BEHAVIORAL MODELING AND SIMULATION OF DATA CONVERTERS

Edward Liu, Georges Gielen, Henry Chang, Alberto L. Sangiovanni-Vincentelli, and Paul R. Gray
Department of Electrical Engineering & Computer Sciences
University of California
Berkeley, California 94720

Abstract

This paper presents a behavioral representation for the class of Nyquist rate A/D converters that captures the nominal behavior, as well as all the statistical variations. To describe noise effects we use a joint probability density function. To describe behavioral effects due to process variations we use a covariance matrix, Σ_t. Applications of the model include identification of important A/D error sources, efficient computation of the distributions of integral nonlinearity (INL) and differential nonlinearity (DNL), signal-to-noise plus distortion ratio (TSNR), and efficient worst case and Monte Carlo system simulations. Finally, parameter extraction results are presented that agree well with actual measurements.

1 Introduction

The verification of data converters poses a significant challenge to the design community, partly due to the need for higher yield at high resolutions, and partly due to the emphasis on predicting converter performance when used in a system since more converters will be integrated with entire systems. For example, to verify adequate yield of a high resolution converter, it is not enough to find a deterministic circuit response from a deterministic input. The entire transfer curve, as well as its variations due to process variations, has to be estimated. To compute circuit yield, the distributions of integral nonlinearity (INL) and differential nonlinearity (DNL) are needed. Furthermore, for systems containing these converters, system designers need to estimate system performance degradation, such as signal-to-noise and distortion ratio (TSNR), due to converter non-idealities.

The traditional SPICE simulation is clearly not viable because (a) the simulation speed is too slow for worst case or Monte Carlo simulations for yield predictions (b) SPICE cannot calculate noise effects of sampled data circuits, so TSNR cannot be calculated for architectures with such circuits. Recent approaches[1] model components of A/D converters with behavioral models and are capable of finding the transfer function in a more timely fashion. This type of modeling can be expressed mathematically as

$$code = A(V_{in}, \hat{v}), \hat{v} = realization(v) \tag{1}$$

where V_{in} is the input, and $\hat{v}$ is a set of parameters describing process variations. Since the sources of non-idealities in analog circuits are statistical, $\hat{v}$ is a realization of a multivariate distribution v. There are two problems with this behavioral modeling approach. (a) Suppose v is an m-variate, m can be very large because every circuit element that affects the transfer function has process variations. Also, it is unclear which realizations $\hat{v}$ should be selected from v. For example, to verify worst case system performance without detailed knowledge of the A/D architecture, 2^m simulations have to be performed. (b) Noise effects are not modeled, so TSNR cannot be calculated.

In this paper we present a behavioral model[2] for the class of Nyquist rate A/D (Section 2) and some parameter extraction techniques for effects due to process variations (Section 3.1). Circuit designers benefit from using this model because there is a systematic and efficient way to compute from model parameters the important A/D error sources (Section 3.5), the distributions of INL and DNL (Section 3.2), and TSNR (Section 3.4). The efficiency is illustrated with an optimization example (Section 3.3). System designers benefit from using this model because the model statistically summarizes all relevant A/D non-idealities, speeding worst case and Monte Carlo system simulations (Section 4).

2 New behavioral model

2.1 Modeling process variations

The transfer curve of a Nyquist A/D plots the output code on the range against the input continuous value on the abscissa. (Figure 1). Assuming the nominal A/D has N bits of resolution, is designed to be monotonic and has no missing codes, the nominal transfer curve can be characterized by $2^N - 1$ real numbers in a vector, μ_t, representing the expected value of the transition points $t_1, t_2, \ldots$ shown in Figure 1. Thus, μ_t represents the expected transitions with no process variations.

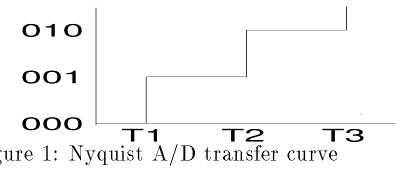

Figure 1: Nyquist A/D transfer curve

Due to statistical process variations, the transfer curve has statistical variations. The statistical variations can be conveniently represented by a distribution, t, which we can find by making the following assumptions. (1) *Process variations are the cumulative effects of many integrated circuit fabrication steps. Thus, according to the Central Limit Theorem in probability theory, the distribution of the process variations, v, is approximately multivariate normal.* (2) *In general, an A/D is designed to be insensitive to process variations which are relatively small. Thus, the transition points, t, change relatively little with respect to process variations. So, to a good approximation, t changes linearly with respect to the process variations.*

From the linear approximation in assumption 2,

$$t = S_v v + \mu_t \tag{2}$$

where v is a random vector of parameters, corresponding to process variations such as errors in resistors and capacitors values. From assumption 1 and (2), the covariance matrix of t is given by

$$\Sigma_t = S_v \Sigma_v S_v^T \tag{3}$$

where Σ_v is the covariance matrix of v, and S_v is the sensitivity matrix $\frac{\partial t}{\partial v}$. Thus, the distribution of t is

$$t \sim normal(\mu_t, \Sigma_t) \tag{4}$$

In general, the rank r of Σ_t is much less than full rank since not all the transition points are independent. In other words, any realization of t must fall into a space spanned by r independent basis vectors. A set of such independent basis vectors can be found by the singular value decomposition of Σ_t. Mathematically,

$$\Sigma_t = U \Sigma_c U^T \tag{5}$$

where U is an 2^N x r transformation matrix, Σ_c is an r x r diagonal matrix, and r is the rank of Σ_t. The columns of U are called the **error signatures** [3] and form a set of orthonormal vectors spanning the **error space**. For storage and computational efficiency we can express t with the following equation instead of (2),

$$t = Uc + \mu_t \tag{6}$$

where $c \sim normal(0, \Sigma_c)$ is a small zero mean multivariate normal distribution with r statistically *independent* variates. The parameters U and μ_t are deterministic and unique for each A/D architecture, while the distribution c represents a mixture of the relevant process variations.

2.2 Modeling noise

Due to noise, an A/D produces a wrong output with finite probability. For example, if the input is near a transition point, the probability of getting a wrong output code, the code at the other side of the transition point, is non-zero. As a result, the noise effects depend on the input, and they become more significant as the input gets closer to a transition point. Mathematically, this is represented by a probability distribution function of the output codes for any input value. Our behavioral model captures this behavior by a joint probability density function of the input V_{in} and the output code for a given realization of the transfer curve $\hat{t}$:

$$f(code, V_{in}, \hat{t}), \hat{t} = realization(t) \tag{7}$$

2.3 Model summary

In our approach, we separate the effects due to noise and process variations. Process variations effects can be represented by a statistical distribution t that describes the random variations of the transition points in the transfer curve. Furthermore, based on a linear approximation, the random transition points can be optimally modeled by (6) as the sum of the nominal values, μ_t, and a random location, Uc, in the error space. For effects due to noise, we propose a joint probability density function. Summarizing, our entire model consists of three equations,

$$t \sim normal(\mu_t, \Sigma_t) = Uc + \mu_t \tag{8}$$

$$c \sim normal(0, \Sigma_c) \tag{9}$$

$$f(code, V_{in}, \hat{t}), \hat{t} = realization(t) \tag{10}$$

3 Applications in A/D design

The model presented in the previous section contains sufficient information to compute important A/D performances of interest to the circuit designers. Using a fabricated algorithmic A/D[4] as an example, Section 3.1 illustrates parameter extraction techniques for effects due to process variations, followed by algorithms to compute distributions of INL and DNL (Section 3.2), a D/A synthesis example (Section 3.3), an efficient algorithm for computing TSNR (Section 3.4), and a procedure to identify important A/D error sources (Section 3.5).

3.1 Parameter extraction for μ_t and Σ_t

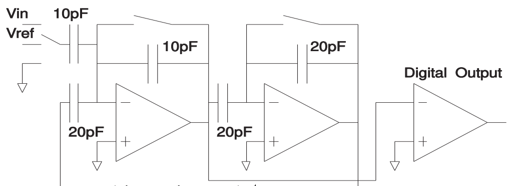

Figure 2: Algorithmic A/D

The algorithmic A/D (Figure 2) is modeled by a network of nonideal component blocks with non-idealities such as comparator and opamp offsets and statistical variations of the sampling and integrating capacitors. The nominal transition points, μ_t, are found by simulation of the A/D using nominal components and searching for each of the transition points using an iterative bisection search algorithm[5]. The sensitivity of the transition points with respect to the process variations, $S_v = \frac{\partial t}{\partial v}$, is found using the finite difference method by perturbating the nominal system with each of the process variations. The covariance matrix of the process variations, Σ_v, is unknown in this example, although in practice the designers should have at least an estimate of Σ_v for proper design. Here, we estimate that the process variations are not correlated, (Σ_v diagonal), with offset voltages $5mV$ and capacitor mismatches 1%. Using Σ_v and S_v in (3) and (5), parameters U and Σ_c are calculated. The diagonal entries of Σ_c represent the relative importance of the corresponding error vectors. In this example, the last three error vectors is expected to be two orders of magnitude less than the dominant first error E_1 shown in Figure 3. Therefore, they are ignored, and the rank r of Σ_t is set to three instead of six.

To verify that the first three errors vectors are sufficient to characterize this A/D, we obtained the measured error, $\hat{t} - \mu_t$, of a fabricated A/D[4], and try to work backwards from the data to the error location described by $U\hat{c}$ using (6). For instance, the realization of (6) with measurement noise is as follows

$$\hat{t} - \mu_t = U\hat{c} + \delta \tag{11}$$

where $\hat{c}$ is an unknown realization of c that minimize the measurement noise $||\delta||^2$. Figure 4 shows $\hat{t} - \mu_t$ matches well with $U\hat{c}$, proving that the model sufficiently captures the A/D behavior. Furthermore, if sufficient data are available, the distribution c can be estimated empirically, so that Σ_t can be extracted from data for comparison against the one predicted.

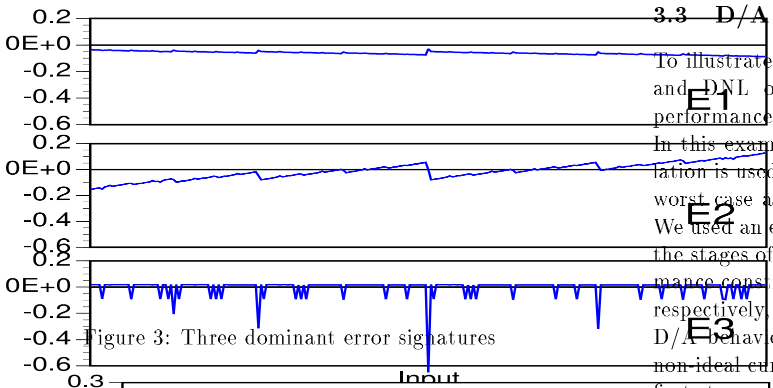

Figure 3: Three dominant error signatures

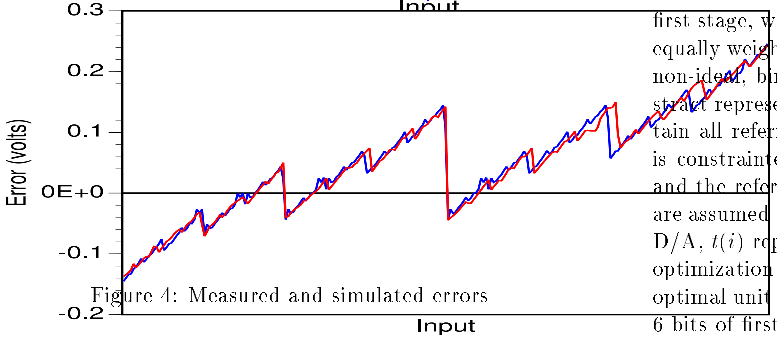

Figure 4: Measured and simulated errors

3.2 Computing INL and DNL

Once the model parameters μ_t and Σ_t are found, INL and DNL are calculated. The INL is a vector whose i^{th} entry represents the deviation of the i^{th} transition point from its ideal location *after* the transfer curve is compensated for gain and offset errors. The compensation is done by linear transformation of the transition points to make the first and last points perfect. (Physically, this is done by scaling the input and adding an offset such that the first and last transition points are perfect.) Suppose L is a vector corresponding to the ideal transition points, the INL has distribution s,

$$s(i) = at(i) + b - L(i) \tag{12}$$

where $a = \frac{L(n)-L(1)}{t(n)-t(1)}$, $b = L(1) - at(1)$, and $n = 2^N - 1$. Therefore, the nominal values of s are given by

$$\mu_s(i) = \frac{L(n) - L(1)}{\mu_t(n) - \mu_t(1)}(\mu_t(i) - \mu_t(1)) + L(1) - L(i) \tag{13}$$

The covariance matrix of s is given by

$$\Sigma_s = (\frac{\partial s}{\partial t})\Sigma_t(\frac{\partial s}{\partial t})^T \tag{14}$$

$$\frac{\partial s}{\partial t}(i,j) = \begin{cases} \frac{(L(n)-L(1))(\mu_t(i)-\mu_t(n))}{(\mu_t(n)-\mu_t(1))^2} & i \neq 1,n \quad j = 1 \\ 0 & i \neq 1,n \quad j \neq 1,i,n \\ \frac{L(n)-L(1)}{\mu_t(n)-\mu_t(1)} & i \neq 1,n \quad j = i \\ 0 & i = 1,n \quad j = 1\ldots n \\ \frac{(L(n)-L(1))(\mu_t(1)-\mu_t(i))}{(\mu_t(n)-\mu_t(1))^2} & i \neq 1,n \quad j = n \end{cases} \tag{15}$$

The DNL is simply the first order difference of the INL, so its distribution d is given by $d = D_+ s$, where D_+ is the first order difference operator. The DNL's nominal value and covariance matrix are given by μ_d and Σ_d, respectively,

$$\mu_d = D_+ \mu_s \tag{16}$$

$$\Sigma_d = D_+ \Sigma_s D_+^T \tag{17}$$

3.3 D/A synthesis

To illustrate the efficiency of this approach to computing INL and DNL of a high resolution D/A, a simple example of performance-driven hierarchical synthesis[6] is briefly described. In this example, numerical optimization with behavioral simulation is used for high level architecture selection, as opposed to worst case analytical analysis in [7] or ruled based systems[8]. We used an existing optimizer to find the optimal resolutions of the stages of a two stage, interpolative 10 bit D/A for a performance constraint of $\pm 3\sigma$ INL and DNL less than 2 and 0.5 lsb, respectively, where lsb refers to the smallest output step. The D/A behavioral model consists of two non-ideal stages and a non-ideal current mirror connecting the first to the second. The first stage, with N_1 bits of resolution, consists of 2^{N_1} non-ideal, equally weighted transistors, and the second stage consists of N_2 non-ideal, binary weighted transistors. The transistors are abstract representations of the real current sources, and they contain all referred mismatches. The transistor dimension, W/L, is constrained to be larger than four for speed requirements, and the referred mismatches, $\sigma_L = 0.02\mu m$ and $\sigma_W = 0.02\mu m$, are assumed to be independent for all transistors. Note that for D/A, $t(i)$ represents the analog output for input code i. From optimization runs using a DEC 5000, the total active area and optimal unit transistor sizes are shown in Figure 5 for 4, 5, and 6 bits of first stage resolutions. The second design was chosen because it has the smallest transistor active area. Also, it can be shown by Monte Carlo integration that nearly all of the realizations of its INL and DNL will fall within the $\pm 3\sigma$ INL, DNL bounds shown on Figure 6.

N1	W1 μm	L1 μm	W2 μm	L2 μm	WM μm	LM μm	Area mm^2	CPU hr
6	63.6	15.9	11.1	2.77	18.3	4.60	6.54	21.0
5	63.8	15.9	17.4	4.36	34.6	8.65	3.56	4.1
4	63.8	15.9	34.1	8.53	80.8	22.2	3.81	1.3

Figure 5: Optimization results

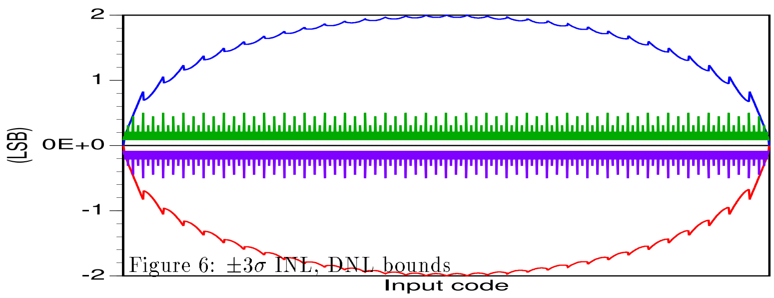

Figure 6: $\pm 3\sigma$ INL, DNL bounds

3.4 Computing TSNR

After the joint probability density function of an A/D is extracted, the signal-to-noise plus distortion ratio (TSNR) for a system consisting of the A/D followed by an ideal D/A can be computed directly *without* using time domain simulation or spectral estimation. Let x be the input and $g(x)$ be its distribution function. Let $A(\cdot, \hat{t})$ be the A/D transfer function, where the gain and offset errors of $\hat{t}$ has been corrected using a linear transformation with parameters a and b in (12). Let $D(\cdot)$ be the ideal D/A transfer function. The TSNR for an input x to an A/D instance $A(\cdot, \hat{t})$ is given by

$$TSNR = \frac{E[x^2]}{E[error^2]} = \frac{\int_{-\infty}^{\infty} x^2 g(x) dx}{\int_{-\infty}^{\infty} \sum_{c=0}^{c=n} (x - D(c))^2 f(c, x, \hat{t}) g(x) dx} \tag{18}$$

3.5 Identification of A/D error sources

Another application for the model is to enable A/D circuit designers to more easily visualize and discover the sources of important errors. Conventionally, designers would vary each of what they thought were the important process variations *individually* and observe the effect on INL and DNL using analytic equations or behavioral simulations. Then, they would get a feeling for the relationships, and try to control the important variations in the design. In contrast, our model parameter extraction technique varies all process variations and extracts Σ_t that captures all relationships simultaneously. Furthermore, the most important process variations can be identified automatically using the following algorithm: (i) Find dominant error vector from Σ_t. (ii) Cross-correlate the dominant error vector with the error vector of each of the process variations. The cross-correlation result will be numbers, each of which quantify the corresponding process variation's contribution to the dominant error. (iii) Rank the process variations according to the magnitude of their number. (iv) The process variations that rank the highest will be relatively more important and should be reported to the designer.

Because our model considers all process variations simultaneously, it also helps a designer produce a better design than the conventional manual design methodology. Traditionally, designers consider process parameters individually, and they design circuit components from the bottom-up to meet the performance requirements due to each process variations individually. Often, the components are over-designed to guarantee a functional system design. In a top-down, global optimization approach using our model, the INL and DNL requirements can be spread more evenly over less critical circuit components. As a result, a better design can be achieved.

4 Applications in system design

The A/D model presented is computationally efficient for performance prediction of systems containing an A/D as a block because the A/D non-idealities are all summarized statistically in the model. For example, section 3.4 shows an algorithm to compute TSNR of an A/D without using time domain simulation or spectral estimation. In this section, we present the advantages of using our model in two approaches to time domain system simulations, namely, worst case and Monte Carlo approaches.

In order to verify the worst case system performance due to non-idealities in an A/D block with the traditional model in (1), we must perform 2^m system simulations since there are m variates in the process variations v, and we need to check every corner of an m-dimensional box. On the other hand, there are only r variates in c, so we can verify the system using 2^r simulations. With $r \ll m$ in general, considerable simulation time is saved.

In the Monte Carlo approach, random instances of A/D are generated and put into the system simulation. In our model, the smaller number of variates in c will speed the generation of A/D instances. Also, since the variates in c are statistically independent, the generation of A/D instances is much easier than the case when the variates are correlated. In general, the variates in v will be correlated, so A/D instance generation is longer if the model in (1) is used.

5 Conclusion

We have presented a behavioral representation for the class of Nyquist rate A/D converters that captures the nominal behavior, as well as all the statistical variations. Also presented were applications of the model such as identification of important A/D error sources, efficient computation of the distributions of INL and DNL, TSNR, and efficient worst case and Monte Carlo system simulations. Finally, parameter extraction results were presented that agree well with actual measurements.

6 Acknowledgements

The authors acknowledge the many helpful discussions by the members of CAD and IC groups at U. C. Berkeley. This project is supported by the Semiconductor Research Corporation. Its support is gratefully acknowledged.

References

[1] Ruan, Genhong "A Behavioral Model of A/D Converters Using a Mixed-Mode Simulator", *IEEE Journal of Solid-State Circuits*, Vol. 26, No. 3, March 1991

[2] E. Liu, et al. "A Behavioral Representation for Nyquist rate A/D Converters", *Proc. IEEE ICCAD* Nov. 1991.

[3] T. M. Souders, G. N. Stenbakken, "Cutting the high cost of testing", *IEEE Spectrum*, March 1991.

[4] R. McCharles, "Charge Circuits for Analog LSI", Ph.D thesis, U. C. Berkeley, 1980.

[5] W. Press, et al. *Numerical Recipes in C*, Cambridge University Press, Cambridge 1988.

[6] H. Chang, et al. "A top-down, Constraint-Driven Design Methodology for Analog Integrated Circuits," *Proc. IEEE CICC*, May 1992

[7] P. Allen and P. Barton "A silicon compiler for successive approximation A/D and D/A converters", *Proc. IEEE CICC* 1986.

[8] L. R. Carley, et al. "ACACIA: The CMU Analog Design System", *Proc. IEEE CICC* 1989.