# Verification of Delta-Sigma Converters Using Adaptive Regression Modeling *

Jeongjin Roh, Suresh Seshadri and Jacob A. Abraham
Computer Engineering Research Center
The University of Texas at Austin
Austin, TX 78712

## Abstract

A new verification technique for $\Delta\Sigma$ analog-to-digital converters (ADC) is proposed. The ADC is partitioned into functional blocks, and adaptive regression models for each partition are constructed using transistor-level simulation data. Non-idealities in circuit behavior are captured by the adaptive regression technique from the collected data. The algorithms have been implemented in a simulation program ARSIM (Adaptive Regression Simulator), which performs data sampling, model building, and simulation. Experimental results using ARSIM are shown on a second-order $\Delta\Sigma$ modulator, and they demonstrate the effectiveness of our technique as a fast and accurate approach for verifying $\Delta\Sigma$ converters.

## 1 Introduction

Delta-Sigma modulation has become a dominant technique for implementing analog-to-digital conversion (ADC). As compared to other popular techniques like flash and pipelined A/D conversion, the performance of $\Delta\Sigma$ modulators is not dictated by matching of component values, which facilitates the design of high performance systems without precise analog circuitry [1]. Its ability to trade off between speed and resolution, and relaxed requirements on performance and matching of analog components enabled by oversampling and digital filtering, have made $\Delta\Sigma$ converters an attractive choice for silicon VLSI technology that is optimized for digital implementation.

As observed from the simple first-order $\Delta\Sigma$ modulator in Figure 1, the quantized output signal is given by

$$p_i = x_{i-1} + (e_i - e_{i-1}). \quad (1)$$

Thus, the modulator differentiates the quantization error while the signal remains unchanged except for a delay.

While oversampling has advantages in the form of noise shaping, it also presents a bottleneck to the simulation of

*This research was supported in part by the National Science Foundation under grant CCR-9731433, and in part by the Texas Advanced Technology Program/Development and Transfer under project #003658-433.

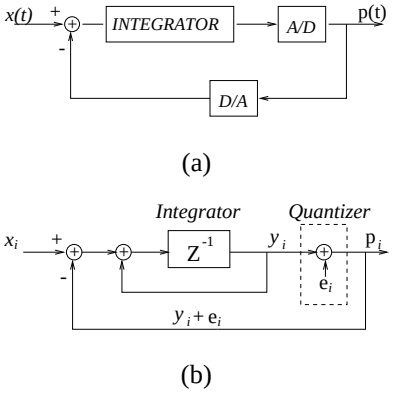


Figure 1: (a) A first-order $\Delta\Sigma$ modulator and (b) its equivalent sampled data representation.

$\Delta\Sigma$ converters. Due to oversampling, thousands of clock cycles are needed for sufficiently simulating these ADCs to calculate their signal-to-noise and distortion ratio (SNDR). With increasing complexity of designs, circuit simulators such as SPICE may take an inordinate amount of time for simulation. This problem not only affects the design cycle time for $\Delta\Sigma$ data converters but also makes their verification a cumbersome task.

A well known technique for modeling and simulating a $\Delta\Sigma$ modulator is using difference equations for the sampled-data operation of a modulator, which is usually implemented in switched-capacitor circuits. As the operation of switched-capacitor modulator generates sampled data for each clock, it can be represented in the discrete-time domain as in Figure 1. Such a technique is well suited for obtaining an ideal model. However, capturing non-idealities of the analog components into the difference equations is a non-trivial task. This limitation renders simple difference equation models inaccurate for simulation purposes.

Another approach for the accurate modeling of analog components in the modulator is to generate analytical behavioral expressions for each circuit non-ideality. In [2], analytical expressions were generated for a second-order modulator, for each circuit non-ideality such as integrator leakage, slew rate, sampling jitter, etc.; then, each expres-

sion was included in the difference equation simulations to find the effects of the modeled non-idealities. However, the overall effect of the non-idealities cannot be calculated by simply summing up individual analytical equations as there may be complex interactions between two or more non-ideal effects. Even though it could be possible to analyze the circuit for simple modulators, as the complexity of the $\Delta\Sigma$ modulator increases, accurate analytical modeling would require expertise and may be prone to errors in analysis.

Today, the $\Delta\Sigma$ converter architecture has become quite complicated with many stages and a reduced power supply voltage to comply with the digital logic in mixed-signal circuits. Increasing demand for low-power designs [7] has lead to low voltage designs. However, to maintain signal integrity in such low-voltage designs it becomes necessary to add more analog components such as voltage doublers for switches which make the analysis even more difficult. As a result, analytical models tend to make some simplifying assumptions, which are unrealistic and may severely affect simulation results.

More efficient modeling and simulation techniques have also been proposed which take into account the sampled-data operation of switched-capacitor $\Delta\Sigma$ modulators [3, 4, 5, 6], which are reviewed in Section 2 in detail.

We are using an adaptive regression technique called Multivariate Adaptive Regression Splines (MARS) [8, 9] for sampled data simulation. MARS has earlier been applied to classify faulty and fault-free circuits to enhance fault coverage [10] and to model a faulty circuit to speed up statistical fault simulation [11]. This paper explores the application of MARS as an accurate performance simulation tool for $\Delta\Sigma$ modulators and develops an efficient tool achieving better accuracy with smaller memory requirements than table-based techniques in [5, 6].

## 2 Sampled Data Simulation and Previous Work

In this section, essential issues involved in analog sampled data simulation are discussed and previous approaches to modeling switched-capacitor $\Delta\Sigma$ modulators are reviewed.

Switched-capacitor circuits have become a popular method of implementing $\Delta\Sigma$ modulators due to their compatibility with CMOS technology and better accuracy. As a result the operation of the modulator needs to be analyzed in the discrete-time domain. Sampled data techniques use the clocked operation of the switched-capacitor circuits to formulate difference equations in the discrete-time domain. The two phase non-overlapping clock, $\phi 1$ and $\phi 2$, shown in Figure 6 serves to separate the sample and integrate operations. After each clock, the state of the circuit settles to certain values as given by the output voltage of each integrator stage. The state at each discrete-time point depends on the circuit non-idealities such as switching resistance, slew rate, finite gain of the op-amp, and clock jitter. The state of the circuit could be settling to a certain value during switching, but the transition state, which may be ringing or slewing is not important and only the final voltage level at the end of that clock cycle will have a bearing on the next state of the circuit. However, this voltage level tends to be different from the ideal value as a result of a cumulative effect of all the non-idealities. Our objective is to model each integrator in the presence of all the non-idealities using sampled data points for different values of input, previous output and quantized output.

The simple difference equation model for an ideal integrator in the discrete-time domain, as shown in Figure 1, can be obtained from the ideal integrator transfer function in Equation 2 and is given by $y_i = y_{i-1} + x_{i-1} - p_{i-1}$ where $y_i$ is the output of the integrator at the end of the $i^{th}$ time step, $x_{i-1}$ and $y_{i-1}$ are input and output of the integrator at the end of $(i-1)^{th}$ time step and $p_{i-1}$ is quantized output.

$$H(Z) = \frac{Z^{-1}}{1 - Z^{-1}} \tag{2}$$

A more accurate model would need to include integrator non-idealities such as finite gain, slew rate and sampling jitter. Some non-idealities such as the integrator leakage due to the finite gain of an op-amp can be included in the ideal model easily by adding a leakage factor, *A*, as given by Equation 3.

$$H(Z) = \frac{Z^{-1}}{1 - A \times Z^{-1}} \tag{3}$$

In the above equation *A* is given by $f_A(\mathcal{P}, \mathcal{N})$ where $\mathcal{P}_i s$ are the circuit parameters and $\mathcal{N}_j s$ are the non-idealities. The DC gain of the integrator is now given by $H_0 = 1/(1 - A)$, instead of being infinity when $A$ is 1 for the ideal model. However, $A$ itself may not be a fixed value for a circuit, and may vary depending on the state of the integrator because of its dependence on operating voltage level, slew rate, clock feedthrough, and other non-idealities of the circuit. Therefore, accurately modeling non-idealities into the ideal transfer function may not be as simple as a simple constant leakage factor. Also, some of the earlier work assume white noise for the quantization error which simplifies the analysis but limits the kind of input one can use to get accurate results. Finally, the complexity of the currently used higher order, low-voltage $\Delta\Sigma$ modulators [7] would make the analytical technique an infeasible approach to the modeling problem.

In [3, 4], it was proposed to use state variable analysis utilizing the sampled data property of the switched-capacitor $\Delta\Sigma$ modulator. Although state equations are useful to describe the characteristics of a circuit, it is a known

fact that forming state variable matrices is a difficult procedure even for small circuits.

Table-based techniques have been proposed in [5] [6]. These techniques can model non-idealities without needing to deal with detailed analytical modeling of individual non-idealities. They represent the integrators by two-dimensional tables with input and previous output of the integrator for each dimension. Since most input/output pairs will not be same as the stored values and would be located between discrete values, an interpolation process is required to calculate the output from the table model. The interpolation routine could be a major factor in simulation time for the table-based technique.

Instead of storing each data point, the proposed technique computes an expression model for the modulator using an adaptive regression technique. The model building process is a one time procedure performed before simulating the model, and the storage overhead is avoided. Also, as explained in next section, the interpolation process is moved into the model building process, which eliminates interpolation calculations at every time point.

## 3 Proposed Technique

The proposed technique models non-idealities using mathematical models built from the circuit-level simulation data. Since our mathematical models are based on circuit-level simulation, our technique does not make any assumptions which would be necessary for analytical techniques.

First, the modulator is partitioned into functional blocks comprised of integrators. Then, each integrator is simulated using a circuit simulator to collect data samples. Thereafter, the adaptive regression technique, which is explained in detail in Section 4.2, is used to extract and model the nonidealities from the data samples. The regression model thus obtained is used, instead of the transistor-level netlist, to simulate the entire modulator using our discrete-time domain simulator, ARSIM.

The actual relation of the integrator is an unknown, nonlinear 4-dimensional function as given by Equation 4 following the notation in Figure 1.

$$y_i = f(y_{i-1}, x_{i-1}, p_{i-1}) \tag{4}$$

Equation 4 models the entire non-ideal function consisting of the ideal signal, $s_i$, and the non-idealities, $n_i$ as shown in Equation 5. While previous approaches model the entire signal, $y_i$, ARSIM models only the non-ideality signal, $n_i$ in Equation 5 and during simulation it is added to the ideal signal obtained from the ideal model. In this paper, the non-ideality function, $n_i$, is defined as the difference between the the ideal output and the HSPICE output, given by $I$ and $f$ respectively in Equation 6.

$$y_i = s_i + n_i \tag{5}$$

$$n_i = I(y_{i-1}, x_{i-1}, p_{i-1}) - f(y_{i-1}, x_{i-1}, p_{i-1}) \tag{6}$$

The lack-of-fit (LOF) criterion in modeling techniques can be quantified using a distance function which may be defined as the squared-error loss given by Equation 7, where $\hat{f_M} = \hat{s_i} + \hat{n_i}$ is the modeling function and $X_i s$ are the predictor variables, such as the input and the previous output values of the integrator in our case.

$$LOF(\hat{f_M}) = [y_i - \hat{f_M}(X_i)]^2 \tag{7}$$

Rewriting Equation 7 using the two signal components defined in Equation 5 we get

$$\begin{aligned} LOF(\hat{f_M}) &= [s_i + n_i - \hat{s_i} - \hat{n_i}]^2 \\ &= [s_i - \hat{s_i} + n_i - \hat{n_i}]^2 \\ &= [e_{1i} + e_{2i}]^2 \end{aligned} \tag{8}$$

The ideal signal value is typically several orders of magnitude stronger than the non-idealities. From our experiments we observe that the non-ideality function value is at least 4 orders of magnitude smaller than the ideal signal value. Hence, if we assume the modeling error for a given technique to be 1% of the signal value, the accuracy of the model will be decided by the modeling error, $e_{1i}$, and the non-ideality signal, $n_i$, would be completely masked by it. This would significantly affect the modeling accuracy.

In ARSIM, $\hat{f_M}$ is the non-ideality function, $\hat{n_i}$, that we are modeling and $LOF(\hat{f_M})$ is given by $[e_{2i}]^2$, the measure of fit that is used in the adaptive regression technique described in Section 4.2.

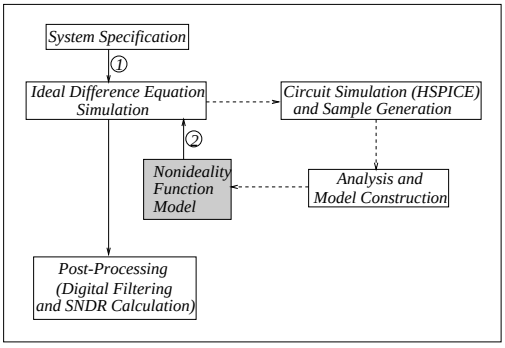


Figure 2: Flowchart for *ARSIM*.

Figure 2 shows the flowchart of the CAD tool we have implemented. The ideal difference equations for each integrator stage are extracted from system specifications. Transistor-level netlists for individual integrators are simulated and samples generated using HSPICE, which are given as input to the model construction routine. Models for the non-ideality function are built using MARS and added to the ideal signal value during simulation. The post-processing stage performs digital filtering and computes the SNDR curve.

## 4 Regression-Based Modeling

This section reviews the basic concept of regression and introduces MARS as a modeling technique for $\Delta\Sigma$ modulators. Regression is a statistical technique for predicting values of one or more responses from a collection of predictor variable values. Use of well-developed statistical techniques like adaptive regression for circuit modeling can overcome some of the drawbacks associated with table-based techniques. This is particulary true of the computational overhead involved in interpolating table values to obtain the required output values at every time point in the simulation.

### 4.1 Regression Technique

The problem of adequately approximating a function of several variables is encountered in many different fields of study. With an increasing number of variables, the number of data points required to obtain a reasonable approximation increases exponentially, and is termed the *curse of dimensionality*. Prior analysis of the circuit can reduce the number of data points to linear complexity in the number of variables. However, the basic idea of our approach is to avoid such a time consuming analysis of the circuit non-idealities and hence, our regression technique works without any information about the behavior of the circuit. Moreover, we are trying to model the transient response of the modulator in the presence of non-idealities, which could be highly non-linear. Parametric regression techniques, which assume the entire data to be one single continuous function and try to find the best fit for the parameters, may result in very inaccurate approximations even with large number of data points for our application. MARS on the other hand divides the input range into several regions and tries to find best fits using splines for each region in the input space. This results in two advantages over parametric regression; 1) it is well suited for high-dimensional problems as the number of data points needed is much smaller, and 2) it can handle highly non-linear responses.

Our goal is to model the dependence of the output $y$ on one or more variables $x_1, x_2, ..., x_n$, given measurement, or simulation data $\{y_i, x_{1i}, x_{2i}, ..., x_{ni}\}$. This relationship can be expressed as in Equation 9. The $\varepsilon$ is an error component of the function $f(x_1, ..., x_n)$, and the expected value $E[\varepsilon]$ is 0.

$$y = f(x_1, ..., x_n) + \varepsilon \tag{9}$$

For modeling the integrator, $y$ is the output of the integrator, and $x_i s$ are the input, previous outputs and quantized output($V_{dd}$ or $V_{ss}$) values. We can represent the output of an integrator for a $\Delta\Sigma$ modulator as in Equation 10. Then, this regression equation can be the accurate model of the integrator function in Equation 4.

$$y_i = f(y_{i-1}, x_{i-1}, p_{i-1}) \tag{10}$$

Data points needed for model generation are obtained from transistor-level simulations of the individual integrator stages. The data generation procedure is described in Section 3.

### 4.2 MARS Modeling

As seen in the previous section, MARS can be an accurate modeling technique for transient simulation of both linear and non-linear circuits efficiently capturing non-idealities in circuit behavior.

$$f(\mathbf{x}) = \sum_{i=1}^{m} w_i g_i(\mathbf{x}) + w_0 \tag{11}$$

Equation 11 shows a MARS function, where $w_i s$ are the parameters of the approximating functions, $g_i(\mathbf{x})\mathbf{s}$ are basis functions, and $\mathbf{x}$ is the vector of input variables. In the above MARS model, $f(\mathbf{x})$ is the same as the $f(\mathbf{x})$ in Equation 10 and $\mathbf{x}$ corresponds to simulation inputs and previous outputs as explained earlier.

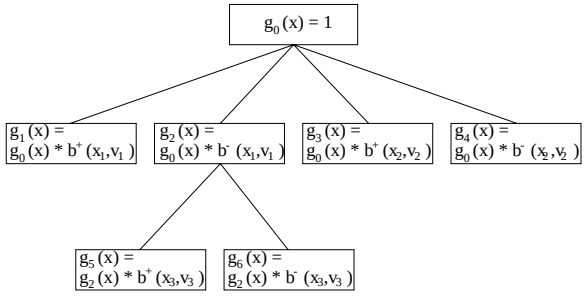


Figure 3: Example of a MARS tree.

Figure 3 shows an example of a MARS basis function tree [9], which is built by recursive partitioning and function representation based on splines. A spline is a series of locally defined low order polynomials that is used to approximate data. The local splines have their own end points called knots. Knots are used to divide the nonlinear function into several relatively linear parts so that they can be modeled with less effort. The basic function of the MARS model is a left/right pair of univariate basis functions $\mathbf{b}^+$, $\mathbf{b}^-$ with a particular knot location $v$ for a particular input variable $x$.

In Figure 3, each node represents a product of these univariate basis functions. Using a greedy optimization algorithm, two child nodes are created by taking the product of each of the univariate basis function pairs with the same parent basis function. Equation 12 and 13 show an example of two child nodes where $v_j$ is a knot location for an input variable $x_k$.

$$g_{children^+}(\mathbf{x}) = g_{parent}(\mathbf{x}) * b^+(x_k, v_j) \tag{12}$$

$$g_{children^-}(\mathbf{x}) = g_{parent}(\mathbf{x}) * b^-(x_k, v_j) \tag{13}$$

Generalized Cross Validation (GCV) [8] is used as a measurement of fit to determine approximating parameters $w_i$ for the basis function in Equation 11, and knot locations $v_j$ in Equation 12 and 13.

## 5 Modeling and Validation

In order to validate our modeling technique, we generated random samples and simulated them using both HSPICE and our adaptive regression model. Figure 4 and 5 show the non-ideality function for the first integrator stage obtained from HSPICE and the MARS model when the feedback level is 0 volt. The average root mean square error of the model is only 0.3% in this case.

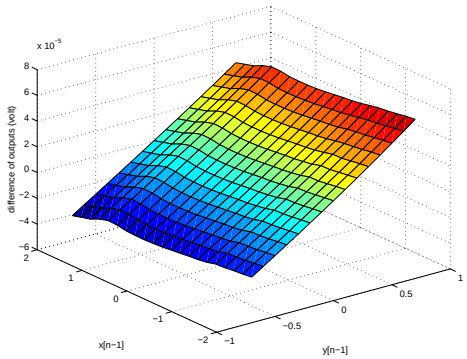


Figure 4: Non-ideality function from HSPICE.

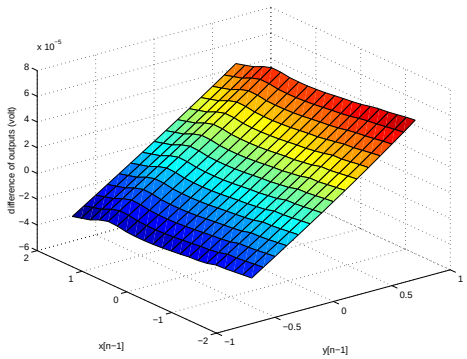


Figure 5: Non-ideality function of ARSIM model.

Sample generation is a crucial part of the modeling process. It is essential to ensure that simulation conditions during the sample generation procedure mimic the actual operation of the integrators. Hence, the output node of the integrators should be driven to a stable initial value before the sampling clock becomes active. In the case of integrator 1 the samples were made just before the end of $\phi 1$, and for integrator 2 samples were made just before the end of $\phi 2$. The switches at the output of the integrators are also included in each integrator circuit when we generate samples to accurately model the output load. Also, as we are dealing with small voltage differences with a magnitude of $10^{-4}$ volt or less it is necessary to set appropriate tolerance options to get accurate results.

Two significant advantages of our technique over table-based techniques are improved simulation time and model accuracy. Our technique moves the interpolation procedure, performed at every time step in the table-based technique, into the model generation phase, thus significantly improving the simulation time. More importantly, our technique models only the non-ideality function which yields accurate models.

## 6 Experimental Results

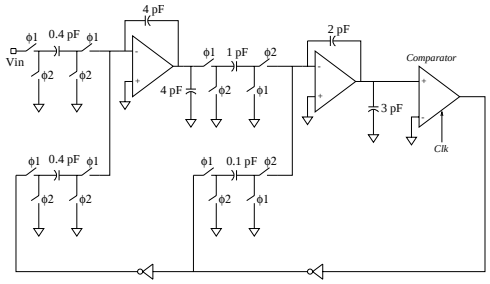


Figure 6: Second-Order $\Delta\Sigma$ Modulator

Simulation results with the proposed modeling technique are shown in this section. We have modeled and simulated the second-order modulator shown in Figure 6. The modulator has been implemented in the HP $0.8\mu$ technology from MOSIS using the BSIM3 MOS model with a 5V power supply voltage. The switched-capacitor integrators operate at 4 MHz with an oversampling ratio of 256, with the corresponding Nyquist sampling rate being 16 kHz. Experimental results with minimum as well as enlarged switch sizes are shown to illustrate the effect of increasing non-idealities on modulator performance. The input to the circuit is a 2 kHz sine wave. The signal-to-noise and distortion ratio (SNDR) results are obtained by simulating the modulator for 74k clock cycles. The first 10k data points are discarded to eliminate any initial circuit transients.

Models for both 0V and 5V DAC feedback are generated for each integrator stage during the model building phase and appropriate models are chosen at the beginning of each clock pulse during simulation depending on the feedback value.

The total time taken by ARSIM is the sum of sample generation, model building and simulation times. ARSIM takes only 46.21 minutes for generating the SNDR curve with 30 different input signal amplitudes while HSPICE takes an estimated 1894.4 hours for simulating the modulator shown in Figure 6. Since it would be impossible to

simulate for such a long time, the HSPICE time was estimated from the first 100 clock simulations. The overall speedup is 2460 times over HSPICE for this design. The details of the simulation times for ARSIM are given below.

- Sample generation for $\sim 400$ points for both integrator stages: 31.26 mins,
- Model building for both integrators: 55 secs,
- Simulating the model of the modulator for 74k clocks for one SNDR point: 28 secs.

These simulations were performed on a SUN $UltraSPARC^{TM} - II$ machine operating at 400MHz with 512 MB memory. Figure 7 shows the simulation result with ARSIM for 30 different input amplitudes for two circuit implementations.

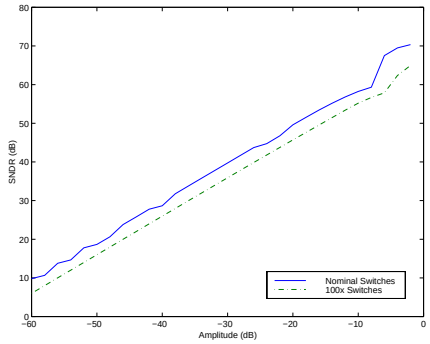


Figure 7: SNDR performance for a sampling frequency of 4MHz.

We have performed experiments with 2 different circuit implementations of the modulator. One with minimum switch sizes and the other with a less optimized circuit with enlarged switches in order to make the effect of clock-feedthrough more pronounced. The enlarged switches are 10 times lareger in width and length compared to the nominal switches. Even though the ideal peak SNDR can approach 100 dB for this second-order modulator with oversampling ratio of 256, the nonidealities from the operational amplifiers and switches significantly reduce the modulator performance.

Since the signal swing of the integrators is limited by the power supply and the clipping of the integrator outputs would significantly affect the performance, usually each integrator of $\Delta\Sigma$ modulator is preceded by an attenuation [2]. In this experiment, the attenuation of 0.1 preceding the first integrator is selected [3, 5]. For the ideal simulation, the overall SNDR will not be affected by these attenuations. In a real circuit implementation, however, signal attenuation decreases signal power, and causes the SNDR of the modulator to be affected more by the non-ideality function, resulting in reduced SNDR performance. As observed from Figure 7 the peak Signal-to-Noise and Distortion Ratio is 70.5 dB, and it decreases with increasing switch sizes, implying that non-idealities have increased.

## 7 Conclusions

A verification technique for $\Delta\Sigma$ modulators using adaptive regression modeling has been proposed. The models derived for each integrator stage have errors less than 0.3% compared with HSPICE results in our experiments. Furthermore, a speedup of three orders of magnitude is achieved over HSPICE. Our approach is to model the non-ideality function instead of the entire non-ideal response, thus modeling the non-idealities in the individual integrator stages with greater precision.


### Acknowledgements

The authors would like to express their gratitude to Professor Mani Soma and Giri Devarayanadurg of the University of Washington and Dr. Pramodchandran N. Variyam of Texas Instruments for their support for the MARS program.


## References


[1] S. R. Norsworthy, R. Schreier and G. C. Temes, "Delta-sigma data converters: theory, design and simulation," *IEEE press*, 1997

[2] B. E. Boser and B. A. Wooley, "The design of sigma-delta modulation analog-to-digital converters," *IEEE J. of Solid-State Circuits*, pp.1298-1308, December 1988

[3] A. Opal, "Simulation of oversampled sigma delta convertors," *Proc. IEEE ISCAS*, pp.727-730, 1996

[4] A. Opal, "Sampled data simulation of linear and nonlinear circuits," *IEEE Trans. Computer-Aided Design*, Vol.15, No.3, pp.295-307, March 1996

[5] G. T. Brauns, R. J. Bishop, M. B. Steer, J. J. Paulos and S. H. Ardalan, "Table-based modeling of delta-sigma modulators using ZSIM," *IEEE Trans. Computer-Aided Design*, pp.142-150, February 1990

[6] R. J. Bishop, J. J. Paulos, M. B. Steer and S. H. Ardalan, "Table-based simulation of delta-sigma modulators," *IEEE Trans. Circuits and Systems*, Vol.37, No.3, pp.447-451, March 1990

[7] S. Rabii and B. A. Wooley, "A 1.8-V digital-audio sigma-delta modulator in 0.8-$\mu$m CMOS," *J. of Solid-State Circuits*, Vol.32, No.6, pp.783-796, June 1997.

[8] J. H. Friedman, "Multivariate additive regression splines," *Annals of Statistics*, Vol.19, pp.1-141, March 1991

[9] V. Cherkassky and F. Mulier, *Learning from data*, John Wiley & Sons, 1998

[10] P. N. Variyam and A. Chatterjee, "Enhancing test effectiveness for analog circuits using synthesized measurements," *Proc. IEEE VTS*, pp.132-137, 1998

[11] G. Devarayanadurg, P. Goteti and M. Soma, "Hierarchy based statistical fault simulation of mixed-signal ICs," *Proc. IEEE ITC*, pp.521-527, 1996