obtain a waveform that is significantly inferior in magnitude to the previous waveform. The function normalizes to one at $x = 0$. Because we are interested in modeling HCF linear- or triangular-shaped structure using a sinc function, each sinc function two-sided main period (x between -1 and 1 around the center point) must fit an HCF triangle interval of length $2^N/2^{N_1}$ (256 for this ADC). Thus, the number of sinc functions here is 16 ($=2^N/256$). Translating the linear code ($k = 0, \dots, 2^N - 1$) into a linear space from -16 to $+16$ ($4096/128=32$) gives the desired period intervals and the normalized code $\tilde{k}$. Having 16 sinc functions, their respective centers will lie at a normalized code distance of 2. Each of the sinc functions has two pseudo periods around this center, i.e. lying in $[-1 + c_p, 1 + c_p]$. As a consequence of these normalization and center point selections, every sinc function primarily depends on the code in its interval (the first sinc period) and is only partially dependent on the codes outside of the interval, particularly the distant ones. The first sinc function center (or peak) is taken to be the first maximum or minimum of INL data, and

the other centers occur automatically at $2^N/2^{N_1} = 256$ code distances. The first peak shift from $k = 0$ is primarily due to offsets in the flash comparators of the first pipeline stage. As previously mentioned, the size of these intervals does not precisely align with $2^{N_1} = 256$ codes, but the flexibility of this scheme (as will be shown in the forthcoming results) allows the length of the intervals to be automatically selected.

The LCF component is defined (in a similar vein to [9]) in (3). However, in this work, the polynomial order L is not optimized to render the HCF data centered around zero. There are no severe constraints on the latter to be centered, as required in [9]. The order L can be increased as long as it attains more of the INL data information (the LCF component). Increasing the order of the LCF polynomial leads to a reduced residual (for example in root mean square (rms) sense) after subtracting the estimated LCF polynomial from INL data. However, this residual is almost unchanged for orders L larger than 5 for the considered ADC. Hence, the latter value is adopted for the LCF polynomial order.

The INL model relates the average INL data $\bar{\mathbf{i}}$ in a vector format as

$$\bar{\mathbf{i}} = \mathbf{g}\boldsymbol{\eta} + \boldsymbol{\ell}\boldsymbol{\theta} + \text{res}, \quad (5)$$

where $\boldsymbol{\eta} = (\alpha_1, \dots, \alpha_p, \dots, \alpha_P)^T$ is a vector gathering the weights α_p given in (4) and $\mathbf{g}$ is a $[2^N - 1, P]$ dimensional matrix defined as

$$\mathbf{g} = \begin{pmatrix} \text{sinc}(-16 - c_1) & \cdots & \text{sinc}(-16 - c_P) \\ \vdots & & \vdots \\ \text{sinc}(\bar{k} - c_1) & \cdots & \text{sinc}(\bar{k} - c_P) \\ \vdots & & \vdots \\ \text{sinc}(16 - c_1) & \cdots & \text{sinc}(16 - c_P) \end{pmatrix} \quad (6)$$

and $\boldsymbol{\ell}$ is the Vandermonde matrix that contains the normalized codes $\bar{k}$

$$\boldsymbol{\ell} = \begin{pmatrix} 1 & -1 & \cdots & (-1)^L \\ \vdots & \vdots & & \vdots \\ 1 & \frac{k-2^{N-1}}{2^{N-1}-1} & \cdots & \left(\frac{k-2^{N-1}}{2^{N-1}-1}\right)^L \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & 1^L \end{pmatrix} \quad (7)$$

$\boldsymbol{\theta} = (\theta_0, \dots, \theta_L, \dots, \theta_L)^T$ summons the LCF polynomial coefficients, and res is a residual that includes the model imperfections, noise, etc.

Fig. 4. The normal distribution or Gaussian function used in this work.

B. HCF modeling by Gaussian basis functions

For a more accurate representation of the HCF data, one can add another sinc basis function between the already chosen functions, i.e., a sinc function at every 128 codes. In this approach, a sinc function is centered at every maximum and minimum of the HCF data, effectively doubling the number of required functions. For example, in Fig. 2, between the two maximums at approximately $k = 257$ and $k = 513$, a minimum comes at $k = 385$. However, this representation is not significantly improved because the convergence of the sinc function to zero is not fast enough, despite the normalizing of the output code k . Therefore, the HCF data from other intervals influence the HCF modeling in a given interval. To cancel the mutual dependency, other modular basis functions that have a more rapid convergence to zero) are better choices.

At this point, we introduce the Gaussian function or the normal distribution basis function. The normal basis function used here is defined as $f(x) = \exp(\frac{-x^2}{2})$. This function is plotted in Fig. 4. The magnitude of this Gaussian function is equal to one at its center, or for $x = 0$. It has a rapid convergence to zero (in comparison to the sinc function) and can be considered approximately equal to zero for $|x| > 3$. Using the function in Fig. 4 instead of the sinc function (see Fig. 3) renders the effect of the remote HCF bins obsolete if the output code k is properly normalized. To do so, every code interval of length 128 must be downscaled into a $[0, 3]$ or a $[-3, 0]$ interval, because the Gaussian function is almost zero outside of these intervals (refer to Fig. 4). Therefore, the ADC output code k is normalized to the interval $\tilde{k} = [-48, 48]$, following the same reasoning as in the case of sinc functions. Selecting the peaks or centers of the Gaussian basis functions is a straightforward task; the middle of every interval is a good choice. Practically speaking, the first center is selected at the middle of the given interval, and the rest

lie at 128 output code distances. Note that this modeling technique is also valid with a code interval of length 256 codes, although the model with the sinc basis function will perform better in this case.

The HCF model can now be written as

$$h_k = \alpha_1 \exp\left(\frac{-(\tilde{k} - c_1)^2}{2}\right) + \cdots + \alpha_P \exp\left(\frac{-(\tilde{k} - c_P)^2}{2}\right) \quad (8)$$

The model and data are entered into a system of equations in a similar fashion to the sinc function case (5), that is

$$\bar{\mathbf{i}} = \mathbf{g}\boldsymbol{\eta} + \boldsymbol{\ell}\boldsymbol{\theta} + \text{res}, \quad (9)$$

where $\mathbf{g}$ is a $[2^N - 1, P]$ dimensional matrix, defined here as

$$\mathbf{g} = \begin{pmatrix} \exp\left(\frac{-(-48-c_1)^2}{2}\right) & \cdots & \exp\left(\frac{-(-48-c_P)^2}{2}\right) \\ \vdots & & \vdots \\ \exp\left(\frac{-(\tilde{k}-c_1)^2}{2}\right) & \cdots & \exp\left(\frac{-(\tilde{k}-c_P)^2}{2}\right) \\ \vdots & & \vdots \\ \exp\left(\frac{-(48-c_1)^2}{2}\right) & \cdots & \exp\left(\frac{-(48-c_P)^2}{2}\right) \end{pmatrix} \quad (10)$$

C. Solution

The INL model counterparts for the sinc and Gaussian bases (5) and (9), respectively, can be estimated with the LS method out of the average INL data sequence $\bar{\mathbf{i}}$, in a similar vein to [9], [15]. First, the estimated weights of the sinc or Gaussian functions become

$$\hat{\boldsymbol{\eta}} = (\mathbf{g}^T \boldsymbol{\pi}_\perp \mathbf{g})^{-1} \mathbf{g}^T \boldsymbol{\pi}_\perp \bar{\mathbf{i}} \quad (11)$$

where $\boldsymbol{\pi}_\perp$ is a projection matrix

$$\boldsymbol{\pi}_\perp = \mathbf{I} - \boldsymbol{\ell}(\boldsymbol{\ell}^T \boldsymbol{\ell})^{-1} \boldsymbol{\ell}^T \quad (12)$$

The estimated average LCF model or polynomial coefficients

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{\ell}^T \boldsymbol{\ell})^{-1} \boldsymbol{\ell}^T (\bar{\mathbf{i}} - \mathbf{g}\hat{\boldsymbol{\eta}}) \quad (13)$$

Finally, the estimated INL model (using the sinc or Gaussian functions bases) for a given average INL data can be written as

$$\hat{\mathbf{i}} = \mathbf{g}\hat{\boldsymbol{\eta}} + \boldsymbol{\ell}\hat{\boldsymbol{\theta}} \quad (14)$$

V. THE MODELING AND CALIBRATION RESULTS

In this section, we compare the two HCF modeling techniques with respect to the HCF data. The HCF data, denoted as $h[k]$, can be obtained by subtracting the estimated average LCF polynomial $\ell\hat{\theta}$ from the average INL data $\bar{i}$. The centered HCF segment model of [9] is also included in the comparison. As previously mentioned, the Gaussian function-based model demonstrates superior performance for narrow intervals, whereas the sinc function-based model performs better in cases of wider intervals; therefore, the modeling results will only be presented for those cases.

A. Test-Bed

Two samples of the ADC (AD9430) were tested, and the measured data are modeled. The ADCs are denoted as ADC1 and ADC2, respectively. The Rohde&Schwartz SMU200A vector signal generator (VSG) is used to generate the clock and input signals to ADC under test. A noncommercial wideband ultra low distortion amplifier with a gain of 14.3 dB is used at the output of the VSG to avoid overdriving the latter. Mini-Circuits SLP filters are used at the input of the ADC under test to attenuate all the harmonics and spurious signals to be below -80 dBFS at least (verified by the signal analyzer). Data are collected from the ADC by a frame grabber and analyzed offline. More details concerning the device under test and the test-bed can be found in [9]. The work in this section was performed using Matlab software.

B. Modeling results

A polynomial of order $L = 7$ is used to obtain the centered HCF data. An order of L equal to 5 would have provided similar results, but for a similar comparison with the model in [9], the order $L = 7$ is employed here. A total of 16 sinc basis functions and 32 Gaussian basis functions are used for both ADCs. In both ADCs, the first extremum is a positive peak, indicating a negative gain error in the first pipeline stage.

In the case of ADC1, the results of the HCF modeling using the sinc and Gaussian basis function are shown in Fig. 5 and 6, respectively. The results from the HCF modeling of ADC2 are shown in Fig. 7 and 8. In general, ADC2 demonstrated more efficient modeling schemes (in terms of representing the HCF data information) because its HCF data are more structured, indicating a significant dominance (compared to ADC1) of first-stage errors, especially the (negative) gain error. In comparison with the HCF segment model in [9], the schemes here demonstrate a superior ability to grasp the extremes in the

Fig. 5. The ADC1 HCF data are noted in red, and the sinc function-based model is given in blue.

Fig. 6. The ADC1 HCF data are noted in red and the Gaussian function-based model is given in blue.

HCF data, thus enabling a more efficient representation of the error and a superior calibration (when used in compensation), as will be shown later.

C. Model efficiency

The ability of the HCF model to represent the data is tested with rms calculations. The rms of the HCF data and the residual rms after subtracting the HCF models are compared. The centered segment-model given in [9] is also included in the comparison, which is performed for both ADCs.

From Tab. I, it is clear that the use of Gaussian basis functions produces superior results than the sinc function model, which has fewer parameters. The choice between the two models is application dependent; the criteria determine the model selection, either simplicity (in the number of parameters, for example) or the accuracy of the data representation. The segment model in [9] provided smaller residuals than both HCF models in this paper for ADC1. However, we recall that the structural derivation for the

Fig. 7. The ADC2 HCF data are noted in red, and the sinc function-based model is given in blue.

Fig. 8. The ADC2 HCF data are noted in red, and the Gaussian function-based model is given in blue.

model in [9] was quite lengthy, and a minimization criterion was implemented to obtain the number of segments and their sizes. These actions are not performed in the straight-forward methods developed here, particularly the minimization process. However, the HCF data of ADC1 is very small in magnitude and it was shown that the ADC1 HCF has no perceivable effect on the (static) calibration in [10]. In general, the Gaussian function-based model yielded better modeling results than the sinc approach because it has

TABLE I
THE CAPABILITY OF EACH HCF MODEL TO REPRESENT THE HCF DATA IN TERMS OF THE RMS CALCULATIONS.

HCF data and models	ADC1	ADC2	Number of parameters
HCF data	0.14	0.30	$2^N - 1 = 4095$
Segments [9]	0.10	0.23	16
Gaussian	0.11	0.18	32
Sinc	0.12	0.22	16

twice the number of parameters.

D. Modeling performance in terms of static calibration simulations

These modeling techniques presented the HCF data quite well. We have every reason to believe that our techniques will induce performance improvements when used for calibration because they exhibit a superior HCF representation in comparison to [9], on which a model-based calibration approach was developed, in [2]. The latter model led to performance improvements in terms of reductions in the dynamic and static errors. The effect of these HCF modeling approaches can be tested using a model-based (either sinc or Gaussian) static calibration on ADC1 and ADC2; however, the results will not provide an unquestionable proof that these modeling methods work for all pipeline ADCs of the same type. **More actual ADCs should be tested to ensure that the HCF modeling techniques are completely valid for any pipeline ADC model-based static calibration. However, testing more ADCs will consume considerable time with no guaranteed significant scientific contribution. Due to the random nature of the static errors (i.e. stage gains, flash comparators' offsets and DAC levels' errors), acquiring some more ADCs may not add much to the diversity of the INL data scheme. ADC1 is emblematic example of this theory: its static INL data (HCF) is quite small in magnitude; therefore it leads to no tangible static calibration results, while its modeling is quite efficient.** Fortunately, this study is focused on pipeline ADCs; in this structure, the origins of static errors are well known to be: the stages gain errors, flash comparator offsets and DAC offsets, as previously noted. To further test the ability of the developed modeling techniques to post-correct the potential pipeline ADC at hand, Monte Carlo simulations can be run in which the pipeline structure errors are defined as random variables with a given (the desired or design value) mean and a typical variance. **The simulations provide a much wider base for statistical analysis and considerations and are a good alternative to testing several additional off-shelf ADCs.** The obtained HCF sequence (which, in this case is the INL sequence because no dynamic errors are assumed) in each simulation is then modeled, and the respective model is used to compensate the ADC in a similar fashion to that given in (1). The HCF model performance is also compared to the HCF data performance in compensation. The simulation results are important for providing enhanced understanding regarding the expected performance improvements, that this method would provide if it is applied (by a manufacturer, for example) to a large number of produced pipeline ADCs.

Both HCF modeling techniques (the sinc and Gaussian bases), with normalized intervals of 128 and 256 code lengths are used. One thousand Monte Carlo simulations are run, in which there is no adequate

possibility to acquire the first maximum or minimum peak of the synthetically generated HCF data. The first peak would range between $k = 1$ and $k = 128$, depending primarily on the flash comparator offsets in the first stage. Therefore, the first extremum, denoted as K_1 , is stepped in the simulations at the code position of $k = 1, 32, 64, 96$.

E. Calibration results

Each constructed (by Matlab) ADC is compensated by its respective INL data and model, in a similar fashion to that of (1). In the latter equation, the quantity $i[m, k]$ is replaced by the HCF data $h[k]$ and model h_k , i.e., a static LUT and model calibration. The results are analyzed in terms of the spurious free dynamic range (SFDR), which is the most important ADC figure of merit, especially when dealing with static compensation.

The results of the various simulations are summarized in Tab. II. It can be observed that the HCF models (both the sinc and Gaussian based) with an interval code length of 128 outperform the HCF models with an interval code length of 256. Both basis functions yield similar results for similar interval code lengths, which was unexpected at this stage. It was assumed earlier that the sinc model performs better with longer code intervals, whereas the Gaussian model provides better results with smaller intervals. However, one can explain the similar performance of the sinc and Gaussian models with the 128 code lengths as resulting from the higher sensitivity of the Gaussian model to the selection of the first extremum and, consequently, to the remaining peaks. The sinc model is more flexible and can adapt better to poor selections of the basis function centers. If the first peak or minimum is adequately selected (which is nearly impossible in the simulations), the Gaussian model performs better, as experienced with the measured INL data for both tested ADCs. In general, both models enable enhanced performance improvements if proper peak selection is allowed, thereby becoming more comparable to the HCF data performance. Another reason for the reduced model performance (in comparison to the data) is the several cases in which the HCF is rather unstructured or is noisy in shape. These factors are the results of unconventional gain errors in later pipeline stages, that could be significantly large in comparison to the first stage gain error. These cases are highly improbable and are generally calibrated (in-chip) inside the ADC (prior to the post-correction that is performed in this work); however, they cannot be separated from the simulations. Note that no quantization noise is added to the quantization process in these simulations. If quantization noise is added, the improvements obtained using the INL data would tangibly diminish, and the model and data

TABLE II
THE SFDR IMPROVEMENTS IN dB OBTAINED USING THE DIFFERENT HCF MODELS.

Models	$K_1=1$	$K_1=32$	$K_1=64$	$K_1=96$
Gaussian 128	9	9.4	9.5	9.1
Sinc 128	9.2	9.3	9.4	9.2
Gaussian 256	4.9	5.5	5.2	5
Sinc 256	4.5	4.8	5.3	5
HCF data	12.7	12.7	12.7	12.7

Fig. 9. The SFDR of the original sequence is compared against the SFDR of the calibrated sequence using the LUT and Gaussian model-based method with 128 code length. Only 100 realizations are shown for clarity.

calibration results would converge. In general, the models using a normalized code length of 128 show superior performance of approximately 4 dB in comparison to those with a code length of 256.

The results of the (constructed) pipeline ADC static calibrations using Gaussian basis functions with a normalized code length of 128 and the first extremum chosen at $k = 64$ are shown in Fig. 9. It can be observed that the model-based calibration always leads to performance improvements in terms of the SFDR. The model-based calibration is generally inferior to the LUT-based calibration, although they are identical in many cases. The LUT-based calibrations have an average SFDR improvement of 12.7 dB, while the average improvement is 9.5 dB for the model-based calibrations. The histogram of the model-based improvements is plotted in Fig. 10, showing the distributions of the SFDR improvements.

We show an additional plot in Fig. 11, in which the HCF data exhibit an unorthodox scheme. Even so, the model at hand (the Gaussian basis model) can adequately represent the latter due to its flexibility, despite the fact that it is not tailored for such data schemes. The SFDR improvements due to calibration with the HCF data and model are 12.1 and 11.9 dB, respectively.

Fig. 10. The histogram of occurrence of the SFDR improvements for the Gaussian function model with a normalized code length of 128. An average of 9.5 dB is obtained.

Fig. 11. The synthetically generated HCF data modeled using a Gaussian basis function model with a normalized code length of 128.

F. Static Calibration enhancement on the pipeline ADC at hand

The effects of static LUT- and model-based calibrations on the pipeline ADC at hand are also tested. To completely eradicate the dynamic error (which originates from the front-end input of the pipelined structure), a synthetic digitized ADC output is generated with the HCF data (of ADC2) and an additive white Gaussian noise (AWGN) are added to the quantization process. The AWGN is power regulated so that the noise floor of the synthetic data equals the noise floor of the measured data.

The frequency response of such an output is shown in Fig. 12. The spurious signals for this test frequency (30 MHz) appear around the second and third harmonic signal bins. The harmonic signals, being strictly related to dynamic errors [14], [16], are nonexistent in this spectrum (i.e., the double and

Fig. 12. The achievable performance improvement in terms of the SFDR based on the LUT and the Gaussian basis function model (code length 128) calibrations.

triple fundamental frequency bins are at the noise floor in this spectrum) because we initially assumed that no dynamic errors were present. The calibrated sequence frequency responses using the LUT and model based approaches are also shown in Fig. 12. The model-based compensation has led to the suppression of spurious signals or attenuations of these signals to the noise level vicinity, achieving an SFDR improvement of 10.2 dB. The LUT enables the total (static) calibration of the output; however, it adds no more than 0.3 dB to the model-based calibration, which uses far fewer parameters. The LUT (which is identical to that given in [10]) contains $2^N - 1 = 4095$ parameters, whereas the Gaussian model-based calibration is constructed of only 32 parameters. The HCF segment model in [9] achieved a compensation of only 4.3 dB SFDR calibration improvement when used in [2].

VI. CONCLUSIONS

Static INL modeling was performed in this paper, whereby the HCF component of the INL was modeled using both sinc and Gaussian basis functions. The model coefficients were estimated using the LS method. The developed models required minimal knowledge of the pipeline architecture for deriving their structures. Both HCF modeling techniques represented the HCF data quite well by using only few parameters. The HCF modeling was also performed for synthetically generated HCF data from pipeline ADCs constructed in Matlab. The estimated models were used to post-correct the respective ADCs. The results demonstrated that the estimated HCF models always lead to performance improvements in terms of the SFDR. These improvements are similar to those obtained when the HCF data are used in the calibrations. Furthermore, the HCF of one investigated commercial ADC was combined with some AWGN to the quantization

process of an ideal ADC, and the respective HCF model was used to calibrate the synthetic ADC data. The SFDR improvement results exhibited nearly complete calibration of the ADC, which demonstrated additional support for the increased efficiency of the HCF modeling technique.

REFERENCES

- [1] *IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters*, IEEE Standard 1241-2010.
- [2] S. Medawar, P. Händel, N. Björzell and M. Jansson "Postcorrection of pipelined analog-digital converters based on input dependent integral nonlinearity modeling," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, pp 3342 - 3350, Oct. 2011.
- [3] P. Arpaia, P. Daponte and L. Michaeli, "A dynamic error model for integrating analog-to-digital converters," *Measurement* 25, pp. 255-264, June 1999.
- [4] L. Michaeli, P. Michalko and J. Saliga, "Identification of ADC error model by testing of the chosen code bins," *12:th IMEKO TC4 International Symposium*, September 25-27, 2002, Zagreb, Croatia.
- [5] L. Michaeli, P. Michalko, and J. Saliga, "Identification of unified ADC error model by triangular testing signal," *10th Workshop on ADC modelling and testing*, Gdynia/Jurata, Poland, pp. 605-610, 2005.
- [6] A. Cruz Serra, F. Alegria, L. Michaeli, P. Michalko, J. Saliga, "Fast ADC testing by repetitive histogram analysis," *IEEE Instrumentation and Measurement Technology Conference*, April 24-27, 2006, pp. 1633-1638.
- [7] L. Michaeli, P. Michalko, J. Saliga, "Unified ADC nonlinearity error model for SAR ADC," *Measurement* Vol. 41, Issue 2, pp. 198-204, February 2008.
- [8] A.C. Serra, M.F. da Silva, P.M. Ramos, R.C. Martins, L. Michaeli and J. Saliga, " Combined spectral and histogram analysis for fast ADC testing," *IEEE Transactions on Instrumentation and Measurement*, Vol. 54, No. 4, pp. 1617-1623, August 2005.
- [9] S. Medawar, P. Händel, N. Björzell and M. Jansson, "Input dependent integral nonlinearity modeling for pipelined analog-digital converters," *IEEE Transactions on Instrumentation and Measurements*, vol. 59, pp 2609 - 2620, Oct. 2010.
- [10] S. Medawar, P. Händel, B. Murmann, N. Björzell and M. Jansson, "Dynamic Calibration of Undersampled Pipelined ADCs by Frequency Domain Filtering," *IEEE Trans. Instrum. Meas.*, available online April 2013, DOI: 10.1109/TIM.2013.2248289.
- [11] T. Kuyel and H. Bilhan, "Relating linearity test results to design flaws of pipelined analog-to-digital converters," in *Proc. Int. Test Conf.*, Atlantic City, NJ, pp. 772-779, Sept. 1999.
- [12] S. Rapuno, P. Daponte, E. Balestrieri, L. de Vito, S. J. Tilden, S. Max and J. Blair, "ADC parameters and Characteristics – part 6 in a series of tutorials in instrumentation and measurement," *IEEE Instrumentation & Measurement Magazine*, Vol. 8, No. 5, pp. 44–54, December 2005.
- [13] T.E. Linnenbrink, J. Blair, S. Rapuno, P. Daponte, E. Balestrieri, L. de Vito, S. Max and S. J. Tilden, "ADC testing – part 7 in a series of tutorials in instrumentation and measurement," *IEEE Instrumentation & Measurement Magazine*, Vol. 9, No. 2, pp. 39–49, April 2006.
- [14] S. Medawar and P. Händel, "On the Calibration of Wide-Band Analog-Digital Converters," *IEEE Trans. Instrum. Meas. letters.*, available online May 2013, DOI: 10.1109/TIM.2013.2255991.
- [15] P. Händel, N. Björzell and M. Jansson, "Model based dynamic characterization of analog-digital-converters at radio frequency— Invited paper," *2007 International Conference on Signal Processing and its Applications*, 12-15 February 2007, Sharjah, UAE.
- [16] N. Björzell and P. Händel, "Achievable ADC Performance by Postcorrection Utilizing Dynamic Modeling of the Integral Nonlinearity," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, Article ID 497187, 10 pages, 2008. doi:10.1155/2008/497187