Behavioral Simulation for Noise in Mixed-Mode Sampled-Data Systems*

Edward W. Y. Liu   Alberto L. Sangiovanni-Vincentelli

Department of EECS, University of California, Berkeley, CA 94720

Abstract

In this paper a "direct" noise analysis approach for mixed-mode systems is presented with experimental results compared with results from the traditional Monte Carlo approach. The direct approach computes noise effects by performing arithmetic on moments of distribution functions that characterize electronic noise. One key advantage of this approach is its ability to compute low error probabilities. From experimental results, it is shown that very low order moments, such as second order, are sufficient for a good estimate of noise effects.

1 Introduction

The simulation of mixed analog and digital systems is crucial in system verification as more analog and digital circuits are integrated on the same integrated circuit in data acquisition, automotive, or disk drive electronics applications. For these systems, there is a trend to reduce system power consumption by reducing the supply voltage or minimizing parasitic capacitances. One of the fundamental limit to these power reduction techniques is the electronic noise inherent in the mixed analog/digital subsystems. While reducing the supply voltage reduces the allowable signal power, the noise power due to physical effects remain the same. Due to the potential decrease in signal-to-noise ratio, it is desirable to simulate the mixed-mode system for noise performances. Unfortunately, there are no noise simulators that can analyze noise effects for mixed-mode systems.

To develop such a simulator, we follow the **behavioral simulation** paradigm in which circuits are modeled mathematically as in [1] [2][3][4]. In this paper, we present a new **noise behavioral model** and a **direct noise analysis** approach for mixed-mode systems. Using the appropriate model, there is no need for circuit or macromodel simulation, but a **direct algebraic** approach where the objects being manipulated are noise characterizations.

In Section 2, the problem is defined. In Section 3, the traditional Monte Carlo method for noise simulation is described. Then the new behavioral models and new approach for noise analysis are presented (Section 4) with experimental results (Section 5).

2 Problem definition

The class of circuits under consideration includes sampled-data systems that have mixed analog/digital inputs and outputs. This includes, but is not limited to, A/D converters, D/A converters, and receivers of digital signals. The objective of the simulation is to determine the noise effects due to electronic noise. Thermal and flicker noise in the transistors and thermal noise in resistors contribute to the noise of the circuit. Traditionally, electronic noise is modeled with a Gaussian random process described by its power spectral density[5]. From *the noise sources*, *the system architecture*, and *the deterministic input*, we seek the distribution of the output. For example, we seek the continuous output distribution of a D/A converter, the bit error rate of a receiver, or the output code distribution of an A/D converter. To illustrate the problem, Figure 1 shows a switched capacitor implementation of a cyclic A/D converter. Due to noise effects, the output of the comparator is wrong with a non-zero probability. Since the N-bit digital output code is a logic function of N successive outputs of the comparator, we seek the probabilities for each of the 2^N possible output codes given a deterministic input.

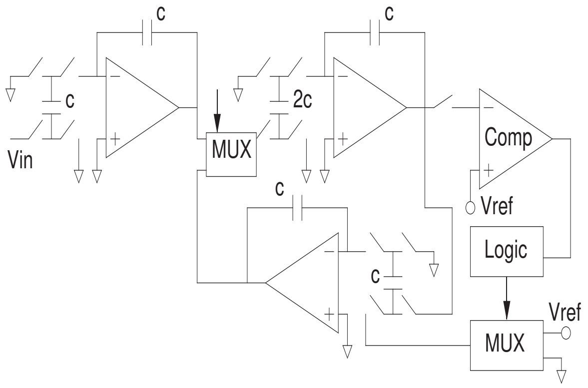

Figure 1: Cyclic A/D

3 Previous work

Traditionally, analog circuit noise is analyzed with the SPICE simulator[6]. The circuit is linearized at the operating point, and the noise is computed by solving the AC equivalent circuit, where the noise is represented by a sinusoidal source in parallel to the noisy elements. The problem with using this approach for mixed-mode systems are two-folded. First, in mixed-mode systems, the comparator has a discontinuous transfer function and cannot be linearized. Secondly, mixed-mode systems are usually non-steady state (e.g. an A/D produces an output code after a *finite* time), so an operating point does not exist for the circuit. Simulators such as [7] can analyze noise in switched-capacitor circuits using linear systems theory, but unfortunately, it cannot handle noise in mixed-mode switched-capacitor circuits because mixed-mode circuits are highly nonlinear.

Without simulators for noise effects, designers resort to the Monte Carlo noise simulation technique. In this technique, they model the circuit as a deterministic network of components, and inject random numbers into the signal path. They run the system many times, and hope to get a wrong output code by chance. To illustrate, a major component in a sampled-data system is the

*This project is supported by the Semiconductor Research Corporation under grant 92-DC-008. Its support is gratefully acknowledged.

sample-and-hold whose noise effects can be modeled as follows. The electronic noise source v in the simplified sample-and-hold circuit in Figure 2(a) has **power spectral density**, $S_v(f)$, where f is frequency. It can be modeled by an additive noise, n, in a sampled-data event-driven simulation as shown in Figure 2(b), where samples of n are generated from a random number generator and added explicitly into the deterministic system simulation. The

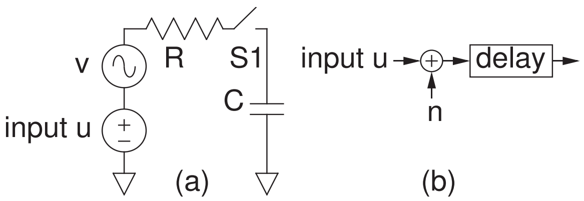

Figure 2: Sample-and-hold

distribution of n is computed from the noise power spectral density, $S_v(f)$. Due to the lowpass filter formed by R and C, the voltage on the capacitor when the switch, S1, is closed has power spectral density given by

$$\frac{S_v(f)}{1+(2\pi fRC)^2} \tag{1}$$

As S1 switches, thus sampling the voltage onto the capacitor, the equivalent **sampled noise power spectral density** on the capacitor is then given by the sampling theorem as

$$S_d(f) = \sum_{k=-\infty}^{\infty} \frac{S_v(f-kf_s)}{1+(2\pi(f-kf_s)RC)^2} \tag{2}$$

where f_s is the switching frequency. The inverse Fourier transform of $S_d(f)$ gives the time domain discrete autocovariance function $c_d(t)$ from which the noise n is defined.

$$n \sim normal(0, \Sigma_n) \tag{3}$$

where $\Sigma_n(i,j) = c_d(iT - jT)$, n is a random vector with m components, mT is the total duration of the simulation, and T is the sampling period. Each component in random vector n is $n(i)$ which represents a noise sample at time iT. From a random number generator, realizations of noise sample $n(i)$ are generated and injected into the simulation at time iT.

Despite the simplicity of the Monte Carlo approach, it has problems with computing low error probabilities and machine dependency. For a system with error probability p_e, the number of system runs required for a confident statistical error estimate is approximately $\frac{1}{p_e}$. In a receiver where the required bit error rate is below 10^{-8}, 100 million or more samples are needed. Furthermore, pseudo-random number generators on computers often do not generate such a large sequence of independent random numbers, but will re-use old random numbers instead. In such instances, the extra simulations do not yield a more accurate statistical estimate.

4 New approach

4.1 Signal Representation

One of the key disadvantage of the Monte Carlo technique is the representation of noise by computer generated samples of noise because many noise samples are required for good accuracy. In contrast, in our approach we represent the noise by a **random signal**. For example, each noise sample $n(i)$ in (3) is a random variable obtained from sampling the random signal n at time iT. A **random variable**, X, is described by its distribution function f_X, or alternatively by all of its **moments**, where the k^{th} moment x_k is defined as

$$x_k = E[X^k] = \int_{-\infty}^{\infty} u^k f_X(u)du \tag{4}$$

For example, $x_0 = 1$ by definition since the area under the distribution curve is unity. In our system model, we will approximate a random variable by its first $k+1$ moments from x_0 to x_k, where k is an integer selected by the user based on accuracy requirements. The moments are parameters to characterize a distribution function. If all moments are used, then the distribution function is completely characterized. The advantage of using a finite number of lower order moments is computational efficiency. Instead of manipulating a distribution function, we manipulate a set of numbers representing the moments. Furthermore, from the Central Limit Theorem, random variables resulting from many noise sources will converge to a Gaussian distribution which can be characterized by the first two moments, x_1 and x_2. As a result, low order moments are often the principal components of a random variable in practical circuits. The disadvantage of using a finite number of moments is that effects of higher order moments are not considered.

4.1.1 Assumptions

Although the moments describe the distribution of a random variable, they do not provide information about the correlation between each random variable. In general, noise in a system are correlated because the noise sources themselves may be correlated or the system may contain architectures with **reconvergent fan-out**. As a first step, we will focus on systems with noise that are *independent* for the following reasons.

- Noise sources from physically different components should be independent of each other.
- In most sampled-data systems, white noise dominates non-white noise. Thus, successive samples in time of a noise source are approximately independent.
- We focus on systems such as converters and finite impulse response filters that lack reconvergent fan-out.

4.2 System simulation algorithm

After describing the random signal representation, we will focus in this section on the algorithm used to simulate the noise in sampled-data systems. The algorithm is event-driven as described in detail below, yet it must handle noise represented as random signals in mixed-mode, sampled-data systems. In such systems, the comparator is a special interface since its input is analog while its output is digital (e.g. H or L). Due to noise, the analog signals are random; therefore, both comparator output states are possible. The technique used in our algorithm is to compute the probability of each possible outcome, and continue the simulation recursively for each possible **scenarios**. Schematically, the simulation follows a tree of scenarios based on comparator outputs as shown in Figure 3. To compute the probability of a particular scenario occuring, we take the product of the probabilities on the unique path from the tree root to the scenario. The simulation algorithm is as follows.

4.2.1 Simulation algorithm

- (a) Extract noise parameters from circuits using (1) and (2). Construct moments of distributions for noise sources.
- (b) Topological sort network to find the evaluation order of components. The procedure always succeed because all feedback paths are broken by at least one delay in sampled-data systems.
- (c) Evaluate each component in order, return if end of list.
- (d) When evaluating a comparator, check distribution at comparator input X, find probabilities for output Y = H and L, execute recursively (e) and (f)
- (e) Assume the scenario with output Y=H, find conditional distribution of X | Y=H, replace X with X | Y=H, continue simulation in (c)
- (f) Assume the scenario with output Y=L, find conditional distribution of X | Y=L, replace X with X | Y=L, continue simulation in (c)

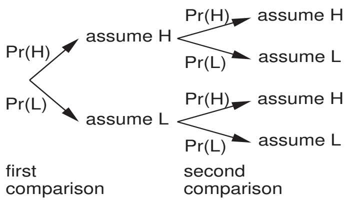

Figure 3: Scenario tree

4.3 System component modeling

As the analog signals are generalized from a deterministic representation to a random representation, the system components must be generalized to handle such random signals as well. Figure 4 shows the cyclic converter example modeled with the new

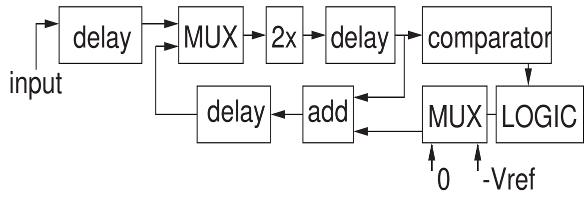

Figure 4: New model for Cyclic A/D

components such as delays, multiplexers, multipliers, adders, and comparators. The delay and multiplexer components are trivially generalized to process random signals since the former copies the input to the output, and the latter routes one of many inputs to the output. The rest of the components will be defined in subsequent sections.

4.3.1 Adder

The adder sums two random variables X and Y to produce an output Z. To find the moments of Z from the moments of X and Y we use

$$z_k = E[Z^k] = E[(X+Y)^k] = \sum_{j=0}^{k} \left(\begin{array}{c} k \\ j \end{array} \right) E\left[X^{k-j}Y^j\right] \quad (5)$$

using the binomial expansion of $(X+Y)^k$. From the independence of X and Y assumed in section 4.1.1, we then have

$$z_k = \sum_{j=0}^{k} \left(\begin{array}{c} k \\ j \end{array} \right) EX^{k-j}EY^j = \sum_{j=0}^{k} \left(\begin{array}{c} k \\ j \end{array} \right) x_{k-j}y_j \quad (6)$$

4.3.2 Multiplier

The multiplier multiplies two random variables X and Y to produce an output Z. To find the moments of Z from the moments of X and Y we have

$$z_k = E[(XY)^k] = E[X^kY^k] = E[X^k]E[Y^k] = x_k y_k \quad (7)$$

using the independence of X and Y.

4.3.3 Comparator

The comparator has as inputs a random variable X and a threshold value t, and produces the probability p_H for output high, the probability p_L for output low, the conditional distribution of X given the discrete output Y being low $(f_{X|Y}(x|L))$, and the conditional distribution of X given Y being high $(f_{X|Y}(x|H))$. Mathematically, the probability for output high is defined as

$$p_H = \int_t^{\infty} f_X(u)du \quad (8)$$

where f_X is constructed from the given moments x_n (Appendix A). Conversely, the probability for output low is defined as

$$p_L = \int_{-\infty}^{t} f_X(u)du = 1 - p_H \quad (9)$$

In [8], it is shown that the conditional distribution of X given that $Y = y$ is

$$f_{X|Y}(x|y) = \frac{P\{Y = y|X = x\}}{P\{Y = y\}} f_X(x) \quad (10)$$

Also, we have $P\{Y = H\} = p_H$ and

$$P\{Y = H|X = x\} = \left\{ \begin{array}{ll} 1 & x \geq t \\ 0 & x < t \end{array} \right.$$

Applying (10) in our comparator, we find the conditional distribution of X given $Y = H$ is

$$f_{X|Y}(x|H) = \left\{ \begin{array}{cc} \frac{f_X(x)}{p_H} & x \geq t \\ 0 & x < t \end{array} \right. \quad (11)$$

Conversely, the conditional distribution of X given $Y = L$ is

$$f_{X|Y}(x|L) = \left\{ \begin{array}{cc} 0 & x \geq t \\ \frac{f_X(x)}{p_L} & x < t \end{array} \right. \quad (12)$$

Once the conditional distributions are determined, they are converted to a set of $k + 1$ moments (Appendix A).

5 Experimental results

5.1 Cyclic A/D

A 12 bit cyclic A/D (Figure 1(a)) has been modeled in the C++ language for both Monte Carlo simulations and direct noise calculations. The gain parameter of a sample-and-hold can be computed from a circuit level implementation using SPICE transient analysis given the fixed system clock period. In this experiment, we assumed the gain to be one for $1x$ S/H and two for $2x$ S/H. The comparator threshold can be computed using SPICE also, but we assumed it to be $1V$. Input voltage is half of the full range input. The effective noise of a sample-and-hold can be computed using (2) and (3). However, for this experiment we assume that the effective noise is given and is white with standard deviation of $0.365mV$, which may be considered high. However, a high noise value is used to prevent excessive CPU time in Monte Carlo simulations, while a lower noise value would not affect our method.

Figure 5 compares the predicted output code distribution from a reference Monte Carlo simulation (10^7 trials) with the output code distributions from direct noise calculations ($k = 2$) and ($k = 3$), where k is the highest moment of interest. Notice that the results are comparable, but the CPU times are much less (9620s vs. 0.1s and 3.7s). Next, in Figure 6 we compare the Monte Carlo method with the direct method for accuracy as a function of CPU time. The number of trials in Monte Carlo simulations is varied from one hundred to seven hundred thousand. Assuming the reference simulation gives the true solution, the absolute errors in the probabilities are computed for all output codes. The average of the absolute error is then plotted against CPU time. The error curve for the Monte Carlo method indicates that its accuracy is inversely proportional to CPU time. The error curve for the direct method shows that error decreases with higher order approximation. Besides, the direct method has smaller error than the Monte Carlo method for comparable CPU times.

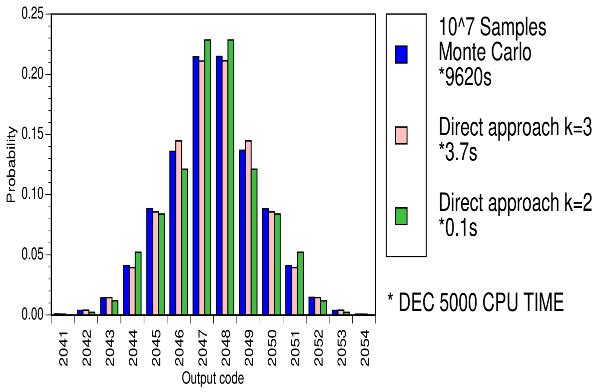

Figure 5: Twelve-bit cyclic A/D code distribution

5.2 High speed flash A/D

Noise in flash A/D cause errors in comparator outputs, leading to errors in digital decoding. If a simple digital decoder is used, the output code can be very different from the correct code, in which case the error is called a **sparkle**[9]. Because the output error probabilities are usually very small, ($6.4 \cdot 10^{-4}$ in this example), many simulations are needed in the Monte Carlo method to predict sparkles.

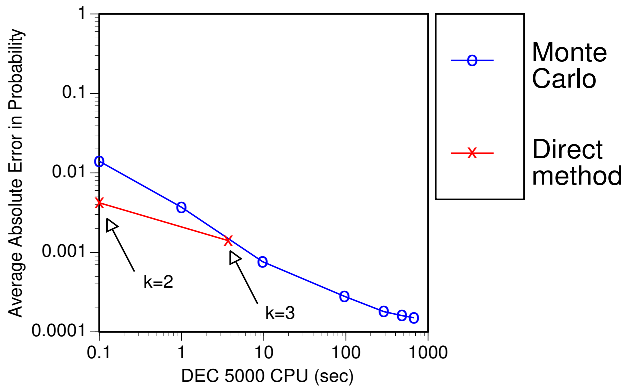

Figure 6: Error vs. CPU time

In contrast, using the direct noise computation method, we have modeled two flash A/D's, one with a simple thermometer-to-binary ROM decoder and one with a more advanced decoder. Effective comparator noise can be estimated from [10]. In this experiment, for convenience we set the standard deviation of the comparator noise to a quarter of the smallest input voltage step size. Input voltage is half of the full range input. Output code distribution, computed using direct method (k=2), is shown in Figure 7.

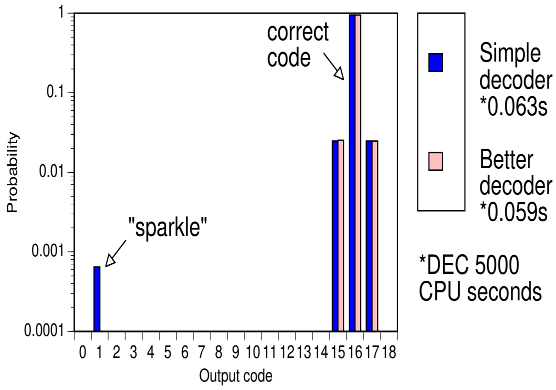

Figure 7: Five-bit flash A/D code distribution

6 Conclusion and future work

We have presented a "direct" noise analysis approach for mixed mode systems, and compared our approach with the traditional Monte Carlo approach. The direct approach computes noise effects by performing arithmetic on moments of distribution functions that characterize electronic noise. One key advantage of this approach is its ability to compute low error probabilities. From experimental results, we have shown that very low order moments, such as second order, are sufficient for a good estimate of noise effects.

In the future, we plan to implement higher order comparator models to handle higher order moments, e.g. fourth and fifth, for better accuracy, as well as investigate extension of the technique to correlated noise using **joint moments**.

A Algorithms for comparators

In this appendix, we will show the algorithms to compute from input moments of X and the comparator threshold t the probabilities of the comparator output Y=H and Y=L, as well as the moments for the conditional distributions $f_{X|Y}(x|H)$ and $f_{X|Y}(x|L)$. We will show the cases for $k = 2, 3$, where k is the highest moment of interest.

A.1 Highest moment k=2

When $k = 2$, the random variable X is approximated by a Gaussian random variable. The mean, μ, is given by x_1, and the variance σ^2 is given by $x_2 - x_1^2$. With these parameters, the distribution of X is

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{13}$$

The probability for $Y = H$ is given by integrating the distribution from the trigger point t to infinity,

$$p_H = \int_t^\infty f_X(x) = 1 - Q\left(\frac{t-\mu}{\sigma}\right) \tag{14}$$

where $Q(\cdot)$ is a standard mathematical function defined as

$$Q(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} du \tag{15}$$

Letting $A \sim f_{X|Y}(x|H)$, and using (11), we have

$$f_A(x) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma p_H} e^{-\frac{(x-\mu)^2}{2\sigma^2}} & x \geq t \\ 0 & x < t \end{cases} \tag{16}$$

The moments of A are then given by

$$a_1 = \int_{-\infty}^{\infty} x f_A(x) dx = \frac{\sigma}{\sqrt{2\pi} p_H} e^{-\frac{(t-\mu)^2}{2\sigma^2}} + \mu \tag{17}$$

$$a_2 = \int_{-\infty}^{\infty} x^2 f_A(x) dx = \frac{\sigma}{\sqrt{2\pi} p_H}(t-\mu) e^{-\frac{(t-\mu)^2}{2\sigma^2}} + \sigma^2 + 2\mu a_1 + \mu^2 \tag{18}$$

The probability for $Y = L$ is then given by $p_L = 1 - p_H$. Letting $B \sim f_{X|Y}(x|L)$, the moments of B are given by

$$b_1 = (x_1 - a_1 p_H)/p_L \tag{19}$$

$$b_2 = (x_2 - a_2 p_H)/p_L \tag{20}$$

A.2 Highest moment k=3

When $k = 3$, the random variable, X, is approximated by distribution[11] created by joining together two halves of two different Gaussian random variables defined as

$$f_X(x) = \begin{cases} \frac{2}{\sqrt{2\pi}(\sigma_1+\sigma_2)} e^{-\frac{(x-a)^2}{2\sigma_1^2}} & x \geq a \\ \frac{2}{\sqrt{2\pi}(\sigma_1+\sigma_2)} e^{-\frac{(x-a)^2}{2\sigma_2^2}} & x < a \end{cases} \tag{21}$$

where its moments are given by

$$x_1 = \left(\sqrt{\frac{2}{\pi}}\sigma_1 + a\right)\frac{\sigma_1}{\sigma_1+\sigma_2} + \left(a - \sqrt{\frac{2}{\pi}}\sigma_2\right)\frac{\sigma_2}{\sigma_1+\sigma_2} \tag{22}$$

$$x_2 = \left(\sigma_1^2 + 2\sqrt{\frac{2}{\pi}}\sigma_1 a + a^2\right)\frac{\sigma_1}{\sigma_1+\sigma_2} + \left(\sigma_2^2 - 2\sqrt{\frac{2}{\pi}}\sigma_2 a + a^2\right)\frac{\sigma_2}{\sigma_1+\sigma_2} \tag{23}$$

$$x_3 = \left(2\sqrt{\frac{2}{\pi}}\sigma_1^3 + 3\sigma_1^2 a + 3\sqrt{\frac{2}{\pi}}\sigma_1 a^2 + a^3\right)\frac{\sigma_1}{\sigma_1+\sigma_2} + \left(-2\sqrt{\frac{2}{\pi}}\sigma_2^3 + 3\sigma_2^2 a - 3\sqrt{\frac{2}{\pi}}\sigma_2 a^2 + a^3\right)\frac{\sigma_2}{\sigma_1+\sigma_2} \tag{24}$$

A closed form solution for σ_1, σ_2, a in terms of x_1, x_2, x_3 is intractable; therefore, parameters σ_1, σ_2, a are computed from x_1, x_2, x_3 using the downhill simplex optimization method in multidimensions[12] to best match the moments. The cost function being used is $c = \sum_{n=1}^{k}(x_k - x'_k)^2$, where x_k are the moments given, and x'_k are the resulting moments given by (22) to (24).

Let $A \sim f_{X|Y}(x|H)$, then using (11) we have

$$a_1 = \sqrt{\frac{2}{\pi}}\frac{\sigma_1}{\sigma_1+\sigma_2}\frac{\sigma_1}{p_H} e^{-\frac{(t-a)^2}{2\sigma_1^2}} + a \tag{25}$$

$$a_2 = \sqrt{\frac{2}{\pi}}\frac{\sigma_1}{\sigma_1+\sigma_2}\frac{\sigma_1}{p_H}(t-a) e^{-\frac{(t-a)^2}{2\sigma_1^2}} + \sigma_1^2 + 2a_1 a + a^2 \tag{26}$$

$$a_3 = 2\sqrt{\frac{2}{\pi}}\frac{\sigma_1}{\sigma_1+\sigma_2}\frac{\sigma_1^3}{p_H}\left(1 + \frac{(t-a)^2}{2\sigma_1^2}\right) e^{-\frac{(t-a)^2}{2\sigma_1^2}} + 3a_2 a + 3a_1 a^2 + a^3 \tag{27}$$

References

[1] E. Liu, A. Sangiovanni-Vincentelli, G. Gielen, and P. Gray "A Behavioral Representation for Nyquist Rate A/D Converters", *Proc. IEEE ICCAD*, November 1991.

[2] G. Gielen, E. Liu, A. Sangiovanni-Vincentelli, and P. Gray "Analog Behavioral Models for Simulation and Synthesis of Mixed-Signal Systems", *Proc. European Design Automation Conference*, March 1992.

[3] E. Liu, G. Gielen, A. Sangiovanni-Vincentelli, and P. Gray "Behavioral Modeling and Simulation of Data Converters", *Proc. IEEE International Symposium on Circuits and Systems*, May 1992.

[4] E. Liu and A. Sangiovanni-Vincentelli, "Behavorial Representation for VCO and Detectors in Phase-Lock Systems", *Proc. Custom Integrated Circuits Conference*, May 1992.

[5] P. R. Gray and R. G. Meyer *Analysis and Design of Analog Integrated Circuits*, 2nd edition, p. 637, John Wiley & Sons, 1984

[6] L. Nagel, "SPICE2: A Computer Program to Simulate Semiconductor Circuits," Memo No. ERL-M520, Dept. of EECS, U. C. Berkeley, 1975.

[7] J. Vandewalle, H. De Man, and J. Rabaey "The adjoint switched capacitor network and its application to frequency, noise and sensitivity analysis", *Circuit Theory and Applications*, Vol 9, 77-88 (1981), Wiley, 1981

[8] S. Ross, *A First Course in Probability*, 2nd edition, p. 211. Macmillan Publishing Company, New York, 1984.

[9] Y. Gendai, Y. Komatsu, S. Hirase, M. Kawata "An 8b 500MHz ADC," *Proc. IEEE ISSCC*, pp. 172-174, Feb. 1991.

[10] A. Matsuzawa, S. Nakashima, I. Hidaka, S. Sawada, H. Kodaka, S. Shimada "A 6b 1Ghz Dual-Parallel A/D Converter," *Proc. IEEE ISSCC*, pp. 174-175, Feb. 1991.

[11] S. M. Sze *VLSI Technology*, p. 226, McGraw-Hill Book Company, 1983

[12] W. Press, B. Flannery, S. Teukolsky, W. Vetterling *Numerical Recipes in C*, p. 305, Cambridge University Press, 1989.