

Linearity Test of Analog-to-Digital Converters Using Kalman Filtering

Le Jin, Degang Chen and Randall Geiger

Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011

Abstract

This work introduces an efficient code-density linearity testing algorithm for ADCs that can achieve high accuracy within short test time. The proposed algorithm uses Kalman filtering to suppress the effect of errors in the histogram counts based on characteristics of input noise and circuit mismatches. Appropriate versions of the algorithm for ADCs of flash and pipelined architectures are introduced respectively. Simulation results show that this approach can reduce the INL_k estimation error by over 50% and achieve desired accuracy with a much smaller number of samples as compared to the conventional algorithm. Simulation and experimental results show that the proposed algorithm can significantly shorten the linearity test time by a factor of 10 or higher. Therefore, it can enable test and help maintain the quality of high-performance ADCs, and reduce the production test time and cost for medium and low resolution ADCs.

I. Introduction

Integral nonlinearity (INL) and differential nonlinearity (DNL) are two critical specifications on static performance of analog-to-digital converters (ADCs). Accurate INL and DNL test results can indicate the quality of an ADC and provide information of nonideality in design and manufacture, so they are always tested for ADC products on the market. The test cost of INL and DNL is a big concern of ADC manufacturers for the following reasons. First is that the test cost itself is high. Nowadays the analog test cost of a mixed-signal device can take a significant portion of its total production cost and this ratio is expected to become even higher in the future [11]. Cut off the test cost can significantly reduce the total production cost. The second reason is that the ADC has a high volume. It is one of the world's largest volume analog and mixed-signal (AMS) integrated circuit products [1]. Test cost reduction of INL and DNL has essential meaning to manufacturers as the multiplication effect of the large volume is considered. Furthermore, for current and upcoming high-resolution ADCs, time for full-code INL and DNL test, which is directly related to the cost, is prohibitively long because of the large number of variables to be accurately measured. Reduced-code test methods are often adopted for high-resolution parts, but they do not provide complete characterization of the ADC under test. There is lack of a full-code linearity test strategy with a sufficiently low cost for production test of high-resolution ADCs. Because of these reasons, a testing algorithm that can achieve a desired level of accuracy in a short period of time is very useful and important, especially for ADCs of

16-bit and up resolutions that have a large number of transition levels to be tested with very high accuracy on a test facility.

The code-density test method [2, 3] is widely adopted for testing ADCs' static linearity in the industry, because its implementation is straightforward and its computational complexity is low. The conventional code-density method requires input signals with better linearity than the resolution of the ADC under test and a significant amount of samples to average out the noise effect for accurate measurement results. The requirement on the input signal's linearity has been studied and new methods using nonlinear signals are proposed [6], but not many approaches are proposed for improving efficiency of the code-density test method as authors are aware of. A maximum likelihood estimator (MLE) method related to the code-density test was studied in [5], but its computational complexity is high and not suitable for high-resolution ADCs.

This work introduces a modified code-density algorithm that uses the Kalman filter to reduce the effect of errors on the histogram data. Simulation and experimental results show that the proposed algorithm can achieve the same level of accuracy as that of the conventional code-density algorithm but using a significantly smaller number of samples, which means shorter test time and lower test cost. The additional computations introduced by the modified algorithm is very few and negligible with current days' computing power. The new method can be used to enable testing of high-resolution ADCs with better coverage and reduce the time and cost of testing medium-resolution ADCs.

The rest part of the paper is organized as follows. Section II will briefly review the code-density test method and discuss its estimation errors in details. Section III describes the modified algorithms using Kalman filtering to improve the test accuracy for flash and pipeline ADCs. Simulation results are presented in Section IV, and experimental results are given in Section V.

II. Code-Density Test for ADC Linearity

This section provides some background information on ADC linearity specifications and the code-density test method with discussions on its performance.

A. ADC Linearity Specifications

Terminologies used in this paper will follow [4] with modifications in indexing for simplicity of formulation. An ideally linear ADC is expected to have its transition levels

evenly spaced over the input range. The differential nonlinearity for code k of an n -bit ADC is the relative error in the k -th code bin width and can be written as

$$DNL_k = (N-2) \frac{T_k - T_{k-1}}{T_{N-2} - T_0} - 1(LSB), \quad (1)$$

where $N=2^n$; T_k , $k=0, 1, \dots, N-2$, is the k -th transition level of the ADC between code k and $k+1$; and the unit of DNL_k is called a least significant bit (LSB) and equal to the averaged code bin width

$$Q = \frac{T_{N-2} - T_0}{N-2} = 1(LSB). \quad (2)$$

The integral nonlinearity for code k is

$$INL_k = (N-2) \frac{T_k - T_0}{T_{N-2} - T_0} - k(LSB). \quad (3)$$

Equations (1) and (3) show that the offset and gain of transition levels do not affect values of DNL_k and INL_k . INL and DNL of the ADC are defined as

$$DNL = \max\{|DNL_k|, k=1,2,\dots,N-2\} \quad (4)$$

$$\text{and } INL = \max\{|INL_k|, k=0,1,\dots,N-2\}, \quad (5)$$

respectively. They indicate the maximum deviations of an ADC from its ideally linear counterpart. From (1) and (3) we can also see that

$$INL_k = INL_{k-1} + DNL_k. \quad (6)$$

To test linearity performance of an ADC, its transition levels are usually measured in a relative sense and used to calculate INL and DNL based on equations above. Other static parameters such as the offset voltage and gain error are measured separately.

B. Code-Density Test of ADC Linearity

The code-density method uses a linear ramp or sinusoidal signal as a stimulus to the ADC under test. We assume our test signal is an ideal ramp in following discussions, but the developed algorithms, analyses and conclusions can be applied to the sine wave test as well.

The ADC takes samples y_i on the stimulus signal at a constant rate and generates output codes $c_i = k$ if $T_{k-1} < y_i \leq T_k$. The number of occurrences of code k is recorded as a histogram count H_k . This count can be used as an estimate of the k -th code bin width, since a larger bin will naturally contain more samples and the two quantities are proportional to each other if the samples are evenly spaced on the voltage axis, as in the linear ramp test situation. The code-density method estimates DNL_k and INL_k as

$$D\hat{N}L_k = \frac{H_k}{\bar{H}} - 1 \quad (7)$$

$$\text{and } I\hat{N}L_k = \sum_{j=1}^k H_j / \bar{H} - k, \quad (8)$$

respectively, where

$$\bar{H} = \sum_{j=1}^{N-2} H_j / (N-2) \quad (9)$$

is the averaged histogram count.

By comparing to definitions in (1) and (3), (7) and (8) are equivalent to estimating transition levels of the ADC under test as

$$\hat{T}_k = \sum_{j=0}^k H_j / N_s = \sum_{i=1}^{N_s} v_{ki} / N_s, \quad (10)$$

where N_s is the total number of samples, and v_{ki} is a characteristic function such that

$$v_{ki} = \begin{cases} 1, & c_i \leq k; \\ 0, & k < c_i. \end{cases} \quad (11)$$

The estimated T_k is proportional to the number of samples that give out an output code smaller than or equal to k .

C. Estimation Error in Code-Density Method

Because of the noise in the signal generator and ADC circuitries, the sampled voltage is deviated from the stimulus signal as

$$y_i = x_i + n_i, \quad (12)$$

where x_i is the ideal ramp signal level at the i -th sampling instance and n_i is an additive noise. The output code c_i and consequently v_{ki} become random variables as well. It is usually assumed that n_i is white and takes a normal distribution $N(0, \sigma^2)$, so the probability for the i -th output code being smaller than or equal to k , notated as p_{ki} , can be calculated as

$$\begin{aligned} p_{ki} &= P(v_{ki} = 1) = P(c_i \leq k) \\ &= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{T_k} \exp\left(-\frac{(t-x_i)^2}{2\sigma^2}\right) dt = \Phi\left(\frac{T_k - x_i}{\sigma}\right), \end{aligned} \quad (13)$$

where $\Phi(x)$ is the cumulative distribution function (CDF) of the standard normal distribution $N(0, 1)$. Further discussions on the above modeling and calculation can be found in [5] and its references.

As noting that v_{ki} has a Bernoulli distribution [6] and independent from each other with respect to i by our assumption, the mean and variance of the estimator in (10) can be written out as [7]

$$E\{\hat{T}_k\} = \sum_{i=1}^{N_s} p_{ki} / N_s \quad (14)$$

$$\text{and } \text{Var}\{\hat{T}_k\} = \sum_{i=1}^{N_s} p_{ki} q_{ki} / N_s^2, \quad (15)$$

where $q_{ki} = 1 - p_{ki}$.

Using (3) and (8), we can further show that the INL_k estimation error is equal to the transition level estimation error expressed in LSB as

$$e_k = I\hat{N}L_k - INL_k = (\hat{T}_k - T_k) / Q(LSB), \quad (16)$$

if the two terminal transition levels T_{N-2} and T_0 are specifically measured with high accuracy, which is usually done for offset and gain error measurements. We don't have a closed-form expression of (14), but it can be numerically verified that this mean is very close to T_k . Therefore the estimated INL_k is unbiased with $E\{e_k\}=0$. The estimated value has a variance

$$\begin{aligned} \text{Var}\{e_k\} &= \text{Var}\{\hat{T}_k\}/Q^2 \\ &\cong A \frac{\sigma}{D_s} (\text{LSB}^2), \end{aligned} \quad (17)$$

where A is a constant, σ is expressed in LSB, and D_s is called the sampling density, defined as the average number of hits of an ADC output code in the test and equal to N_s/N . The second equality comes from (15) and is a good approximation for k not close to 0 or $N-1$. The value of A can be numerically calculated as about 0.6.

Equation (17) gives an analytical relationship between testing accuracy of the code-density method and two critical parameters. It can be intuitively interpreted that the test will be accurate if the noise is small and the number of samples is large. As we can see that (7), (8) and (10) are very straightforward estimations based on the histogram counts, the performance of the code-density method is not optimal and can be improved by some sophisticated approaches.

III. ADC Linearity Test with Kalman Filtering

The Kalman filter estimates values of the state variables of a dynamic system from noisy measurements of the variables. It can give optimal linear estimation, in the sense of minimum Mean Square Error (MSE), for signals under Gaussian noise. It is one of the most powerful tools in adaptive control and has extremely wide applications in various disciplines. The ADC linearity test problem can be put in such a way that we can straightforwardly apply the existing theory of Kalman filtering and make significant improvement to testing accuracy. The flowchart of the proposed method is shown in Fig. 1. H_k , \hat{INL}_k and \tilde{INL}_k are the histogram count, estimated INL_k in (8), and the optimal estimation of INL_k using the Kalman filter, respectively.

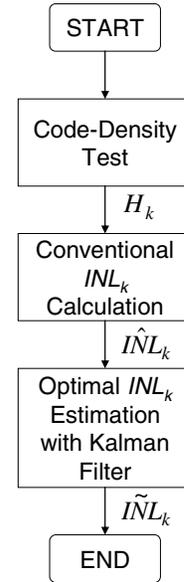


Fig. 1. Flowchart of ADC testing with Kalman filtering.

A. Problem Formulation

Writing (6) and (16) together, we get INL_k of the ADC under test in the following form

$$INL_k = INL_{k-1} + DNL_k; \quad (18a)$$

$$\hat{INL}_k = INL_k + e_k. \quad (18b)$$

INL_k can be viewed as a state variable of the dynamic system described by (18a), and we have a noisy measurement of it using the code-density method, \hat{INL}_k in (18b). Kalman filtering can be applied on (18) to find the optimal estimation of INL_k , $k=0, 1, \dots, N-2$, based on the code-density test result under the following two conditions that

1) DNL_k is a Gaussian random variable of zero mean satisfying

$$E\{DNL_k DNL_l\} = C \cdot \delta_{kl}, \quad (19)$$

where δ_{kl} equals 1 only if $k=l$ and equals to 0 otherwise; and

2) e_k is Gaussian with zero mean and satisfies

$$E\{e_k e_l\} = R \cdot \delta_{kl}. \quad (20)$$

The condition on e_k is generally true based on Section II. The assumption on DNL_k could be violated, and we will discuss appropriate modifications to it later. We further assume DNL_k and e_k are independent which is reasonable under realistic situations.

B. INL_k Test for Flash ADCs with Kalman Filtering

As a complete understanding of the Kalman filter requires much background information outside the scope of this paper, we will not provide derivation and proof details of the general Kalman filter algorithm. Interested readers are referred to [12] for rigorous descriptions and more references. We will provide detailed recursive estimation

steps so that the algorithm can be practically implemented after reading this paper. Some necessary discussions and comments are given as well.

The first condition on DNL_k is not always true for different types of ADCs, but it is a good description of ADCs with a flash architecture [9]. In a flash ADC, mismatch errors of the resistors and offset errors of comparators contribute to DNL_k . These errors are usually due to random process variations and managed to be uncorrelated using some well-known design techniques, such as common-centroid layout. Therefore DNL_k 's are independent of each other in a flash ADC, and the variance C is the summation of the mismatch error variance and offset error variance.

For flash ADCs, the Kalman filter algorithm can be described by the following equations [8] with initial conditions $INL_0 = 0$ and $P[0] = 0$, which are generally true by (3) and (8),

$$R_e[k] = P[k] + R; \quad (21a)$$

$$K_p[k] = P[k]R_e^{-1}[k]; \quad (21b)$$

$$P[k+1] = P[k] + C - K_p[k]R_e[k]K_p[k]; \quad (21c)$$

$$\tilde{INL}_k = \tilde{INL}_{k|k-1} + K_p[k](\hat{INL}_k - \tilde{INL}_{k|k-1}); \quad (21d)$$

$$\tilde{INL}_{k+1|k} = \tilde{INL}_k. \quad (21e)$$

\tilde{INL}_k in (21d) is the optimal estimation of INL_k given all the histogram information. It is calculated from $\tilde{INL}_{k|k-1}$, the predicted value of INL_k based on the histogram information for codes less than or equal to $k-1$, with correction introduced by the measured \hat{INL}_k . The difference between the measured and predicted values, $\hat{INL}_k - \tilde{INL}_{k|k-1}$, is called the measurement innovation. It indicates how good the prediction is but is also affected by the measurement error e_k . $K_p[k]$ in (21d) is the Kalman gain. It determines how much of the innovation should be used as correction to the predicted value. It can be shown that $P[k]$ is the variance of the final prediction error of $\tilde{INL}_{k|k-1}$.

\tilde{INL}_k and $\tilde{INL}_{k|k-1}$ are recursively calculated using (21d) and (21e) for all k values from 0 to $N-2$. The Kalman gain $K_p[k]$ is also recursively determined from (21a) to (21c) and targeted to minimize the mean square error of the final estimation \tilde{INL}_k . An intuitive interpretation of the Kalman gain is as follows. If the measurement error is comparatively small, consequently R small, the measured value is better representing the true value, and vice versa. So when R is small, $K_p[k]$ is close to 1 and the final estimation is more close to the measured value. When R is large, the Kalman gain is smaller than 1 and the final estimation is more dominated by the predicted value.

C. INL_k Test for Pipeline ADCs with Kalman Filtering

The DNL_k usually has some repetitive patterns for ADCs

with other architectures. One of the most widely adopted architectures is the pipelined structure, for which DNL_k and $DNL_{k+N/2}$ are strongly correlated. ([10] provides a very good description of pipeline ADCs.) Under the case of such ADCs, we can first identify the deterministic component in DNL_k and make the residual part independent of each other.

Taking the pipeline ADC as an example, we use the average of DNL_k and $DNL_{k+N/2}$ as an approximation of their deterministic component,

$$DNL'_k = DNL'_{k+N/2} = \frac{D\hat{N}L_k + D\hat{N}L_{k+N/2}}{2}. \quad (22)$$

The true DNL_k can be written as

$$DNL_k = DNL'_k + d_k, \quad (23)$$

where d_k is a random component of zero mean, contributed by the random errors in the ADC and the test errors in the code-density method, satisfying

$$E\{d_k d_l\} = C' \cdot \delta_{kl}. \quad (24)$$

The value of d_k , instead of DNL_k in the flash case, will be predicted by Kalman filtering based on the code-density test results. Replacing (23) into (18a), we can see that the Kalman filtering steps need to be modified as

$$R_e[k] = P[k] + R; \quad (25a)$$

$$K_p[k] = P[k]R_e^{-1}[k]; \quad (25b)$$

$$P[k+1] = P[k] + C' - K_p[k]R_e[k]K_p[k]; \quad (25c)$$

$$\tilde{INL}_k = \tilde{INL}_{k|k-1} + K_p[k](\hat{INL}_k - \tilde{INL}_{k|k-1}); \quad (25d)$$

$$\tilde{INL}_{k+1|k} = \tilde{INL}_k + DNL'_{k+1}. \quad (25e)$$

The major difference between (25) and (21) is in the prediction step. For the flash ADC, the DNL_k is modeled as completely random with zero mean, so the best prediction of INL_{k+1} is \tilde{INL}_k , (21e). For the pipelined ADC, the deterministic component should be added to get the prediction as in (25e), which is intuitively reasonable and mathematically correct. For other architectures of ADCs, similar variations as (25) can be found when appropriate circuit level models are used.

D. Performance Analysis for Proposed Algorithm

We use the flash ADC equations as an example to study how Kalman filtering can help improve test accuracy. The variance of the final prediction error, $P[k]$, can be calculated from (21) by letting k goes to infinity so that the variables settle to their steady state values, notated with a subscript "s",

$$\begin{aligned} R_{es} &= P_s + R, \\ K_{ps} &= P_s R_{es}^{-1}, \end{aligned} \quad (26)$$

$$P_s = P_s + C - K_{ps} R_{es} K_{ps}.$$

Solving above equations gives the final prediction error

variance as

$$P_s = \frac{C + \sqrt{C^2 + 4RC}}{2}, \quad (27)$$

where the negative solution is discarded. This is the variance of the error between $\tilde{INL}_{k|k-1}$ and INL_k . P_s is smaller than R when C is much smaller than R . This is usually true for flash ADCs, since the standard deviation of e_k could be about 1 LSB or larger for median and high resolution ADCs while the standard deviation of DNL_k is much less than 1 LSB. This means that even the predicted value of INL_k can have better accuracy than the code density results.

The final estimation error can be calculated as [12],

$$\begin{aligned} R_e &= \text{var}\{\tilde{INL}_k - INL_k\} \\ &= P_s(1 - K_{ps}) = \frac{P_s R}{P_s + R} = \frac{\sqrt{C^2 + 4RC} - C}{2}. \end{aligned} \quad (28)$$

This variance is strictly smaller than R , which means using the Kalman filter can effectively improve the accuracy of code density testing. Note that for the case of pipeline ADCs, the C in (27) should be replaced with C' .

Equations (21) and (25) show that the computational complexity of the Kalman filter algorithm is proportional to N . For each INL_k value, it takes some basic operations. This complexity is acceptable with current data processing power and does not introduce any significant computation overhead. While the proposed algorithm can achieve the same performance as the conventional algorithm with much fewer sample and consequently shorter testing time, this algorithm has obvious benefits for testing high-resolution ADCs.

IV. Simulation Results

Simulations have been done to validate the effectiveness of the proposed algorithms on various types of ADC architectures and with different test parameters. All results show that the advantage of using Kalman filtering in linearity test is significant.

A. Single ADC Test

The proposed algorithm was first compared with the conventional code-density algorithm for testing a 16-bit flash ADC. DNL_k of the flash ADC has zero mean and a standard deviation of 0.5% LSB. The true INL is 1.39 LSB. The input signal is an ideal ramp plus a white Gaussian noise with zero mean and a standard deviation of 0.5 LSB. The average number of samples per code bin is 32. The INL_k estimation results are given in Fig. 2 and 3. On the top of Fig. 2, true INL_k and estimated values with the conventional code-density algorithm are plotted. The estimation error is plotted on the bottom with a standard deviation of about 0.1 LSB. The INL is estimated as 1.68 LSB. Fig. 3 shows the results when Kalman filtering is used, with true and estimated INL_k on top and estimation

error on bottom. The estimation error has a standard deviation 0.04 LSB. The INL is estimated as 1.50 LSB. Comparing Fig. 2 to Fig. 3, we can see that Kalman filtering significantly reduced the estimation error, by more than 50%, and the estimated INL is closer to the true value, achieved 0.1 LSB accuracy at 16-bit level.

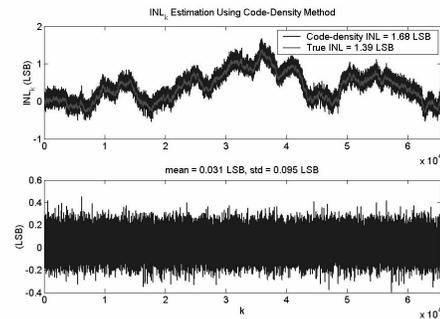


Fig. 2. INL_k estimation using code-density method.

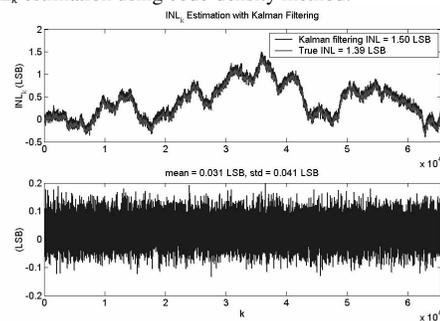


Fig. 3. INL_k estimation with Kalman filtering.

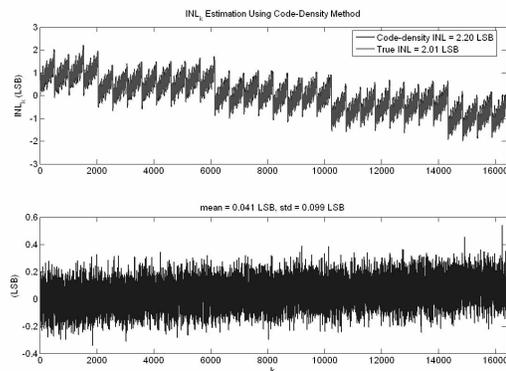


Fig. 4. INL_k estimation using code-density method. Top: True and estimated INL_k ; Bottom: INL_k estimation error.

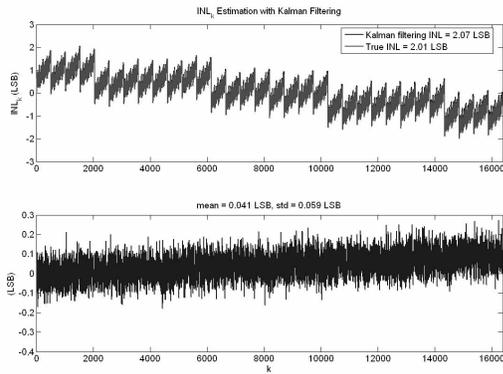


Fig. 5. INL_k estimation with Kalman filtering. Top: True and estimated INL_k ; Bottom: INL_k estimation error.

A 14-bit pipeline ADC with linear errors was then tested using two methods in simulation. Its true INL is 2.01 LSB. The input noise has a standard deviation of 0.5 LSB at the 14-bit level. The average number of samples per bin is 32. The operation as discussed in (23) was applied to the first three stages of the ADC. The results using conventional and Kalman filtering methods are plotted in Fig. 4 and 5, respectively. We can see that the INL_k estimation error with Kalman filtering is much smaller. The standard deviation of the estimation error with the conventional method is 0.1 LSB and is reduced to 0.06 LSB when Kalman filtering is used. The estimated INL with Kalman filtering, 2.07 LSB, is also closer to the true value than the conventional method, 2.20 LSB.

B. Statistical Result

Firstly, the relationship between the accuracy improvement of the Kalman filtering method and the number of samples was studied. For saving simulation time, 14-bit flash ADCs were used. The standard deviation of these ADCs' DNL_k is 0.8% of LSB. The INL of these ADCs ranges from 0.4 LSB to 2 LSB. Input noise has a standard deviation of 0.5 LSB. For each different number of samples, the RMS value of INL_k estimation error was collected from 128 ADCs and the averaged values are plotted in Fig. 6. The Kalman filtering method can reduce the test error by 50% and can achieve the same accuracy level by using only about 1/10 of the number of samples as compared to the conventional code-density method. The mean values of INL estimation error are compared in Fig. 7 for the two methods. We can see significant improvement in test results introduced by the proposed algorithm as well.

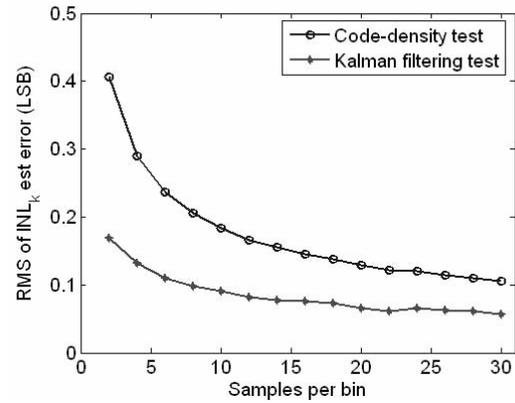


Fig. 6. RMS value of INL_k estimation error vs. number of samples.

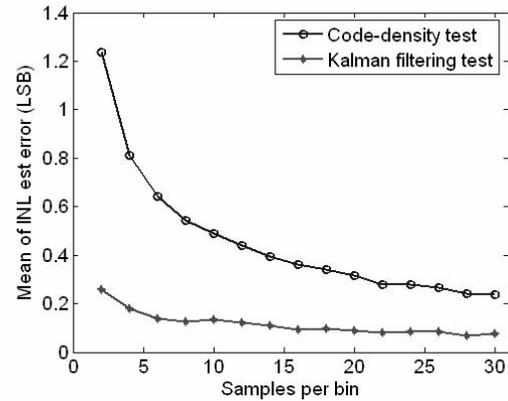


Fig. 7. Mean value of INL estimation error vs. number of samples.

The relationship between the accuracy improvement and the input noise was investigated as well. 14-bit flash ADCs with same error characteristics as in the previous paragraph were test with 32 samples per bin on average. The RMS values of the INL_k estimation error and the mean values of INL estimation error are plotted as a function of the standard deviation of the input noise in Fig. 8 and 9, respectively. Each data point is an average from 128 ADCs.

The INL_k estimation error was obviously smaller with the modified Kalman filtering algorithm and the INL estimation error was reduced even more percentage wise. The proposed Kalman filtering algorithm can improve the test performance of the code-density method and achieve a high accuracy level under different noise situations with a reasonable number of samples.

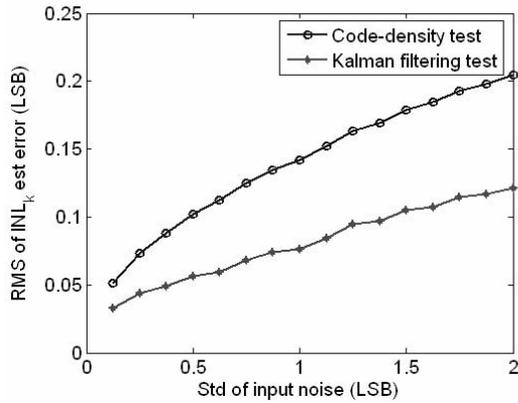


Fig. 8. RMS value of INL_k estimation error vs. standard deviation of input noise.

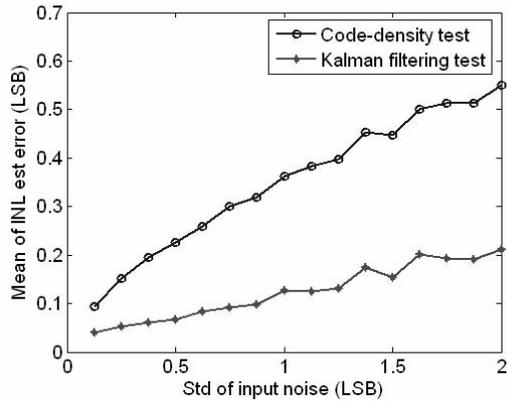


Fig. 9. Mean value of INL estimation error vs. standard deviation of input noise.

C. Discussion

The results in Fig. 6 is in agreement with (17). When the sampling density was multiplied by 4, the INL_k estimation errors can only be reduced by 2 times, see those circles in Fig. 6. This means increasing the number of samples and consequently the test time can not efficiently improve testing accuracy. We can achieve such improvement by applying a short piece of code during the data processing stage. Kalman filtering is very cost-effective.

The capability of the proposed Kalman filtering algorithm for significantly reducing linearity test time can be seen from Fig. 6 and 7. For achieving 0.2 LSB accuracy in INL test, the Kalman filtering algorithm only requires a few samples while the conventional code-density needs more than 30 samples per code bin on average. This can save 90% of test time and make it possible for full-code test of some highest-resolution ADCs.

Analysis of (17) and Fig. 8 show that reducing noise also is not efficient in improving test accuracy. Reducing the standard deviation of the noise by a factor of 4 can only reduce the standard deviation of the INL_k estimation error by a factor of 2. And the noise can not be unlimitedly reduced, because the kT/C noise always exists in an ADC and is determined by design. For ADCs with higher than 16-bit resolutions, it is difficult to reduce the noise to less

than the half LSB level. Fig. 8 and 9 show that the proposed algorithm can achieve 0.1 LSB test accuracy under such a noise level with a few samples per code while the conventional algorithm cannot even with tens of samples.

V. Experimental Results

The experiment platform was a Teradyne A580 Advanced Mixed Signal Tester. Linear ramps were generated by a 20-bit multi-bit delta-sigma DAC for the code-density test. To avoid sudden jump in the output of the waveform generator, triangular waveforms were implemented with appropriate coding of the input to the DAC.

Experiments were taken on 16-bit commercial ADC for validating the proposed algorithm with Kalman filtering. A typical INL_k plot of the ADC used in the test is given on the top of Fig. 10. This result was measured with 32 samples per code on average and will be used as a reference to evaluate the results from histogram data collected with a smaller sample density.

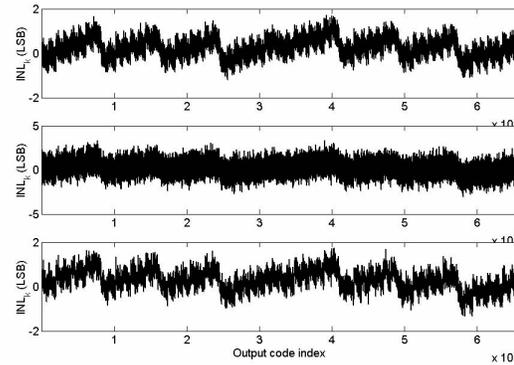


Fig. 10. INL_k measurement for a 16-bit ADC.

INL_k of the same ADC was then tested with D_s equal to one sample per code and the result is plotted in the middle of Fig. 10, which does not track the top curve very well. Significant noise can be observed. The difference between the results using 32 samples per code and one sample per code is plotted on the top of Fig. 11. The difference has an error band of about 2 LSB. The result with 32 samples per code is pretty accurate. The difference mostly came from the one-sample-per-code result, because the additive noise in the system and quantization effects could not be well averaged out by the small number of samples.

Kalman Filtering was applied to the one-sample-per-code result as discussed in Section III. The values of R and C in (21) were first characterized for the system and the ADC under test from previous measurement results. The new INL_k is plotted on the bottom of Fig. 10. It matches with the 32-samples-per-code result very well. Even the small details are almost identical in the two plots. The difference between them is plotted on the bottom of Fig. 11, which are nearly all less than 1 LSB at the 16-bit level this time. By comparing the results, we conclude that the proposed

algorithm can reduce the linearity test error by more than 50% with a very small sample density.

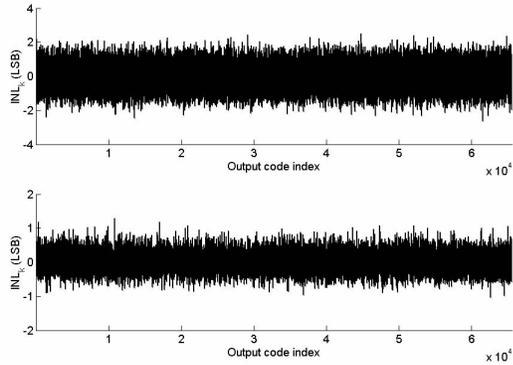


Fig. 11. Estimation error in INL_k test.

More experiments with the same setup were done to further investigate the improvement introduced by Kalman filtering, see Fig. 12. The ADC was tested with one sample per code 32 times. For each run, the results estimated with and without Kalman filtering are compared to the reference result from the 32-samples-per-code test and the difference are taken as errors. The top of Fig. 12 shows the improvement in INL estimation, where the INL estimation errors of one-sample-per-code tests are drawn in circles and the estimation errors of one-sample-per-code tests with Kalman filtering are drawn in diamonds. It can be observed that the errors were reduced from about 2 LSB to less than 1 LSB. The bottom of Fig. 12 shows the improvement in the variance of INL_k estimation errors for different runs. The variance was reduced from about 0.4 LSB^2 to 0.07 LSB^2 using Kalman filtering.

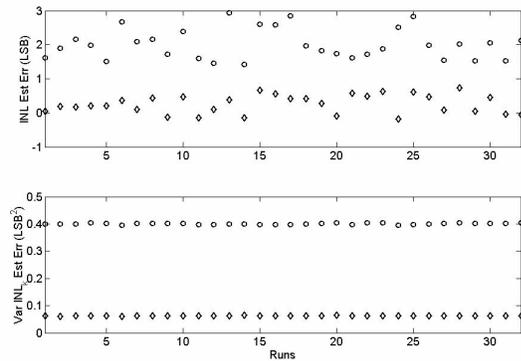


Fig. 12. Performance comparison: with and without Kalman Filtering.

Some more performance validation was done for the algorithm. Two samples per code were used and the results are summarized in Fig. 13. As the convention in Fig. 12, circles represent results without Kalman filtering and diamonds represent results with filtering. The performance improvement with Kalman filtering is still obvious. Because of the increase in the number of samples, both the INL estimation errors and variances of INL_k errors became smaller. The variance of INL_k errors was reduced by a

factor of two when the number of samples doubles, which is in agreement with the discussion in Section II.

Kalman filtering can significantly reduce the errors in ADC test results. From the experimental results, we can conclude that the measured ADC specifications from 32 samples per code and from one or two samples per code with Kalman filtering are very close to each other. That means we can dramatically reduce the test time, by a factor of 10 or more. Since the data processing using Kalman filter does not introduce any hardware overhead and a little computation overhead which is easy to get in nowadays digital technologies, it can help engineers effectively cut down the test cost.

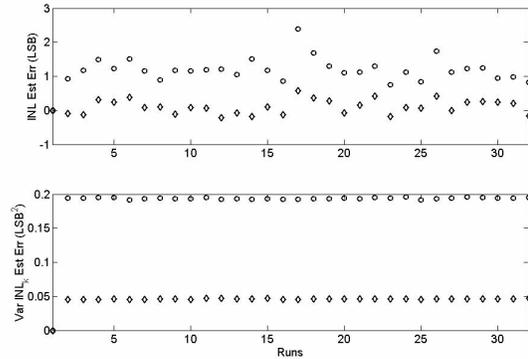


Fig. 13. Results with two-samples-per-code tests.

VI. Conclusion

The performance of the code-density method for ADC linearity testing is analyzed in this work and the optimization algorithm using Kalman filtering for improving test accuracy of the conventional method is introduced for flash and pipeline ADCs. Simulation and experimental results show that the proposed algorithms can effectively reduce the variance of INL_k test errors when using the same number of samples as the conventional algorithm. This method can effectively reduce the test time, by 10 times or more, and provide accurate test results of ADC static linearity. The proposed algorithm provides an enabling solution to the problem of cost-effective test of high-resolution ADCs with 16-bit or higher resolutions. It can also significantly reduce the test cost for high-volume medium and low-resolution ADCs.

Acknowledgements

The authors would like to thank Turker Kuyel and Kumar Parthasarathy with Texas Instruments for their help on collecting the experimental data.

References

- [1] *International Technology Roadmap for Semiconductors*, 2003 edition, available online: <http://public.itrs.net>.

- [2] J. Doernberg, H.-S. Lee, and D.A. Hodges, "Full-Speed Testing of A/D Converters," *IEEE J. Solid-State Circuits*, SC-19, pp. 820-827, December 1984.
- [3] J. Blair, "Histogram measurement of ADC nonlinearities using sine waves," *IEEE Trans. Instrum. Meas.*, vol. 43, pp. 373-383, June 1994.
- [4] *IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters*, Dec 2000.
- [5] P.D. Capofreddi and B.A. Wooley, "The efficiency of methods for measuring A/D converter linearity," *IEEE Trans. Instrum. Meas.*, vol. 48, pp. 763 – 769, June 1999.
- [6] L. Jin, K. Parthasarathy, T. Kuyel, D. Chen and R. Geiger, "Accurate testing of analog-to-digital converters using low linearity signals with stimulus error identification and removal," *IEEE Trans. Instrum. Meas.*, vol. 54, pp. 1188 – 1199, June 2005.
- [7] G. Casella and R.L. Berger, *Statistical Inference*. Pacific Grove, CA: DUXBURY, 2001.
- [8] D.G. Luenberger, *Optimization by Vector Space Methods*. New York, NY: John Wiley and Sons, 1997.
- [9] D. Johns and K. Martin, *Analog Integrated Circuit Design*. New York, NY: John Wiley & Sons, 1997.
- [10] A.N. Karanicolas, H.-S. Lee, and K.L. Bacrania, "A 15b 1Ms/s digitally self-calibrated pipeline ADC," *IEEE J. Solid State Circuits*, vol. 28, pp. 1207-1215, Dec 1993.
- [11] G. W. Roberts, "DFT Techniques for Mixed-Signal Integrated Circuits," *Circuits And Systems In The Information Age*, IEEE Press, pp. 251-271, June 1997.
- [12] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf.