Pipeline ADC Linearity Testing with Dramatically Reduced Data Capture Time

Zhongjun Yu, Degang Chen, Randy Geiger, Iowa State University
Yiannis Papantonopoulos, Texas Instruments

Abstract:

A system identification based method for testing pipeline ADC's linearity is presented. In the method, the pipeline ADC is described by an identifier model consisting of a set of nonlinear equations parameterized with some unknown parameters. A small number of input output response data points of the ADC is then used to identify those unknown parameters. The identified model is then used to compute the ADC's full-code linearity information. Comparing to standard histogram based full-code linearity test methods, the proposed method can reduce the data capture time by a factor of 100 to 1000 without appreciably degrading the testing accuracy. Both simulation results and experimental results are included which demonstrate the efficacy of the proposed method.

1 Introduction

Due to the extensive use of data converters in many electronic products, the performance of ADCs and DACs significantly affects the performance of many integrated circuits and systems [1, 2]. In particular, sufficient linearity performance of ADCs is critically important to many applications in signal processing, communications, instrumentation, medical applications, and other areas. As the performance of such products improves and as more and more mixed-signal functions are integrated in a system on a chip, testing of ADC linearity performance becomes increasingly challenging.

The most ubiquitous approach to ADC static linearity testing is well described in several IEEE standards and involves the use of precise linear ramp signals or ultra-pure sine wave signals as stimulus input to the ADC [3]. The ADC output codes are then sorted into histogram counts which are then processed by well established method to create the INL (integral nonlinearity) and DNL (differential nonlinearity) information about the ADC. To ensure M samples per code, $M*2^n$ (for ramp input) or $\pi*M*2^n$ (for sinusoidal input) samples should be collected according to IEEE standards. For high resolution ADCs (which are typically slow), the data acquisition time may be prohibitively long. To cut down the test time, sometimes reduced-code testing is used instead of full code testing. In this case, only a small subset of the ADC output codes are guaranteed. A commonly used technique for such testing is the servo-loop feedback method due to the need for searching for certain transition points [4]. This technique is quite slow and significantly limits the total number of ADC output codes that can be tested.

In this paper we introduce a new black-box-model based identification and testing approach for testing pipelined ADCs. The proposed method uses a dramatically reduced data set (100s or 1000s times less) to identify key parameters in the black-box model of each stage in a pipeline ADC. It then uses the identified parameters together with the model to predict all the transition points of the ADC and computes the estimated full-code INL_k curve. In the next section, we will present the proposed system-ID based pipeline ADC test method and briefly describe the key ideas behind the algorithm. Due to space limitation, detailed theoretical analysis and algorithm development will not be provided. In section 3 we will present simulation results comparing the identified ADC full code INL_k curve based on the proposed method against the true INL_k curve based on traditional method. In section 4 experimental results using data generated at Texas Instruments will be presented, demonstrating that the proposed method achieved similar testing accuracy to that by the traditional method while using only <0.5% of data points. Finally, conclusion remarks are given in section 5.

2 System-ID based ADC INL testing

A basic pipelined ADC is shown in Fig. 1. In this figure it is assumed that a single reference voltage is applied to the ADC although internal to the individual stages sub-references can be locally generated. The last stage is called the tail stage and it does not generate a residue signal. The tail stage is modeled as a resistor-string flash ADC with random variations in the resistance values, even though the actual tail stage in the ADC may be made of several pipelined stages. All the stages before the tail stage are modeled as 1-bit per stage nominally gain 2 stages, even though the actual ADC may have sub-radix 2 stages or stages with multiple bits per stage. Our stage is simply a black box model of an equivalent stage corresponding to each significant bit in the ADC output code. The analog signal path is the path from the input of the first stage to the input of the tail stage. Each output d_1, d_2, …, d_m is a single bit digital output except that d_m may be several bits wide. An actual ADC will have additional shift registers and logic circuitry for combining each stage's output bit into an overall output code. In our model here, all

of these are lumped into the equivalent stages.

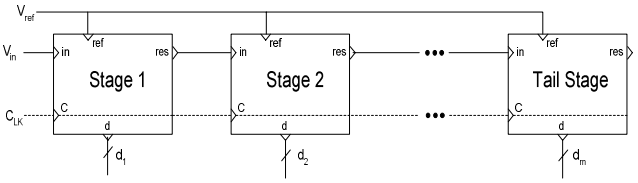

Figure 1 Basic Pipelined ADC

A block diagram of a basic 1-bit stage is shown in Fig. 2. The 1-bit ADC is essentially a comparator. Error in the comparator is modeled as random variations in the threshold voltage. The ADC output d_j is used to determine the voltage of the 1-bit DAC output. The DAC error, the DAC reference error, and the gain error from the DAC output to the residue are all lumped together as one gain error from the ideal DAC reference to the residue.

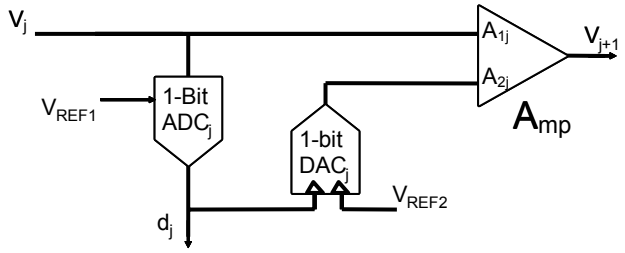

Figure 2 Block Diagram of Stage j of a Pipeline ADC

The nominally gain-2 amplifier is modeled after a switched capacitor amplifier as shown in Figure 3. The diagram in Figure 3 helps us identify how different variables are related to the residue output. In our simulation, the actual ADC stage is modeled by generating capacitors with random variations and finite gain amplifiers with random gain errors and random amount of nonlinearity. The identification algorithm does not know the detailed model of each stage. Instead, the algorithm uses a black box pseudo-static model for each stage as given by the following equation:

$$V_j = g_{1j}V_{j+1} + g_{dj}V_{DAC\text{Ref}} + V_{\text{offset}} + g_{2j}V_{j+1}^2 + g_{3j}V_{j+1}^3 + \ldots$$

Notice that instead of the traditional forward model in which residue is expressed as an expression of other variables, we are using an inverse model or an identifier model in which the input to the stage is expressed in terms of the residue voltage and other variables. The g's are roughly speaking the inverse of the gains. All sources of nonlinearities are represented by a Tailor series expression of the residue voltage. V_{DACRef} is the ideal DAC output voltage when an ideal DAC reference is used. The V_{offset} term captures the error terms due to all sources of offset errors. Although we included the V_{offset} in the stage model in the above equation, our algorithm uses only one offset voltage for the entire pipeline. The g's for each stage and the one V_{offset} for the entire pipeline are the parameters to be identified. For example, if we identify the first 10 stages of a 16 bit ADC with nonlinearities up to the fifth power, then we will have a total of 6X10+1=61 parameters to identify. Theoretically speaking we will only need 61 linearly independent input output measurements to identify all of the 61 parameters. Since our tail stage is not accurate and there will be quantization and measurement noise, we will use many more points than necessary.

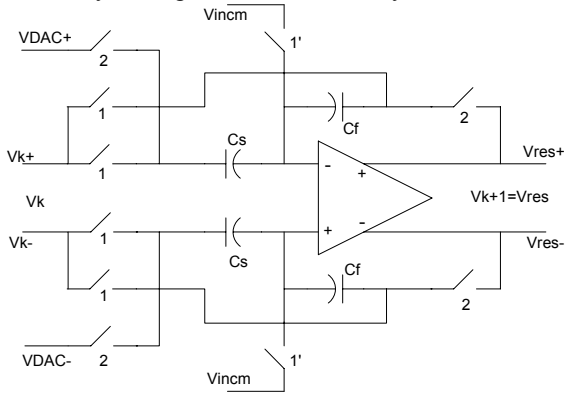

Figure 3 Switch capacitor model of the gain stage

In the identification process, both the analog input to the ADC and the ADC output code are known. Based on an initial guess of the parameters and the measured ADC output code, the black box is used to make a prediction as to what the input voltage is. The actual ADC input and the prediction are used to generate a prediction error. This is done with each input output pair. If M data points are available, M error terms can be formed. Based on this, an optimization problem is set up in which the optimal parameters are solved that minimizes the total summation of the squares of the expected values of the error terms. The problem then becomes a standard nonlinear statistical estimation problem and the Newton-Ralphson iteration method is used to solve the optimization problem.

Once the parameters are identified, the model equations can then be used to compute the transition voltages of the ADC. Since the ADC is modeled with the inverse model or the identifier model, we do not give slowly varying analog input voltages to the ADC and search for the transition voltages. That would be a very time consuming process. Instead, we start with the 2^n possible output codes and use the inverse model to compute what the corresponding input voltages should be. Due to nonlinearities of the ADC, some of the ADC codes should be missing in reality. Since the inverse model uses all output codes, this leads to a computed input output transfer curve that is multiple valued. This problem can be eliminated after the predicted inputs for all codes have been computed by simply removing the invalid brunches. From the transition voltages, the full code INL_k can be finally computed.

3 Simulation Results

To verify the efficacy of the algorithm, we have conducted both simulation study and experimental testing. In the simulation study, we simulated pipeline ADC at various resolution levels and various linearity levels. Our algorithm can handle pipeline ADCs with or without over range protection. Here we will include the simulation

results for one simulated 16-bit pipeline ADC. In contrast to the traditional full code linearity testing methods which would require 2 to 10 million samples depending on the code density requirement, our algorithm uses a pseudo static sine wave as input and takes only 2048 data points. From the 2048 input output data points, our identification algorithm identifies the parameters of a black-box model of the ADC. As we pointed out before, the algorithm offers the choice of the number of stages to be identified. The curve in Figure 4 below is the identified full code INL of the 16-bit ADC based on 2048 measurements. 12 stages are identified. For the nonlinearity, we only kept up to the third order terms. The last four stages, though they are pipelined stages, were taken to be a 4-bit flash structure with random errors. If we change the number of stages to be identified, similar results will be obtained.

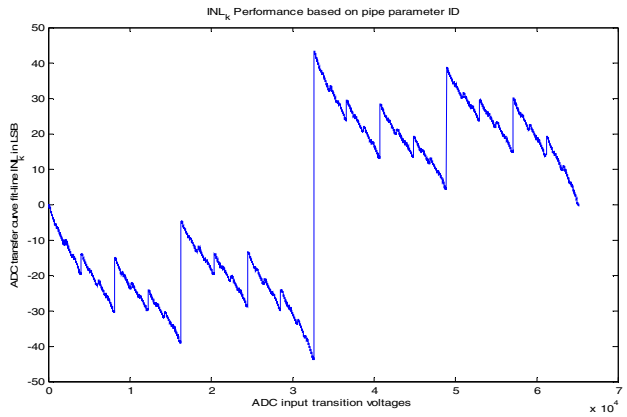

Figure 4 Full code linearity testing of a 16-bit ADC with only 2048 input output measurement points

For comparison, the same ADC is also tested with a traditional histogram method using ideal linear ramp as input and taking $2^{16}*64 = 4$ million samples. Two curves are shown in Figure 5: one is the error of the ADC transfer curve from an ideal ADC's linear transfer curve and the second one (the one that starts and ends at 0 LSB) is the error of the ADC transfer curve from the end-point fit line. Comparing to Figure 4, we can see that the proposed algorithm has accurately identified the full code INL curve of the 16-bit ADC with only 2048 samples.

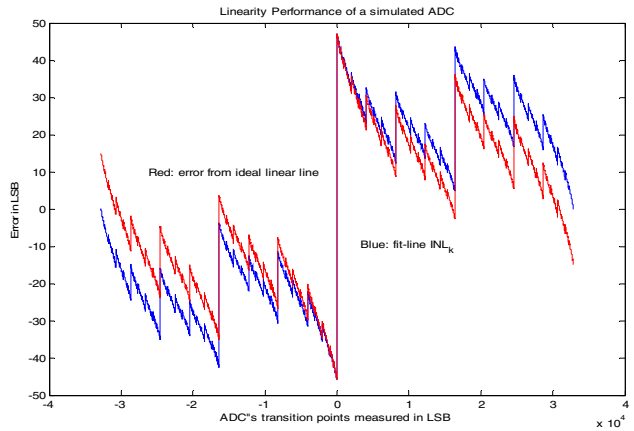

Figure 5. Linearity performance of a 16 bit pipeline ADC

This ADC corresponds to an ADC with inter-stage gains that are less than 2 for over-range protection. As a result, there are many missing codes at major bit transitions. If mechanisms are provided for the missing code to be removed, then the ADC linearity performance will be much better. After that, the transfer curve INL will represent how uniformly distributed the transition points are. In our algorithm, the black box model is in the inverse form and it is really easy to find all the transition voltages once the model parameters are identified. Figure 6 below illustrates the identified transition point linearity performance of the 16-bit pipeline ADC. A total INL of about 5.3 LSB is indicated. For comparison, the true transition point INL of the 16 bit ADC is also computed by scanning through the ADC input range to find all transition points. The true transition point INL is shown in Figure 7, indicating a total INL of about 4.8 LSB. Comparing Figure 6 and Figure 7, we see that the two INL curves are very similar to each other and the difference is at the 16-bit level.

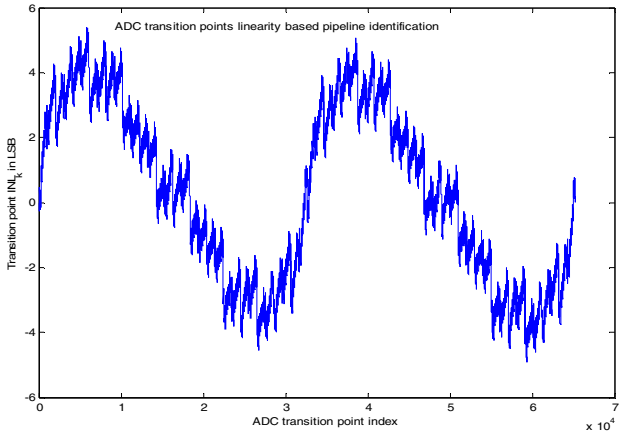

Figure 6 Transition point INL of a 16-bit ADC based on pipeline identification with only 2048 samples

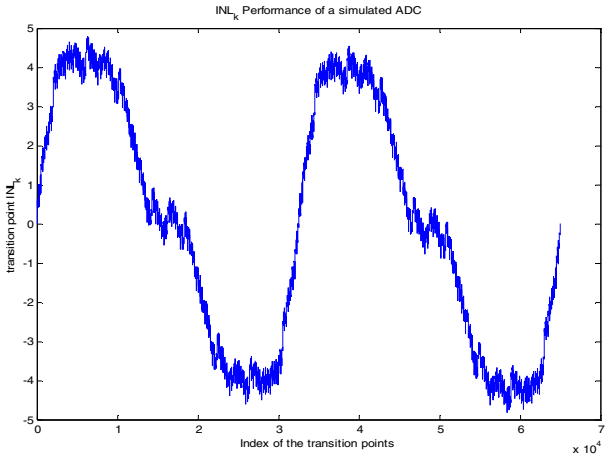

Figure 7 Transition point linearity of the 16-bit ADC

4 Experimental results

Experimental data were collected at Texas Instruments' Data Converters Lab. Description about the experimental set up is omitted here due to space limitations. A 14 bit pipeline test ADC is tested with a sine wave signal as the input. A little over 4500 input output response points corresponding to an integer number of signal periods were captured. The ADC output codes at these points are used in the black-box inverse model of the ADC to generate the predicted ADC input voltage. The prediction errors are then used with Newton-Ralphson iteration to find an optimal solution of the pipeline parameters. The identified parameters are then used with the ADC model to compute the full code INL curve of the ADC. This can be done for various numbers of identified stages. It turns out that the estimated INL curves are very similar to each other for the number of identified stages from 5 to 12. With smaller number of identified stages, the curve looks noisier. Figure 8 shows the identified INL curve when 10 out of the 14 stages are identified. Figure 9 shows the identified INL curve when 7 stages are identified. From the identified curves we can make the following observations about the ADC under test. The estimated INL curves are tilting down going to the right, indicating that the ADC codes near the two ends are a little bit wider than average. The third stage of the ADC is the one contributing the most error, since 8 distinctive pieces can be seen in each curve. The 4th stage contributes the next largest error but the first and second stages are very good. Finally, the overall INL of the ADC is somewhere between 1 and 1.5 LSB.

For comparison, Figure 10 shows the INL curve of the ADC as measured with the traditional method using over one million data points, which is over 200 times more than what is used in the proposed method. Even though the INL curves don't look exactly the same as the estimated ones, the same qualitative conclusion can be drawn from the traditional measurement also. This validates the proposed method.

5 Conclusion

We have introduced a new method for ADC linearity test aimed at greatly reducing the test time associated with capturing large amounts of data for high resolution ADCs. The new method is based on identifying the nonlinear inverse model of the pipeline ADC using a small number of I/O response points. It can cut down the data acquisition time required for full code testing by a factor of 100s to 1000s. Furthermore, it can achieve the test time reduction without adversely affecting the ADC testing accuracy. Both simulation and experimental testing results validate the proposed method. Although the paper title and the text specifically mentions pipeline ADCs, the method can be extended to other high resolution ADC architectures such as SAR.

6 References

[1] *International Technology Roadmap for Semiconductors, 2001 Edition and 2002 Update*, http://public.itrs.net/

[2] Kuyel, T., "Linearity Testing Issues of Analog to Digital Converters," International Test Conference, 1999.

[3] IEEE Std 1241-2000, IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters.

[4] M. Burns and G.W. Roberts, "An Introduction to Mixed-Signal IC Test and Measurement," Oxford University Press, New York, USA 2000.

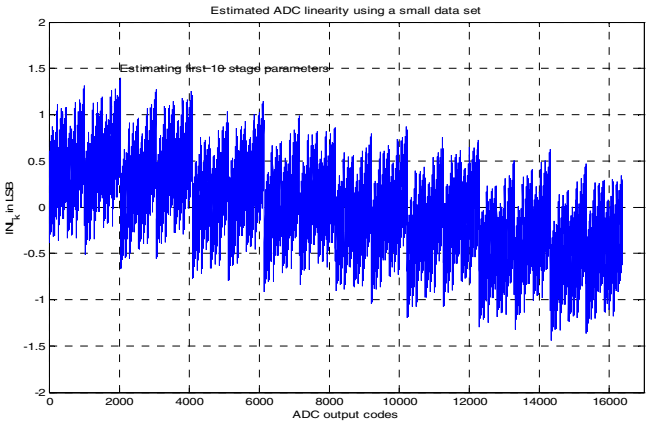

Figure 8 The full code INL curve of a 14-bit ADC computed by the proposed algorithm using only 4500 measurement data points

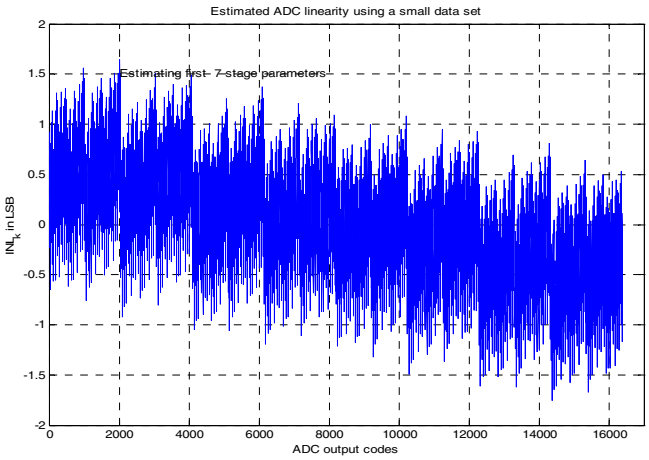

Figure 8 The full code INL curve of a 14-bit ADC computed by the proposed algorithm using only 4500 measurement data points

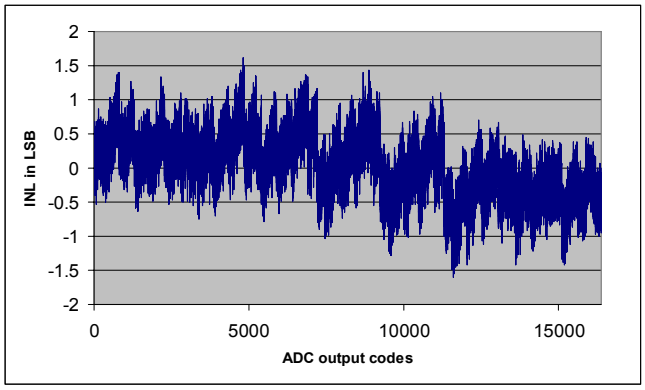

Figure 10 ADC INL curve computed by traditional method using over one million measurement data points