Background Calibration of Pipelined ADCs Via Decision Boundary Gap Estimation

Lane Brooks, *Student Member, IEEE*, and Hae-Seung Lee, *Fellow, IEEE*

***Abstract*—A method of indirect background digital calibration of the dominant static nonlinearities in pipelined analog-to-digital converters (ADC) is presented. The method, called decision boundary gap estimation (DBGE), monitors the output of the ADC to estimate the size of code gaps that result at the decision boundaries of each stage. Code gaps result from such effects as capacitor mismatch, finite opamp gain, finite current source output impedance, comparator offset, and charge injection. DBGE does not require special calibration signals or additional analog hardware and can even reduce the performance requirements of the analog circuitry. The calibration is performed using the input signal and thus requires that the input signal exercise the codes in the vicinity of the decision boundaries of each stage. If it does not exercise these codes, then lack of calibration is less critical because the nonlinearities will not appear in the output signal. DBGE is simple and amenable to hardware and/or software implementations. Simulation results indicate DBGE is highly accurate, robust, and adaptive even in the presence of parameter drift and circuit noise.**

***Index Terms*—Adaptive digital background calibration, capacitor mismatch, finite opamp gain, pipelined analog-to-digital converter (ADC), static nonlinearity.**

I. Introduction

Pipelined analog-to-digital converters (ADCs) are popular for many applications because they can realize high throughput and high resolution simultaneously. CMOS switched-capacitor-based implementations have been widely researched and used in industry. In the absence of trimming or calibration, these implementations typically suffer from static nonlinearities that limit the resolution to 8 to 10 bits [1]–[3].

These nonlinearities have spurned many circuit and calibration techniques for realizing higher resolutions. Analog circuit techniques such as those in [4] and [5] use analog components in the signal path to generate higher linearity at the expense of conversion speed. Digital calibration techniques, which realize the benefits of device scaling, have also been developed and can be categorized into foreground and background techniques.

Foreground calibration measures nonlinearities during a calibration phase which usually occurs during startup. The method demonstrated in [2] measures the nonlinearities by driving the bit decision boundary conditions during calibration to measure the nonlinearities. Many other test-based or statistical-based methods have been developed that measure the nonlinearities using code density or histogram measurements. For example, in [6], the reference voltages of the last pipeline stage are laser trimmed to produce ideal code densities. Likewise, in [7]–[10], digital correction is performed based on foreground code density measurements of the nonlinearities. Since these techniques use foreground calibration, they require interrupting normal ADC operation for calibration. To minimize the interruptions, the calibration phase can be limited to manufacturing or ADC startup, but then calibration drift can result.

In contrast, background techniques operate calibration circuits continuously and transparently so that users do not see service interruption. One class of background calibration measures circuit errors with calibration signals during hidden calibration time slots. A "skip-and-fill" approach is used in [5] where the input samples are interpolated during the hidden calibration phase. A queue-based approach is used in [11]. The drawback of these approaches is that they require redundant channels/stages and/or their overall accuracy is a function of the coverage of the calibration signal, which cannot follow the same path as the signal exactly. Another popular background calibration approach, called gain error correction (GEC) [12]–[16], additively injects an uncorrelated analog calibration signal into the ADC during normal operation. The known calibration signal is then subtracted from the ADC output and the calibration parameters are adjusted to null the correlation of the calibration signal to the corrected ADC output. Since the signal path must be able to accommodate the superposition of the input and the calibration signal, these techniques either reduce the available signal range or over-range protection of the ADC. Furthermore, its accuracy is tied to accuracy of the injected analog calibration signal.

Indirect methods of background calibration overcome the calibration signal coverage and accuracy issues by estimating the errors from the input signal itself without the use of calibration signals. In [1] and [17], the dominant nonlinearities of pipelined ADCs are modeled and corrected using adaptive equalization techniques prevalent in digital communications. It requires an additional "slow-but-accurate" ADC for reference to estimate and correct the errors. In [18] they note that when an input signal has a lowpass input histogram, the nonlinearities of the ADC will generate high-pass components in the output histogram. Thus, they collect an output histogram, lowpass filter it, and generate a correction map from the raw histogram space into the smoothed histogram space. In [19], they also use code densities or histograms with a second ADC to generate a correction map. These techniques are to varying degrees either algorithmically or hardware intensive.

Manuscript received October 30, 2007; revised January 4, 2008 and March 7, 2008. First published April 18, 2008; current version published November 21, 2008. This work was supported in part by NDSEG, CICS, the National Defense Science and Engineering Graduate Fellowship (NDSEG), the MIT Center for Integrated Circuits and Systems, and DARPA under Grant N66001-06-1-2046. This paper was recommended by Associate Editor J. Silva-Martinez.

The authors are with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: lbrooks@mit.edu).

Digital Object Identifier 10.1109/TCSI.2008.925373

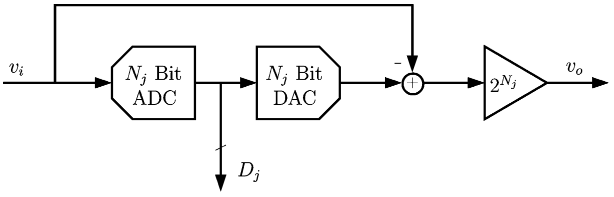

Fig. 1. Block diagram of an N_j bit/stage pipeline stage.

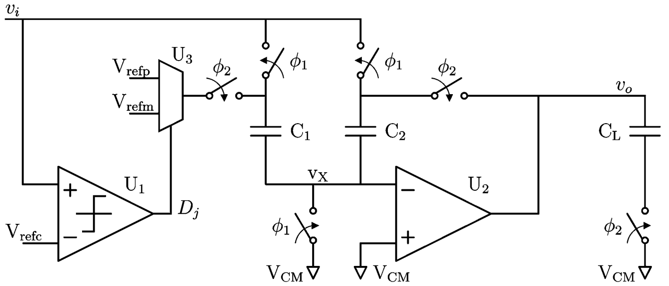

Fig. 2. Typical opamp-based circuit implementation of 1-bit/stage pipeline stage. Single-ended version shown for simplicity.

Indirect calibration requires making assumptions about the input signal and possibly the errors themselves. For example, [18] assumes the input signal distribution is low-pass. The technique presented here is called decision boundary gap estimation (DBGE) for indirect digital background calibration. DBGE removes the dominant nonlinearities of pipelined ADCs that appear as code gaps at decision boundaries. DBGE, therefore, models these gaps and relies on the input signal to exercise the codes in the neighborhood of these gaps to estimate and remove them. Much like the test-based or statistical-based methods, this technique estimates the nonlinearities using code-density measurements. The estimation techniques, however, only require code-densities measurements in the regions surrounding the bit decisions of each stage and have been developed to run continuously in the background using the input signal itself as the stimulus rather than calibration signals.

The remainder of this paper is organized as follows. Section II presents the error models which DBGE uses. Section III introduces the digital correction method on which DBGE relies. Several different error estimation techniques with their associated trade-offs are presented in Section IV and simulation results are shown in Section V. Finally, conclusions and discussions follow in Section VI.

II. Pipelined ADC Error Models

A pipelined ADC consists of low resolution stages, as shown in Fig. 1, concatenated together to form the desired resolution. Initially consider the case when the resolution N_j of the sub-ADC and sub-DAC in each stage is 1. This forms a 1-bit/stage pipelined ADC. A typical opamp-based switched capacitor implementation of a 1-bit/stage pipeline stage is shown in Fig. 2, and a zero-crossing-based implementation [20] is shown in Fig. 3. For either implementation, the ideal voltage or residue transfer of a single stage can be expressed mathematically as

$$v_o = 2v_i - dv_{\text{ref}}$$

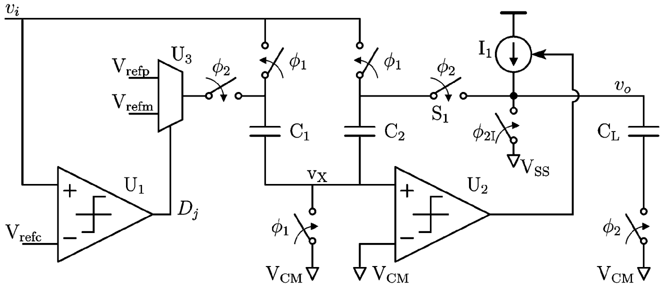

Fig. 3. Typical zero-crossing-based circuit implementation of 1-bit/stage pipeline stage. Single-ended version shown for simplicity.

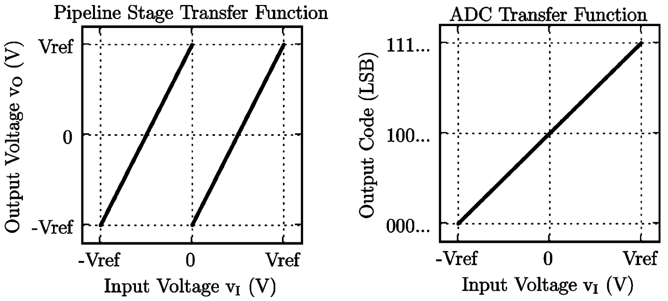

Fig. 4. Ideal stage voltage transfer function (left) and ADC transfer function (right).

where $d = 1$ when the comparator output D_j is high and $d = -1$ when D_j is low. This result along with the resulting ideal digital output is plotted in Fig. 4. Effects such as capacitor mismatch, finite opamp gain (opamp-based implementation), finite current source output impedance (zero-crossing-based implementation), comparator offset, and charge injection often cause static nonlinearities that limit the resolution of pipelined ADCs [1]–[3], [20], [21]. An analysis of each of these effects reveals they each produce similar nonlinearities in the form of either missing or wide codes at the bit decision boundaries of the sub-ADC.

A. *Capacitor Mismatch*

Capacitor mismatch results when capacitors C_1 and C_2 shown in Figs. 2 and 3 are not equal. If we define the amount of capacitor mismatch as $\epsilon = C_1/C_2 - 1$, then the resulting voltage transfer function becomes

$$v_0 = (2 + \epsilon)v_i - (1 + \epsilon)dv_{\text{ref}}.$$

If ϵ is negative, then a code gap results at the decision boundary of the digital output as depicted in the right plot of Fig. 5. This shows how the negative capacitor mismatch lowers the gain of the amplifier. If ϵ is positive, then a duplicate or wide code region results in the digital output transfer function as depicted in Fig. 6. Here the mismatch increases the gain of the amplifier.

Capacitor mismatch calibration techniques have been studied extensively as historically capacitor mismatch has been the most significant artifact limiting pipelined ADC resolution. Some calibration techniques such as those in [22], [23] are only effective at removing the effects of capacitor mismatch. More recently, however, as technology scaling has reduced voltage supplies and intrinsic device gain, finite opamp-gain has emerged as another major issue such that some calibration techniques such as GEC

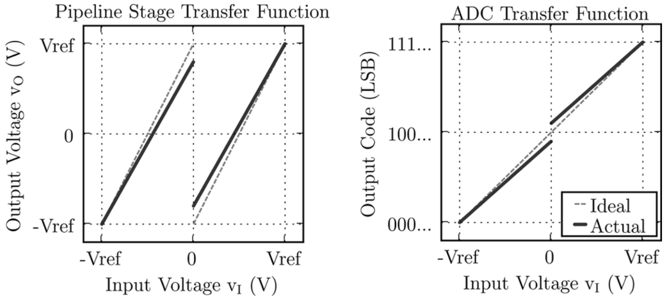

Fig. 5. Single stage and ADC transfer function from capacitor mismatch when $\epsilon < 0$.

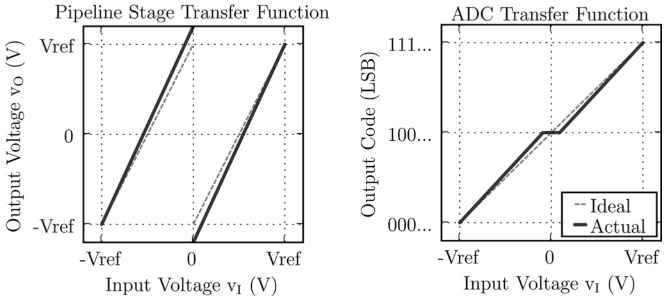

Fig. 6. Single stage and ADC transfer function from capacitor mismatch when $\epsilon > 0$.

[14], [16] correct other issues while relying on accurate capacitor matching. This transition away from capacitor matching as the dominant issue is perhaps due the continuing improvement of lithographic tolerances at each technology node, the requirement for increased total capacitance to maintain the same SNR at decreased voltage supplies, and the rise of finite opamp gain as a significant issue in scaled technologies.

B. Finite Opamp Gain

Finite open-loop opamp gain produces an effect that is similar to capacitor mismatch. If the opamp open-loop gain is A, then the voltage transfer function becomes

$$v_o = \frac{2v_i - dv_{\text{ref}}}{1 + \frac{2}{A}}. \quad (1)$$

This shows that the output voltage v_o depends on A such that the ideal gain of 2 is attenuated by its inverse. Therefore, a designer must ensure that A is large enough to meet the desired linearity requirements[1]. As device technology continues to scale, realizing opamps with sufficient gain and bandwidth has become increasingly difficult. An example of the system response to an opamp with insufficient open-loop gain is shown in Fig. 7. The result is a missing code gap in the ADC transfer function at the bit decision boundary.

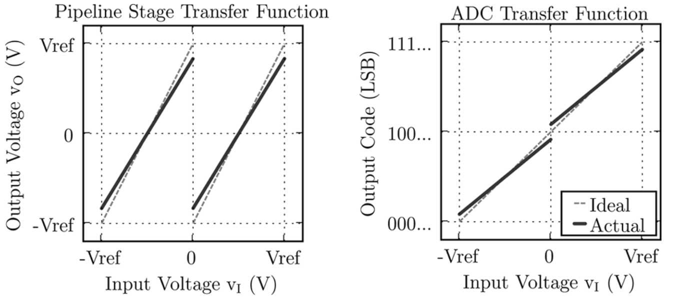

Fig. 7. Single stage and ADC transfer function from finite opamp gain or finite current source output impedance.

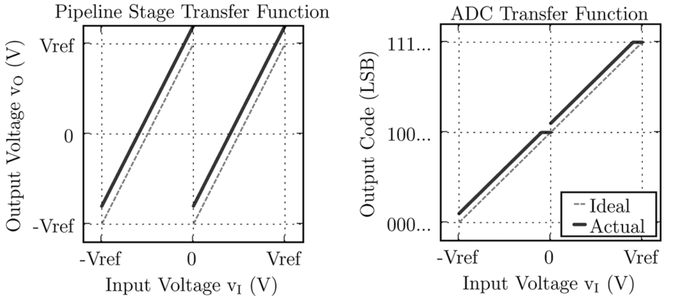

Fig. 8. Single stage and ADC transfer function from positive charge injection or stage transfer offset.

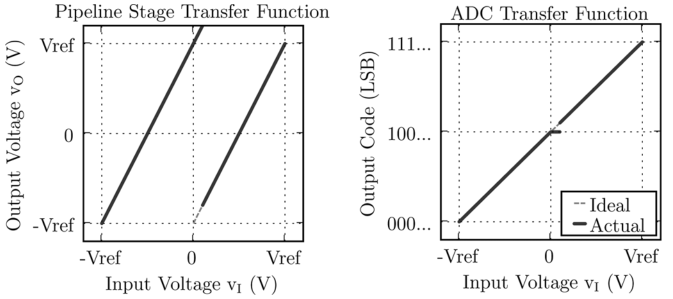

Fig. 9. Single stage and ADC transfer function from a positive bit decision comparator offset.

C. Finite Current Source Output Impedance

When zero-crossing-based circuits are used to realize the charge transfer then the finite output impedance of the current source and the finite delay of the zero-crossing detector will produce an effect that is very similar to finite gain in an opamp-based circuit. The finite output impedance of the current source can be captured by its effective Early voltage V_A. In [20] the residue voltage is found to be

$$v_o = \frac{2v_i - dv_{\text{ref}}}{1 + \frac{\Delta V}{V_A}} \quad (2)$$

where ΔV is the baseline voltage overshoot due to the finite delay of the zero-crossing detector. Since this result has the same form as (1), the transfer functions of Fig. 7 also apply to this case.

D. Offset Errors

Charge injection, opamp offset, zero-crossing detector offset, and bit-decision comparator offset produce wide code effects at the bit decision boundary as shown in Figs. 8 and 9.

[1] Nonlinear opamp gain can also cause static nonlinearity and is not considered here as it does not produce code gaps at the bit decision boundaries of the ADC.

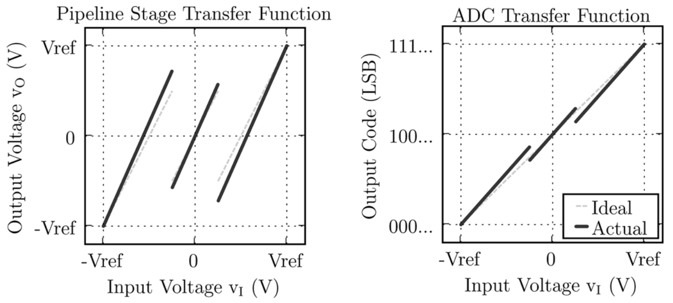

Fig. 10. Single stage and ADC transfer function in a 1.5-bit/stage ADC with capacitor mismatch when $\epsilon > 0$.

E. Redundancy

All the effects previously analyzed have the similarity that the produce missing or wide codes at the digital boundaries produced by the bit-decision comparator that makes up the sub-ADC of each stage. As will be shown in Section III, the nonlinearity produced by a missing code gap can be easily corrected in the digital domain. Wide codes, on the other hand, cannot be corrected as easily. Thus, to use DBGE on a 1-bit/stage ADC, the radix or gain of each stage must be intentionally reduced as in [2] to ensure that even under worst case capacitor mismatch, finite opamp gain, finite output impedance, charge injection, and comparator offset that the resulting nonlinearity is a missing code gap rather than a wide code.

Wide codes result when residue voltage v_o goes out of range. Without redundancy, the radix must be reduced to ensure this does not happen. Redundancy in the sub-ADC and sub-DAC [24], however, can be employed instead of radix reduction to keep the signal from going out of range and producing a wide code. Redundancy causes duplicate or overlapping code gaps rather than wide codes. This is shown in the example transfer functions of Fig. 10 where a 1.5–bit/stage ADC is used with a positive capacitor mismatch. Comparing this ADC transfer function with that of Fig. 6 shows redundancy transforms wide code regions into duplicate code regions. The duplicate code regions can be corrected in the same way as missing code regions.

F. Errors From Multiple Stages

The preceding examples showed the ADC transfer function when only the first stage had the static nonlinearity and the remaining stages were ideal. The effect of each additional stage, however, will also manifest itself as shown in the ADC transfer function in Fig. 11 where the first two stages are given the same low finite opamp gain. The missing code gap from the first stage is the largest and in the middle at the bit decision boundary of the first stage. The missing code gap from the second stage further divides each segment and produces a gap half the size of the first stage at the bit decision boundaries of the second stage. The missing code gap from each additional stage will continue to be half that of the previous stage and further subdivide each segment. As the code gap halves in size for each stage, at some stage the gap size will become smaller than the resolution of the ADC and will produce undetectable effects. These stages at the end of the pipeline can be considered ideal in terms of linearity and allow for the correction of the stages that precede them.

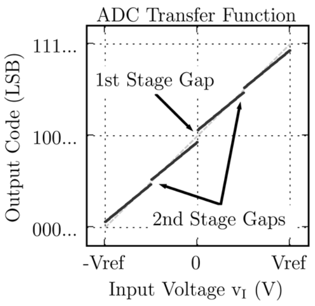

Fig. 11. ADC transfer function when first two stages have finite opamp gain.

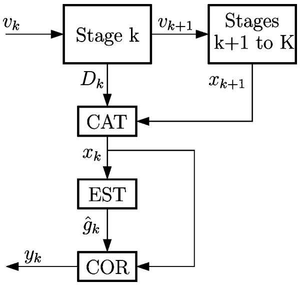

Fig. 12. Block diagram of correction scheme for a single stage.

III. Gap Correction

The calibration procedure of DBGE can be broken into two steps. The first is an *Estimation* phase where the digital output of the ADC is used to estimate the size of the missing code gaps for each stage. The second step is a *Correction* phase where the gaps are digitally removed from the raw samples. The correction technique is described first in this section under the assumption that accurate gap estimates have been measured. The following Section then describes gap estimation techniques of DBGE.

The resolution of a pipelined ADC is set by the number of stages. Suppose that an ADC with B stages is limited in resolution such that the first k stages need calibrated due to any number of the circuit issues aforedescribed. This means that the last $B - k$ stages produce a linear output that does not contain any missing code gaps.

Calibration starts with stage k. The block diagram of Fig. 12 shows the calibration procedure. When stage k produces a bit decision output D_k, it is combined with the reconstructed output of the later stages to produce the raw sample x_k. x_k is passed to the estimator to produce an estimate of the gap size. Assuming the estimator produces a good estimate $\hat{g}_k$ of the gap size, then the nonlinearity is removed from x_k by subtracting $\hat{g}_k$ from all samples above the gap. Expressed mathematically, the linearized or corrected sample y_k is

$$y_k = \begin{cases} x_k & \text{when } D_k = 0 \\ x_k - \hat{g}_k & \text{when } D_k = 1. \end{cases} \tag{3}$$

An example of a raw and corrected ADC transfer function is plotted in Fig. 13. The dashed line represents the raw data and contains a missing code gap at bit decision boundary of the first stage. The solid line shows the corrected response. Observe that the gap or nonlinearity has been removed but that the transfer function does not completely match the ideal response. In fact, the resulting response has a residual offset and gain error. This

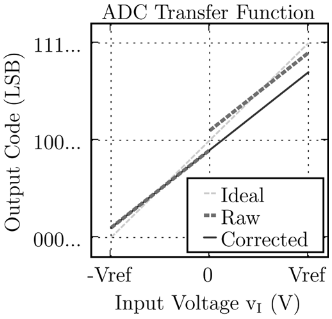

Fig. 13. Transfer function of raw and corrected samples.

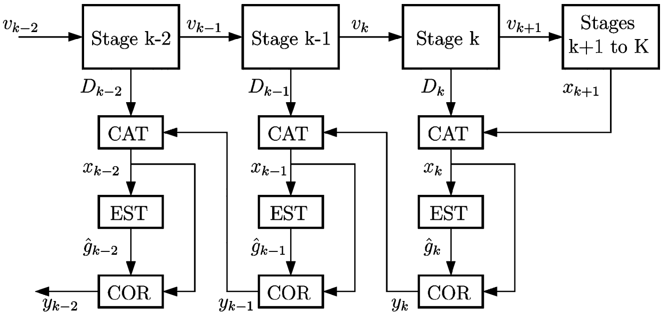

Fig. 14. Block diagram of concatenated stages utilizing DBGE.

residual offset and gain error is not an issue for many ADC applications as they do not cause any nonlinear effects. However, for some applications, such as time-interleaved ADCs, an unknown offset and gain is not tolerable and will need further correcting with other techniques such as those presented in [25].

After correction, sample y_k is free of the nonlinearity that was limiting the overall resolution, and the preceding stage $k-1$ can then be corrected in the same manner as stage k by using the corrected sample y_k. This will produce the corrected sample y_{k-1} which can then be used by stage $k-2$. A block diagram depicting this scheme of successive stage calibration is shown in Fig. 14.

One can use the this correction scheme for as many stages as necessary. If bit decision gaps were the only nonlinearity in the ADC implementation, then this procedure could be used to achieve any arbitrary resolution. In practice, however, eventually other sources of nonlinearity, such as signal dependent charge-injection, nonlinear sampling capacitors, or nonconstant opamp gain, will at some point become dominant and become the limiting factor in the static resolution of the ADC.

This correction scheme has been demonstrated previously in [2]. There a subradix-2 pipelined ADC was designed and the gap was measured directly during a foreground calibration phase by driving the decision boundary voltage into each stage. This technique works well as demonstrated by the 15-bit ADC. The drawback is that foreground calibration requires taking the ADC out of service for calibration. Thus, it suffers from calibration drift and/or service interruptions.

DBGE uses this same correction scheme with the slight extension that if redundancy is used then the stage radix does not need reduced. Redundancy prevents the signal from going out of range and thus allows the code gap g_k to be negative. Without redundancy, the digital code gap gets clamped to be positive.

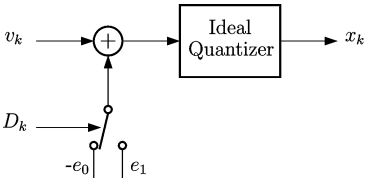

Fig. 15. Signal flow graph of nonlinear error model.

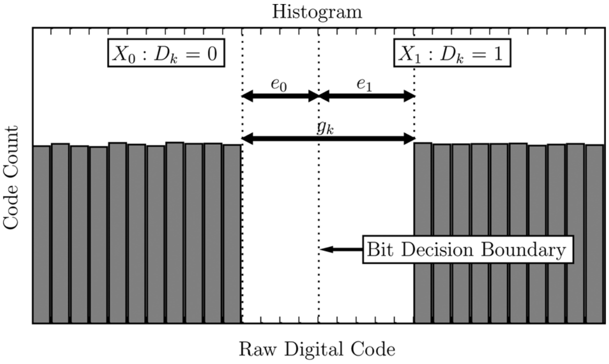

Fig. 16. Histogram of an example data set (in the absence of noise) corrupted by unknown offsets.

IV. Gap Estimation

DBGE differs from the work presented in [2] in the gap estimation method. DBGE is an indirect background calibration technique and relies on the statistics of the input signal to estimate the code gap of each stage. The static nonlinearities described previously cause the code gaps and can be modeled by the signal flow graph of Fig. 15. Here the analog input voltage v_k into stage k is corrupted with an unknown, nonrandom parameter e_1 or e_0 when the MSB decision D_k is 1 or 0, respectively. The resulting analog voltage is then quantized by the remaining stages of the ADC, and the output x_k is the raw output sample and the observation variable. This model initially neglects the effect of circuit noise which will be considered later.

Fig. 16 shows an example of a histogram collected when the first stage has code gaps of $e_0 = 4$ and $e_1 = 5$ and when the input voltage v_k is uniformly distributed in a region near the bit decision boundary. Observe that no codes appear in the histogram within the region of the code gap.

The goal of DBGE is to estimate the gap size g_k, where $g_k = e_1 + e_0$. Although the example of Fig. 16 uses parameters e_1 and e_0 that are integers, in reality they are not likely integers. Since DBGE corrects the digital output and not the source of the nonlinearity, there is little advantage to estimating or correcting the gap size to a finer precision than an integer. Initially the case when the error parameters are integers is considered and more realistic parameters are considered in the simulation results presented in Section V.

Following are several different gap estimation techniques of varying performance, hardware complexity, and robustness to circuit noise. For simplicity, they are all described for the case of a 1-bit/stage ADC where each stage has a single code gap.

These techniques, however, are general to higher resolution stages where each additional bit decision comparator produces an additional gap. For example, since a 1.5–bit/stage ADC requires 2–bit decision comparators, there will be 2-bit decision boundaries and, thus, two independent code gaps that need estimated and corrected separately.

A. Max-Min Gap Estimator

The Max-Min gap estimator utilizes a very simple algorithm for estimating the code gap. Receive a block of N samples. Split it into two sets X_1 and X_0 where X_1 is the set of all samples with an MSB $D_k = 1$ and X_0 is the set of all samples with $D_k = 0$. Estimate the gap $\hat{g}_{mm}$ as

$$\begin{aligned} \tilde{e}_1 &= \min\{X_1\} \\ \tilde{e}_0 &= \max\{X_0\} \\ \hat{g}_{mm} &= \tilde{e}_1 + \tilde{e}_0. \end{aligned} \tag{4}$$

In other words, the Max-Min estimator watches the data stream to find the maximum sample received below the decision boundary and minimum sample received above the decision boundary and subtracts the two to form the estimate $\hat{g}_{mm}$. Once corrected, the effect on the histogram will be to shift the bins on the right side of the code gap to the left to close the gap and remove the nonlinearity. Depending on the probability distribution of input voltage v_k, this estimate has varying degrees of performance. Whenever the probability distribution of v_k peaks or shares a peak at the decision boundary (which is midscale for a 1-bit/stage ADC), then this estimate is a maximum-likelihood (ML) estimate. Qualitatively, the more likely the input signal is to exercise the codes at the decision boundary, the better this estimation performs and vice versa. This is a desirable trend given that the impact of the nonlinearity is a function of the code density of the input near the nonlinearity. Furthermore, if the input signal has finite probability to be within one LSB of the decision boundary, then it can be shown that as the number of samples approaches infinity, the bias of this estimate approaches 0. How quickly it converges depends on the probability density in the region of the decision boundary.

The Max-Min estimator has a very efficient implementation in either hardware or software. A hardware implementation requires two registers for storing the minimum $\tilde{e}_1$ and maximum $\tilde{e}_0$ estimates and comparison logic to determine when to update these registers. Estimation proceeds as each sample is received. First, the bit decision D_k is checked. If it is 1, then the sample is compared to the minimum register and the minimum is updated if necessary. If D_k is 0, then the maximum register is compared and updated if necessary. To track changes in the gap that result from environmental changes, the minimum and maximum registers can be reset at a rate that matches the desired adaptation rate.

The Max-Min gap estimate provided in (4) suffers from a problem when one includes the effects of additive circuit noise in the analog processing path. Fig. 17 shows the addition of circuit noise to the signal flow graph as a random sample w_k. It has the effect of smearing the sharp edges of histogram at the code gap of the raw output samples. This can be seen in the example of Fig. 18 where Gaussian circuit noise with a standard deviation of $\sigma_w = 1.0$ LSBs is added to the signal.

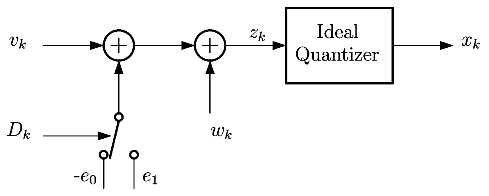

Fig. 17. Signal flow graph of error model including circuit noise w_k.

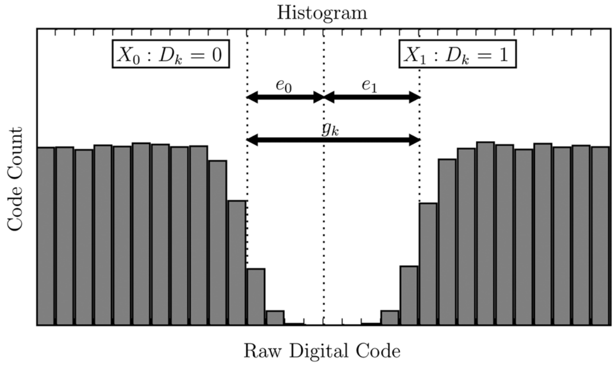

Fig. 18. Histogram of an example data set corrupted by a code gap and additive circuit noise.

With the additive noise smearing the sharp edges of the histogram, the Max-Min estimator will under compensate for the actual gap because the noise smears samples into the missing code region. The example histogram of Fig. 18 shows how samples at the edge of the histogram have spilled into the missing code region and that the minimum and maximum samples according to (4) no longer yield the correct estimate. Therefore, one must ensure that the circuit noise is adequately lower than the quantization noise to ensure the bias that results on the gap estimate when using the Max-Min estimator is sufficiently small. In ADCs where circuit noise is not sufficiently lower than quantization noise, the Max-Min estimator will not perform adequately.

B. Bin-Reshaping Gap Estimator

An additional compensation calculation can be employed to improve the performance of the Max-Min estimator. This technique is call the Bin-Reshaping gap estimator. Consider the case when there is no circuit noise and $e_0 = 3.5$. A sample histogram of such a case is shown in Fig. 19 for the case of a uniformly distributed input in the region of the bit decision boundary. The error parameter e_0 causes the input to only span half of the rightmost bin of set X_0. So that bin will only fill half as much as its neighbor and their ratio tells the fractional part of the error parameters e_0.

The basic concept behind Bin-Reshaping is to first quantize the input data to yield a coarse histogram where quantization noise is larger than the circuit noise. This meets the noise requirement of the Max-Min gap estimator, however, the Max-Min gap estimate will be of lower resolution and thus of limited effectiveness. However, one can extract the fractional

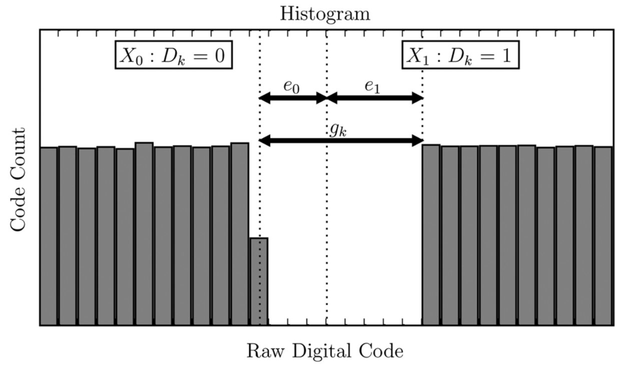

Fig. 19. Histogram of an example data set with fractional gap $e_0 = 3.5$ and no circuit noise.

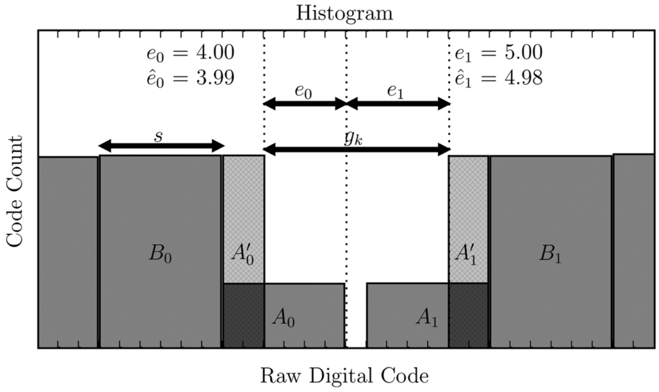

Fig. 20. Histogram showing geometric interpretation of the Bin-Reshaping estimation method.

part of this lower resolution estimate by taking the ratio of adjacent bins and interpolate back to the original resolution.

Geometrically this technique reshapes the inner most histogram bins as shown in the example in 20 where the high-resolution histogram of Fig. 18 is quantized by merging adjacent bins. This can be done by simply dropping the noisy bits prior to binning or by summing s adjacent bins of the high resolution histogram to produce a lower resolution histogram. Expressed mathematically, this is

$$h_s[m] = \sum_{i=m}^{m+s-1} h[i]$$

where $h_s[m]$ and $h[i]$ are the bin counts of the lower and higher resolution histogram, respectively. The bins labeled A_0, A_1, B_0, and B_1 in Fig. 20 make up the low resolution histogram.

The second step is to interpolate the value of the error parameters e_0 and e_1 across the two edge bins. Consider the case of estimating e_1. The bins labels A_1 and B_1 make up the two edge bins. Bin A_1' is created from bin A_1 by reshaping it to the same height as B_1 while preserving the area. The width of A_1' is taken as the effective minimum sample and thus the edge of the missing code gap. A similar procedure on bins A_0 and B_0 and can be used to find the effective maximum sample and thus the other edge of the missing code gap. The Bin-Reshaping gap estimate $\hat{g}_{br}$ is expressed mathematically as

$$\begin{aligned}
\hat{e}_1 &= \tilde{e}_1 + s\left(1 - \frac{h_s[\tilde{e}_1]}{h_s[\tilde{e}_1 + s]}\right)\\
\hat{e}_0 &= \tilde{e}_0 + s\left(1 - \frac{h_{-s}[\tilde{e}_0]}{h_{-s}[\tilde{e}_0 - s]}\right)\\
\hat{g}_{br} &= \hat{e}_1 + \hat{e}_0
\end{aligned} \tag{5}$$

where $\tilde{e}_1$ and $\tilde{e}_0$ are the Max-Min estimates from the same data set.

If s, the number of histogram bins to merge, is not picked large enough to adequately cover the spread in the histogram caused by the circuit noise, then the estimate will continue to under compensate. Thus, s should be selected large enough to span the circuit noise to within good engineering tolerances (e.g., $s > 3\sigma_w$). However, since the Bin-Reshaping gap estimator makes the approximation that the input is uniformly distributed over a width of $2s$ codes, s should be chosen as small as possible. In practice s should be selected after characterizing the amount of circuit noise. In the example of Fig. 20, an extremely conservative choice of $s = 6\sigma_w$ is used.

The Bin-Reshaping gap estimator makes the approximation that the input voltage is uniformly distributed across the two innermost bins on each side of the code gap region. This approximation is reasonable for many applications, especially high resolution ADCs, and is similar in nature to the approximation used when modelling quantization noise as uniformly distributed.

The Bin-Reshaping gap estimator is still very computationally friendly. Each estimate $\hat{e}_0$ and $\hat{e}_1$ requires an additional two registers for accumulating two lower resolution histogram bins. A division of these two registers must be performed, but since the estimate will be running at a very slow rate compared to that of the ADC, it can implemented serially using shifts and subtractions for minimal gate count.

C. *Cost-Minimizing Estimator*

The traditional manner in which ADC linearity is characterized using code density measurements [26], [27] provides the inspiration for another more flexible gap estimator. Code density methods calculate the differential nonlinearity (DNL) and integral nonlinearity (INL) of the ADC by comparing the histogram or code density of the measured response to the theoretical response. When the ADC is stimulated with a uniformly distributed input, then a perfectly linear ADC will produce a histogram with uniform bin counts or code densities. Any nonlinearities in the ADC will produce nonuniform bin counts as seen in the example histograms of Fig. 18. From the bin counts, the DNL is derived from the ratio of adjacent bins and the INL is the cumulative sum of the DNL.

The Cost-Minimizing gap estimator takes an iterative approach to estimating an optimal code gap based on a predetermined cost function run on the histogram response of the ADC in the window of the bit decision boundary. The algorithm is as follows:

1) Receive a block of data from ADC.

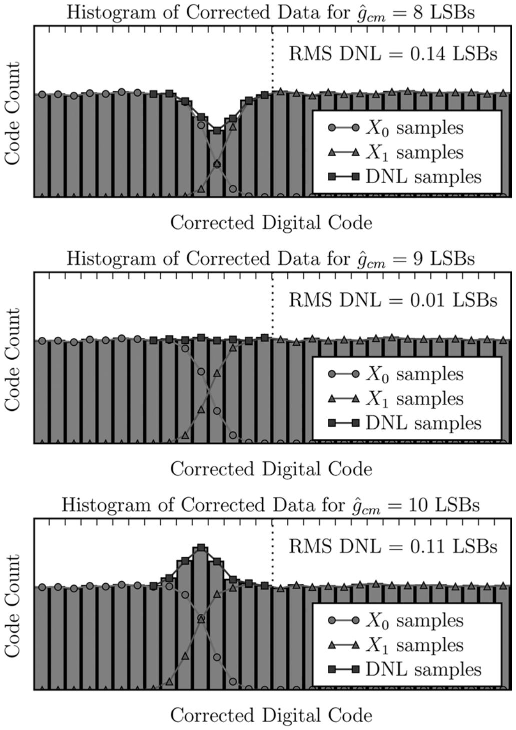

Fig. 21. Histograms under various $\hat{g}_{cm}$ estimates. Actual $g = 9$ LSBs.

2) Divide data into two sets. X_0 is the set where $D_k = 0$ and X_1 is the set where $D_k = 1$.
3) Calculate the histogram of each set.
4) Select an initial gap estimate.
5) Shift the X_1 histogram to the left by the gap estimate amount and add it to the X_0 histogram. This combined histogram is equivalent to the histogram that would result if one corrected the samples with the selected gap estimate.
6) Evaluate the cost function on the combined histogram.
7) Increment the gap estimate and return to step 5. After sweeping the gap estimate over the desired range, select the gap estimate $\hat{g}_{cm}$ that minimizes the cost function and stop.

The plots of Fig. 21 show the histogram manipulations of this procedure for 3 different gap estimates. This example corresponds to the original data set displayed previously in Fig. 18 where circuit noise was introduced into the simulation. The actual gap used in this example is 9 LSBs. In the first plot, a gap estimate of $\hat{g}_{cm} = 8$ LSBs is selected. The histogram of the X_0 is shown as the line marked with circles. The histogram from set X_1 is shown as the line marked with triangles. This histogram get shifted to the left by 8 LSBs and added to the X_0 histogram to produce the gray shaded histogram. For this example, the cost function is selected as the root mean square (RMS) of the DNL over an 8σ circuit noise window where the two sets overlap at the bit decision boundary. The samples used in the DNL calculation of this example are marked with squares. Observe the dip in the histogram for this gap estimate. In the next plot, the gap estimate is updated to $\hat{g}_{cm} = 9$ LSBs. The resulting histogram is flat, which is indicative of a histogram from a linear ADC. In the last plot, the gap estimate is updated to $\hat{g}_{cm} = 10$ LSBs. Observe the mound that results in the histogram. Qualitatively these plots show that a gap estimate of $\hat{g}_{cm} = 9$ LSBs produces the most linear ADC. The RMS DNL is a quantitative metric for determining this. In Fig. 22 the RMS DNL is plotted for this example as a function of the gap estimate. As expected, it is minimized at $\hat{g}_{cm} = 9$ LSBs, which corresponds to the actual gap error used in the simulation. Thus, for this example, the gap estimate of $\hat{g}_{cm} = 9$ would be selected as it minimizes the cost function.

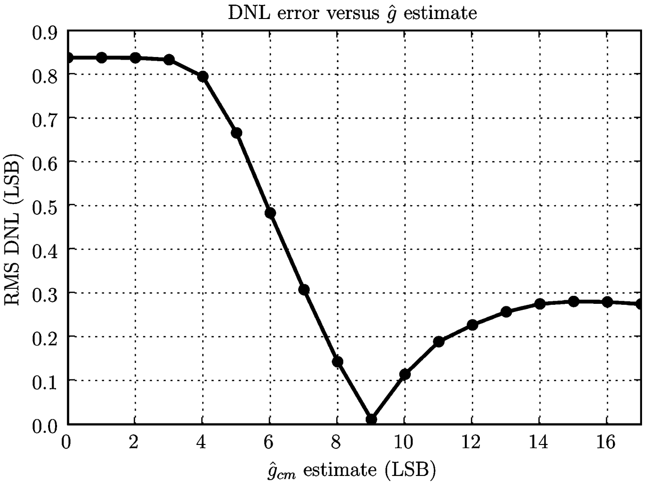

Fig. 22. Plot measured DNL versus Cost-Minimizing gap estimate $\hat{g}_{cm}$.

The size of the window over which the RMS DNL should be calculated is governed by similar constraints to that of the Bin-Reshaping estimator. It should be wide enough to span the spread in the histogram caused by the circuit noise but it should be as narrow as possible to ensure that the input is approximated as well as possible by a uniform distribution. For the example shown in Figs. 21 and 22 a spread of 8 bins is used, which is 8 standard deviations of the circuit noise. This example, therefore, assumes the input can be approximated as uniformly distributed over 8 LSBs.

Even if the input is not well approximated as uniform over the spread of the circuit noise, however, the Cost-Minimizing estimator offers the flexibility of selecting a cost function that is more appropriate for the given input signal. For example, another technique is to run a linear regression of the combined histogram over the desired window and select the gap estimate that produces the lowest RMS error or has the highest coefficient of determination R^2. This first order regression would then allow for inputs with distributions of constant gradients over the spread of the circuit noise. Another variation of this idea that is less complex would be a cost function that calculates the RMS value of the difference between adjacent bins.

The tradeoff for the increased flexibility of the Cost-Minimizing estimator is an increase in complexity and hardware. It requires an increased register count to store histogram bins and also additional logic to perform the iterative search for the gap estimate that minimizes the selected cost function. Despite this, however, this estimator is still relatively simple and would not require a large digital footprint compared to the overall size of the ADC.

D. Estimator Discussion

Because DBGE is an indirect background calibration technique, it does not require service interruptions or suffer from calibration drift as foreground technique do. However, since it is dependant on the statistics of the input signal, it may not be appropriate for applications with input statistics that do not exercise codes in the vicinity of the decision boundaries of the ADC. Such applications, however, can use a combination of foreground and background techniques where at startup the initial gap estimates are measured during a direct foreground calibration phase using a technique like that described in [2]. Then after initialization, DBGE can then be used in the background to track parameter changes to eliminate calibration drift and avoid service interruptions or redundant hardware.

The previous discussions focused primarily on a single stage of a 1-bit/stage ADC. When going to higher resolution stages, unless the code gaps are systematic, each bit decision comparator of the sub-ADC will require independent hardware to estimate each code gap. Furthermore, each stage will require independent gap estimation. For example, suppose the first four stages of a 1.5-bit/stage ADC require calibration. Then 8 code gap estimates will be required for the 2-bit decision comparators in each of the four stages. Since the estimator updates at slower rate than the sampling frequency of the ADC, it is possible to share hardware between the various stages and perform updates in a serial fashion rather than running parallel estimates.

It is also possible to run this algorithm on a processor in a block-based fashion. In this approach, a block of raw data is collected. Then the processor sweeps through the data producing a gap estimate for each stage and correcting each stage in succession.

V. Simulation Results

DBGE has been simulated under many different conditions. Shown here are the results of a 13–stage 1.5-bit/stage pipelined ADC simulated with the mismatch parameters specified in Table I. Circuit noise was included in each stage to limit the effective resolution to 12.5 bits. The DNL, INL, and DFT plots of uncalibrated ADC are shown in Figs. 23–25. These show that the static nonlinearities due to the mismatch parameters of Table I lower the effective resolution to 9.2 bits.

DBGE was performed on the first six stages. Two hundred thousand samples from a zero mean Gaussian input were sent into the ADC. The results of the Cost-Minimizing estimator are shown the INL and DFT responses in Figs. 24 and 25. The effective resolution has been raised to 12.5 bits. This means the resolution is limited by the additive circuit noise and is no longer limited by static nonlinearities. The spurious-free dynamic range (SFDR) goes from 67.7 to 91.0 dB after calibration, and the INL goes from ±23 LSB_{14} to ±1.5 LSB_{14}. The INL improvements are limited to a single LSB because both the estimation and correction algorithms use the digital data obtained from the ADC which is limited to this resolution. The effective number of bits (ENOB), signal-to-noise and distortion ratio (SNDR), SFDR, and INL were calculated according to the procedures in [26]. Table II summarizes the results for both the raw and corrected ADC samples and shows the performance of the various estimators to this setup. Observe that the Max-Min estimator does not perform as well as the others, and this is due to the additive circuit noise introducing a bias. The Bin-Reshaping and Cost-Minimizing estimators, however, perform similarly.

TABLE I
Simulation Mismatch Parameters

Stage	Capacitor Mismatch	Opamp Gain	Comparator Offset	Voltage Offset
13	0.19%	542	0.24%	-0.41%
12	0.04%	606	-0.06%	-0.30%
11	-0.01%	597	4.72%	0.16%
10	-0.15%	454	-2.07%	0.39%
09	0.07%	421	2.71%	-0.15%
08	-0.18%	762	0.26%	-0.43%
07	0.21%	460	2.69%	-0.48%
06	-0.09%	651	-0.99%	-0.04%
05	0.51%	243	3.91%	-0.43%
04	-0.54%	299	-2.16%	-0.26%
03	0.55%	998	-1.47%	0.47%
02	-0.05%	705	3.07%	0.40%
01	-0.12%	535	4.19%	0.35%

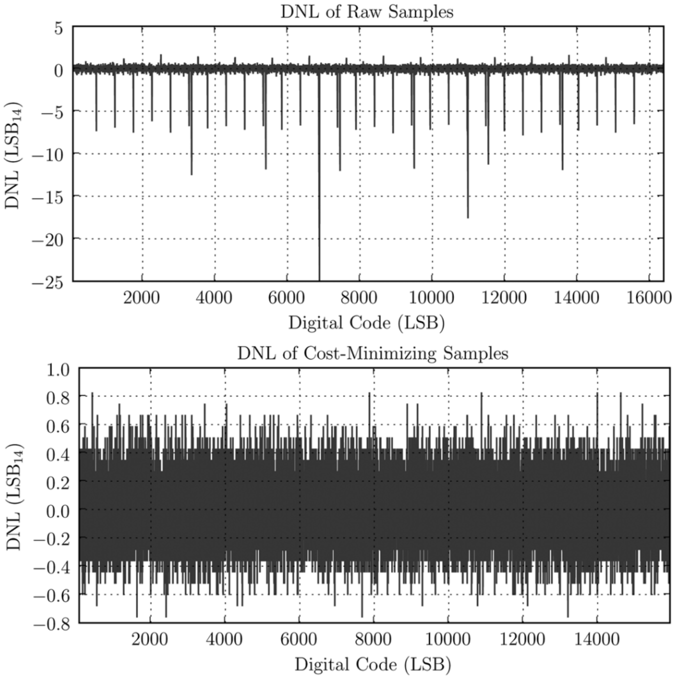

Fig. 23. Raw and calibrated DNL of 13–stage 1.5-bit/stage ADC with mismatch parameters specified in Table I.

Similar results are obtained with a wide range of inputs including sine wave, ramp, and uniformly random. The performance and speed of convergence of DBGE are input signal dependent. For a given estimation performance, the speed of convergence will scale with the probability of the input in the vicinity of a particular code gap. This means that decision boundaries corresponding to inputs with a low probability will take longer to collect enough samples to converge than those with a higher probability.

An input with zero probability at a particular code boundary is problematic if it has finite probability on both sides of the boundary. In this case, the input has a missing code gap, and

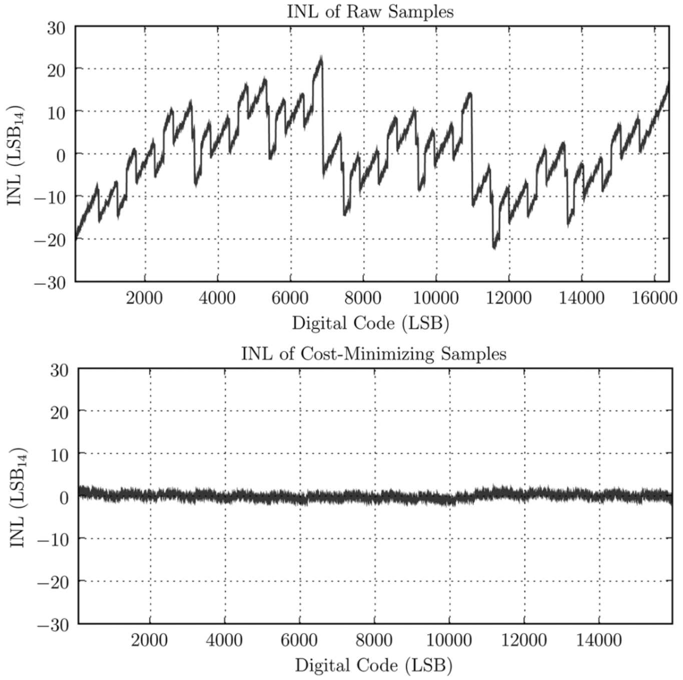

Fig. 24. Raw and calibrated INL of 13-stage 1.5-bit/stage ADC with mismatch parameters specified in Table I.

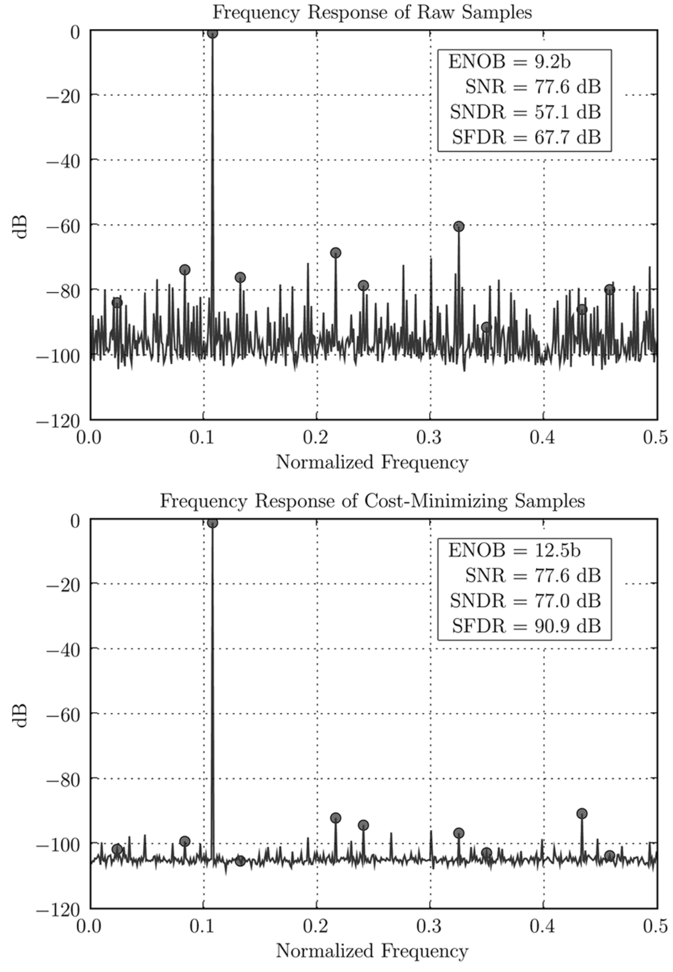

Fig. 25. Raw and calibrated DFT response of 13–stage 1.5-bit/stage ADC with mismatch parameters specified in Table I.

TABLE II
SIMULATION RESULTS

	ENOB (bits)	SNDR (db)	SFDR (db)	INL (LSB$_{14}$)
Raw	9.2	57.1	67.7	±23
Max-Min Estimator	11.8	72.7	85.6	±4
Bin-Reshaping Est.	12.6	77.5	91.1	±1.5
Cost-Minimizing Est.	12.5	77.0	91.0	±1.5

DBGE will close the gap as it is unable to discern whether gaps come from the input signal or from the ADC. Clearly, applications with such inputs characteristics are not good candidates for DBGE. There is no problem if the input has zero probability at a particular decision boundary and has finite probability on only one side of the boundary. This corresponds to the case that a particular input does not fill the full input range of the ADC. Any decision boundaries outside of the range of the input signal will have wrong estimates, but since the input does not exercise those codes, their wrong estimates do not matter.

VI. CONCLUSION

The motivation for DBGE came from the observation that the nonlinearities that dominate CMOS switch-capacitor circuit design cause code gaps at each bit decision boundary of the sub-ADC. This technique, however, is general to a broader class of both implementations and architectures. It applies to any situation where the amplified error or residue from each stage causes a decision boundary gap.

An appropriate follow-up question to the work presented herein is what estimator and cost function achieves optimal performance. The answer to this question and others such as convergence time is beyond the scope of this paper. One reason is that this requires specifying the statistics of the input signal and an additional cost function over which to define optimality. Instead, this work presents a general framework for performing indirect background calibration of the common static nonlinearities in pipelined ADCs. The estimator and cost function should be selected and analyzed based on the specific application and the statistics of the input signal and remains as an open research question.

In its general form, DBGE is an adaptive, digital, indirect method of background calibration. The advantages of DBGE are numerous. There is no need for additional analog hardware, such as a redundant channels/stages or a reference converter to calibrate against. The calibration is highly accurate because the transition points are directly aligned. Furthermore, its simplicity makes it amenable to VLSI and/or processor-based implementations. Thus, DBGE is a calibration approach that can be implemented to improve existing ADC designs or to shape new designs by relaxing analog circuit requirements for high gain opamps, matched capacitors, and low offset comparators. Reducing these design constraints allows the designer to reduce power and/or increase conversion speed, and perhaps most importantly, it can be an enabling factor for ADC design in deep submicron technologies.

REFERENCES

[1] Y. Chiu, C. W. Tsang, B. Nikolic, and P. R. Gray, "Least mean square adaptive digital background calibration of pipelined analog-to-digital converter," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 1, pp. 38–46, Jan. 2004.

[2] A. Karanicolas, H.-S. Lee, and K. Bacrania, "A 15-b 1-msample/s digitally self-calibrated pipeline ADC," *IEEE J. Solid-State Circuits*, vol. 28, no. 6, pp. 1207–1215, Dec. 1993.

[3] U.-K. Moon and B.-S. Song, "Background digital calibration techniques for pipelined ADC's," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, pp. 102–109, Feb. 1997.

[4] P. Li, M. Chin, P. Gray, and R. Castello, "A ratio-independent algorithmic analog-to-digital conversion technique," *IEEE J. Solid-State Circuits*, vol. SSC-29, no. 6, pp. 828–836, Dec. 1984.

[5] B.-S. Song, M. Tompsett, and K. Lakshmikumar, "A 12-bit 1-msample/s capacitor error-averaging pipelined A/D converter," *IEEE J. Solid-State Circuits*, vol. 33, no. 6, pp. 1316–1323, Dec. 1988.

[6] T.-H. Shu, K. Bacrania, and C.-I. Chi, "Statistical correction of A/D converter errors," *IEEE BCTM*, pp. 189–191, Sep. 1996.

[7] A. Dent and C. Cowan, "Linearization of analog-to-digital converters," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 729–737, Jun. 1990.

[8] U. Gatti, G. Gazzoli, F. Maloberti, and S. Mazzoleni, "A calibration technique for high-speed high-resolution a/d converters," *Adv. A/D and D/A Convers. Tech. Appl.*, pp. 168–171, Jul. 1999.

[9] X. Dai, D. Chen, and R. Geiger, "A cost-effective histogram test-based algorithm for digital calibration of high-precision pipelined ADCs," in *Proc. IEEE ISCAC*, May 2005, vol. 5, pp. 4831–4834.

[10] L. Jin, D. Chen, and R. Geiger, "A digital self-calibration algorithm for ADCs based on histogram test using low-linearity input signals," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, pp. 1378–1381.

[11] E. Blecker, T. McDonald, O. Erdogan, P. Hurst, and S. Lewis, "Digital background calibration of an algorithmic analog-to-digital converter using a simplified queue," *IEEE J. Solid-State Circuits*, vol. 38, no. 6, pp. 1489–1497, Jun. 2003.

[12] E. Siragusa and I. Galton, "Gain error correction techinque for pipelined analog-to-digital converters," *Electron. Lett.*, vol. 36, pp. 617–618, 2000.

[13] J. Li and U.-K. Moon, "Background calibration techniques for multistage pipelined adcs with digital redundancy," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 50, no. 9, pp. 531–538, Sep. 2003.

[14] E. Siragusa and I. Galton, "A digitally enhanced 1.8-v 15-bit 40-msamples/s CMOS pipelined ADC," *IEEE J. Solid-State Circuits*, vol. 39, no. 12, pp. 2126–2138, Dec. 2004.

[15] B. Murmann and B. E. Boser, "A 12-bit 75-ms/s pipelined ADC using open-loop residue amplification," *IEEE J. Solid-State Circuits*, vol. 38, no. 12, pp. 2040–2050, Dec. 2003.

[16] B. Hernes, J. Bjornsen, T. N. Anderson, A. Vinje, H. Korsvoll, F. Telsto, A. Briskemyr, C. Holdo, and O. Moldsvor, "A 92.5mW 205 ms/s 10b pipeline if ADC implemented in 1.2v/3.3v 0.13 μm CMOS," in *Proc. ISSCC Dig. Tech. Papers*, 2007, pp. 462–463.

[17] S. Sonkusale, J. v. d. Spiegel, and K. Nagaraj, "True background calibration technique for pipelined ADC," *Electron. Lett.*, vol. 26, pp. 786–788, Apr. 2000.

[18] J. Elbornsson and J.-E. Eklund, "Histogram based correction of matching errors in subranged ADC," in *Proc. ESSCIRC 2001*, Sep. 2001, pp. 555–558.

[19] U. Eduri and F. Maloberti, "Online calibration of a nyquist-rate analog-to-digital converter using output code-density histograms," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 1, pp. 15–24, Jan. 2004.

[20] L. Brooks and H.-S. Lee, "A zero-crossing based 8 bit, 200ms/s pipelined ADC," *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2677–2687, Dec. 2007.

[21] E. Balestrieri, P. Daponte, and S. Rapuano, "A state of the art on ADC error compensation methods," *IEEE Trans. Instrum. Meas.*, pp. 1388–1394, Aug. 2005.

[22] P. Yu and H.-S. Lee, "A 2.5v, 12-b 5-MSample/s pipelined cmos adc," *IEEE J. Solid-State Circuits*, vol. 31, pp. 1854–1861, Dec. 1996.

[23] S. Ray and B.-S. Song, "A 13-b linear, 40-MS/s pipelined ADC with self-configured capacitor matching," *IEEE J. Solid-State Circuits*, vol. 42, no. 3, pp. 463–474, Mar. 2007.

[24] S. Lewis, S. Fetterman, G. Gross, R. Ramachandran, and T. Viswanathan, "A 10-b 20-msample/s analog-to-digital converter," *IEEE J. Solid-State Circuits*, no. 2, pp. 351–358, Mar. 1992.

[25] V. Divi and G. Wornell, "Signal recovery in time-interleaved analog-to-digital converters," in *Proc. IEEE ICASSP*, 2004, pp. 593–596.

[26] *IEEE Standard for Terminology and Test Methods for Analog-To-Digital Converters*, IEEE STD 1241–2000, Dec. 2000.

[27] J. Doernberg, H.-S. Lee, and D. Hodges, "Full-speed testing of A/D converters," *IEEE J. Solid-State Circuits*, vol. SSC-19, no. 6, pp. 820–827, Dec. 1984.

Lane Brooks (S'04) received the B.S. and M.Eng. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 2001.

From 2001 to 2005, he was with SMaL Camera Technology designing mixed signal circuits for imaging applications. In 2005, he returned to MIT where he is currently pursuing the Ph.D. degree.

Hae-Seung Lee (M'85–SM'92–F'96) received the B.S. and M.S. degrees in electronic engineering from Seoul National University, Seoul, Korea, in 1978 and 1980, respectively. He received the Ph.D. degree in electrical engineering from the University of California, Berkeley, in 1984, where he developed self-calibration techniques for A/D converters.

In 1980, he was a Member of Technical Staff with the Department of Mechanical Engineering, Korean Institute of Science and Technology, Seoul, where he was involved in the development of alternative energy sources. Since 1984, he has been with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, where he is now a Professor and the Director of Center for Integrated Circuits and Systems. Since 1985, he has acted as Consultant to Analog Devices, Inc., Wilmington, MA. His research interests are in the areas of analog integrated circuits with the emphasis on analog-to-digital converters in scaled CMOS technologies. He authored or coauthored more than 100 journal and conference papers.

Prof. Lee was a recipient of the 1988 Presidential Young Investigators' Award, and a corecipient of 2002 ISSCC Outstanding Student Paper Award. He has served on a number of technical program committees for various IEEE conferences, including the International Electron Devices Meeting, the International Solid-State Circuits Conference, the Custom Integrated Circuits Conference, and the IEEE Symposium on VLSI circuits. From 1992 to 1994, he was an Associate Editor for the IEEE JOURNAL OF SOLID-STATE CIRCUITS.