Testing of Analog Systems Using Behavioral Models and Optimal Experimental Design Techniques

Eric Felt Alberto Sangiovanni-Vincentelli

Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720

Abstract

This paper describes a new CAD algorithm which performs automatic test pattern generation (ATPG) for a general class of analog systems, namely those circuits which can be efficiently modeled as an additive *combination of user-defined basis functions. The algorithm is based on the statistical technique of I-optimal experimental design, in which test vectors are chosen to be maximally independent so that circuit performance will be characterized as accurately as possible in the presence of measurement noise and model inaccuracies. This technique allows analog systems to be characterized more accurately and more efficiently, thereby significantly reducing system test time and hence total manufacturing cost.*

1 Introduction

The complexity of electronic systems being designed today is increasing in many dimensions: on one hand, the number of components is growing constantly; on the other, several radically different functions must be integrated. For example, in the exploding personal communications market, a device is the combination of wireless transmission, analog and digital signal processing, and digital computing. In this device, antennas, radio-frequency components, and analog and digital subsystems have to be designed in a unified way to meet the performance, power, and size requirements of the application.

The presence of analog components in these systems complicates their testing significantly. Analog circuits, in general, require much longer testing times than digital circuits because second-order effects must be considered and because few CAD tools are available to aid in the design of the test vectors. Analog testing is currently performed on a relatively ad-hoc basis; a design or test engineer relies primarily upon intuition about a circuit's internal functionality to derive the circuit's test suite. This test suite frequently defaults to the complete set of circuit specifications. This approach is becoming increasingly expensive in both test development and test execution times. The specifications of mixed analog-digital circuits are usually very large (e.g. see [1]), which not only results in long manual test development, but also in prohibitive testing times on very expensive automated test equipment (ATE) with mixed-signal capabilities; it is estimated that testing currently accounts for 30% of total manufacturing cost [2]. Furthermore, the use of sophisticated CAD tools has reduced the design cycle so that the influence of testing on time-to-market and final cost of the circuit is increasingly significant. For these reasons, analog testing is considered to be one of the most important problems in analog and mixed-signal design.

In this paper we present an algorithm for deriving a minimal set of test vectors for fully testing the performance specifications of a general class of analog systems. The class of systems to which the algorithm can be applied are those systems which can be modeled in a linear function space, i.e. the system output must be an *additive* combination of user-specified basis functions. Note that the basis functions themselves do not have to be linear. Mathematically the model is formulated as

$$Y = \beta_0 g_0 + \beta_1 g_1 + \beta_2 g_2 + \beta_3 g_3 + \ldots + \beta_n g_n \qquad (1)$$

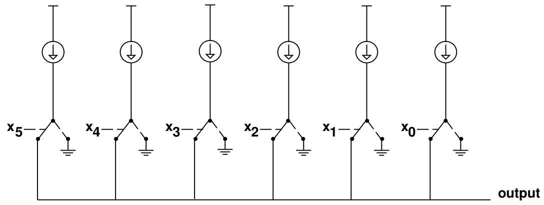


Figure 1: Six switchable current sources connected in parallel.

where Y is the system output, $\{g_i\}$ is a set of arbitrary basis vectors (user-specified), and β_i is the coefficient of the i^{th} basis vector. Many analog systems can be accurately modeled in this fashion.

The algorithm is based upon the statistical theory of optimal experimental design, in which test vectors are chosen to be maximally independent so that the system performance Y will be characterized as accurately as possible in the presence of measurement noise and model inaccuracies. More specifically, we wish to choose the test vectors to minimize the average standard error of the predicted output, thereby maximizing the likelihood that we will be able to conclusively verify that the performance specifications have or have not been met after a minimum number of test vectors. If the minimum number of test vectors is not sufficient to conclusively verify the performance specifications, then additional test vectors are selected and applied, one at a time, until the standard error of the predicted output is low enough to verify the performance specifications. Linear regression is used to analyze the results of the tests and compute the required standard errors.

2 Motivation

Consider a simple system of six current sources connected in parallel, as shown in Figure 1. Each of the current sources can be turned on or off by the controlling inputs $x_5, x_4, \ldots, x_0$. The sources were designed to each output one unit of current when on, but may actually output slightly more or slightly less than that amount because of manufacturing nonidealities.

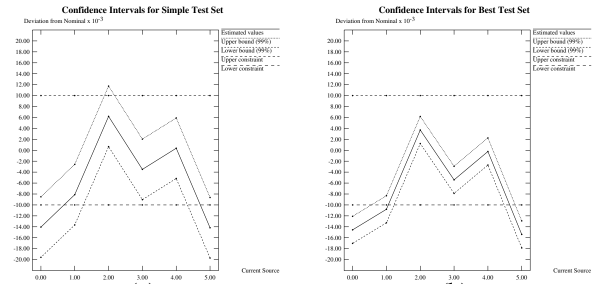
Suppose that one wishes to test this system to insure that the amount of current each sources outputs is within 1% of its nominal value. There are six independent current sources which must be measured, so at least six test vectors must be applied. Furthermore, suppose that the ammeter used to measure the current at the output is known to be accurate to within 0.005 units of current.

The simplest set of test vectors that can be imagined is probably Test Set 1 in Table 1, which tests each current element in turn by setting one of the x_i's to 1 while leaving the others at 0.

An alternative set of possible test vectors, which also happens to be a provably optimal set of test vectors for this circuit, is shown as Test Set 2 in Table 1. Both sets of test vectors can be used to estimate the actual current that each source outputs. The two test sets differ, however, in the accuracy with which they can make this measurement. Figure 2 shows the 99% confidence intervals which could be constructed after applying each set of test vectors to a

Test Set 1						Test Set 2					
x_5	x_4	x_3	x_2	x_1	x_0	x_5	x_4	x_3	x_2	x_1	x_0
1	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	1	1	0	0	1
0	0	0	1	0	0	0	1	1	1	0	0
0	0	0	0	1	0	1	1	0	0	0	0
0	0	0	0	0	1	1	1	1	0	1	0

Table 1: Two sets of test vectors for testing current sources.



(a) (b)

Figure 2: (a) Confidence intervals from applying Test Set 1. (b) Confidence intervals from applying Test Set 2.

simulated deterministic system. For Test Set 1 it can be concluded (with $> 99\%$ confidence) that the x_4 and x_3 current sources fall within their specifications, but it is not possible to draw any definite conclusions about the other current sources. For Test Set 2, on the other hand, the confidence intervals are much tighter and it can be concluded (with $> 99\%$ confidence) that the x_5 and x_0 current sources fall outside their specifications and that the x_4, x_3, and x_2 current sources fall within their specifications. Thus the simple test set leaves four estimates uncertain while the best test set leaves only one estimate uncertain.

A common technique for tightening the confidence intervals is to repeat each test vector several times and then average the results. Each test vector would have to be applied five times, however, for a total of 30 tests, to obtain the same confidence intervals that can be found from one application of the six vectors in the best test set.

From this simple example it is clear that choosing a "good" set of test vectors is desirable because it will lead to more accurate characterizations of system output, and hence possibly smaller test sets. Good test sets are not intuitively obvious, however, even for very simple systems. In the remainder of this paper we discuss some new CAD techniques which have been developed and implemented for finding these good test sets.

3 Previous Work

Most of the previous work in ATPG for electrical systems has been directed at digital circuits, and efficient techniques have been developed for both combinational and sequential digital systems [3, 4]. These test systems are based on the common single stuck-at-0/stuck-at-1 fault model and the controllability and observability of each fault.

Unfortunately most of these ideas cannot be directly applied to analog systems. One of the major problems is that analog systems are, in general, much more difficult to model because second- and higher-order effects must be considered to accurately predict system performance. Another major difference is that analog circuits are less susceptible to catastrophic (e.g. stuck-at) faults, because of the typically larger device and wire sizes, but much more susceptible to parametric faults, which are small variations in component values that cause the system performance to violate its specifications.

In the area of analog testing, work has been done in system modeling and test ordering, both of which are useful toward the goal of reducing testing time. In the area of system modeling, Sounders and Stenbakken proposed using linear models, which can be derived either from simulation [5] or from manufacturing data [6] using QR decomposition. In the area of test ordering, Milor has described an algorithm for minimizing average test time by ordering tests in such a way that the tests which are most likely to detect faults are performed first [7].

In this paper we assume that a suitable linear behavioral model for the system exists (or can be easily derived from sensitivity analysis), and we focus on a new optimization algorithm for selecting the best set of test vectors.

Furthermore, we propose the use of statistical *confidence intervals* to verify system performance in the presence of measurement noise and model inaccuracies. Test points are chosen to make these confidence intervals as tight as possible.

4 Algorithm

Given an arbitrary basis consisting of n functions which span an n-dimensional space, at least n test vectors must be applied to the system in order to fully characterize the output function. If fewer than n input vectors are applied, then at least one dimension of the space remains unexplored and hence the output function is unconstrained in that dimension. Furthermore, because of inevitable measurement noise, n test vectors may not be sufficient to verify that the output function falls within the desired bounds. Using additional test vectors will lower the standard error of the estimates, thereby making it more likely that the system output can be verified to fall within the desired limits.

With these factors in mind, the testing algorithm that we propose is

1. Apply n "maximally orthogonal" vectors to the system, where n is the dimensionality of the space to be characterized.
2. Generate the estimated response function and confidence intervals for that response function.
3. If the confidence intervals for the response function fall definitely within the system performance specifications at all points, then accept the chip.
4. If the confidence intervals for the response function fall definitely outside the system performance specifications at any point, then reject the chip.
5. Otherwise apply additional test vectors which are "maximally orthogonal" to those already applied, one at a time, until the confidence intervals are small enough to determine whether or not the chip meets its specifications.

The choice of test vectors is a difficult optimization problem. The objective is to minimize the standard error of the estimated response function, which is a function of the choice of test vectors. Intuitively, the orthogonality of the test vectors is measured by the degree to which each test vector maximizes the contribution of one basis function while minimizing the contribution of the others.

The algorithm used to derive the maximally orthogonal test vectors is

1. Eliminate any redundant basis vectors.
2. Run the I-optimality algorithm to select best n tests, where n is the dimensionality of the function space after eliminating redundant basis vectors.
3. Run the I-optimality algorithm to select best additional vectors, one at a time, for use if the prior tests are not conclusive.

4.1 Optimality Criteria

There are several different optimality criteria (A-, D-, E-, G-, and I-), the relative merits of which have been debated extensively in the relevant literature [8, 9]. D-optimality, which is generally considered to be the simplest type of optimality, minimizes the average prediction variance of the model coefficients. This type of optimality would be very suitable for fault diagnosis, in which we wish to estimate the actual values of each circuit component as accurately as possible. D-optimality claims nothing about the the average prediction variance of the system output, however, so it is not the best choice for verifying that the system output meets its specifications.

The two types of optimality which do consider the prediction variance of the system output are G- and I-optimality. G-optimality minimizes the *maximum* prediction variance over the response surface of interest. It would probably be the most suitable for verifying that a circuit meets its specifications, since specifications are frequently stated as worst-case bounds. G-optimality is difficult to optimize upon, however, because it is not continuously differentiable. Hence I-optimality, which is continuously differentiable, was chosen for this research. I-optimality minimizes the *average* prediction variance over the response surface of interest. To formulate these ideas mathematically, let

$$y = f(g_1, g_2, \ldots, g_p) + \epsilon \tag{2}$$

where y is the response variable, g_i are the independent basis vectors, and ϵ represents the measurement and modeling errors, which are assumed to be independent with mean 0 and variance σ^2.

Let X be the design matrix, which contains one row for each of the n test vectors.

$$X = \begin{bmatrix} g_1(x_1) & g_2(x_1) & g_3(x_1) & \ldots & g_p(x_1) \\ g_1(x_2) & g_2(x_2) & g_3(x_2) & \ldots & g_p(x_2) \\ & & \vdots & & \\ g_1(x_n) & g_2(x_n) & g_3(x_n) & \ldots & g_p(x_n) \end{bmatrix} \tag{3}$$

The design moment matrix M_X can be calculated as

$$M_X = \frac{1}{n} X^T X \tag{4}$$

and the prediction variance at an arbitrary point x on the response surface is

$$var \;\; \hat{y}(x) = \frac{\sigma^2}{n} f(x) M_X^{-1} f(x)^T. \tag{5}$$

An I-optimal design is one which minimizes the average of this variance over the response surface R

$$I = \frac{n}{\sigma^2} \int_R var \;\; \hat{y}(x) d\mu(x). \tag{6}$$

This integral simplifies [9] to give

$$I = trace\{M M_x^{-1}\} \tag{7}$$

where M is the moment matrix of the region of interest R.

$$M = \int_R f(x)^T f(x) d\mu(x) \tag{8}$$

4.2 Optimization

Finding an exactly I-optimal design is believed to be NP-complete [10] and hence only feasible for very small problems. For larger problems, several heuristic algorithms have been successfully used to find "good" solutions to this and other related problems in the area of optimal experimental design. These heuristic algorithms include simulated annealing [10], greedy swap techniques [11], and gradient descent techniques. For this research we used the gradient descent techniques implemented in the software package `gosset`, which was recently developed by Hardin and Sloane at AT&T Bell Laboratories [12]. The primary focus of `gosset` is low-order polynomial models, which are of only limited use in characterizing typical analog circuits. For this research, therefore, `gosset` was extended to utilize arbitrary Lipschitz continuous functions, such as the piecewise linear output of common behavioral simulators [13] and SPICE [14].

`Gosset` uses an optimization algorithm known as *Hooke and Jeeves pattern search* [15], which is based on the idea of finding a "valley" and following it downward until reaching the lowest point on the response surface, similar to the manner in which a stream flows down a mountain. The optimization begins by selecting a random point on the response surface, calculating the gradient at that point, and proposing a set of small perturbations in the direction of the gradient. If this set of perturbations causes the objective function to improve, then this "move" is accepted and the step size is increased by a constant factor. Otherwise the set of perturbations is rejected and a smaller move is attempted.

The initial point in the search space, $x^{(0)}$, is chosen randomly. The initial velocity vector $v^{(0)}$ is set to 0, where the velocity v_i of input i is defined as being the amount by which that input is perturbed in a given move. The step size s is set to a small value. The search then proceeds as

$$x^{(i+1)} = x^{(i)} + v^{(i+1)} \tag{9}$$

$$v^{(i+1)} = v^{(i)} + s g(x^{(i)}) \tag{10}$$

where $g(x^{(i)})$ is the gradient evaluated at the point $x^{(i)}$. If

$$F(x^{(i+1)}) < F(x^{(i)}), \tag{11}$$

where F is the objective function, then the value for $x^{(i+1)}$ is accepted, s is multiplied by 1.04, and the iteration is repeated. If $F(x^{(i+1)}) \not< F(x^{(i)})$ then $v^{(i)}$ is set to 0 and (9) and (10) are tried again. If there is still no reduction in F, then s is divided by 2 and (9) and (10) are tried again. The algorithm terminates when the step size is less than some small accuracy limit. Then, if desired, a new random starting point can be chosen and the entire minimization algorithm repeated, successively, until a specified number of random starts have been investigated. At that point the algorithm terminates, returning the best design found.

If $x^{(i)}$ moves outside the feasibility region, which is defined by the limited range of values that each input can assume, then it is moved to the closest feasible point.

Note that the optimization assumes that all of the inputs to the system are continuous. If the inputs are discrete, as frequently occurs when analog systems are connected to digital systems, then a post-processing step is performed which is similar to integer programming. Each of the test vectors is sequentially considered, and discrete inputs with illegal values are converted to whichever of the two closest discrete values gives the smallest value of F. The technique is essentially *greedy* integer programming, since the order in which the inputs are considered could cause the algorithm to become stuck in a local minimum. We have empirically observed that the algorithm works well because

1. The optimization pushes many variables to their boundaries, which are usually legal discrete values, and
2. The [usually slight] non-optimality introduced by the rounding off of one test vector can frequently be partially compensated for by the rounding off of a similar test vector in the opposite direction.

Hooke and Jeeves found empirically, in a curve-fitting problem involving a neutron reactor, that the computation time for their pattern search algorithm increased only *linearly* with the number of variables, which makes it especially suitable for the analog testing problem because analog systems may require large numbers of parameters to accurately characterize them.

5 Model Derivation

The statistical design and analysis techniques which we use for system testing require a *homoskedastic*, *linear* function space. The linearity requirement means that any system output can be expressed as an additive combination of a set of basis vectors, as shown in Equation 1. Homoskedastic refers to a requirement that the measurement error ϵ, which is a combination of model inaccuracies and noise, is not a function of the input; this assumption is reasonable for many typical analog testing situations.

There are several simple methods which can be used to choose the $\{g_i\}$ basis vectors. For extremely simple systems the basis functions may be obvious from a simple description of the expected output. Consider, for example, the current sources discussed in Section 2; the output is modeled as

$$Y = \beta_5 x_5 + \beta_4 x_4 + \beta_3 x_3 + \beta_2 x_2 + \beta_1 x_1 + \beta_0 x_0 \qquad (12)$$

where the β coefficients are the unknown model parameters we wish to characterize. The basis functions for this system are simply $\{x_5, x_4, x_3, x_2, x_1, x_0\}$, the set of contributions from each current source, which are *added* together to form the output.

For more complicated systems, the Taylor expansion can be used to derive a very useful additive model

$$f(a+x) = f(a) + xf'(a) + \frac{x^2 f''(a)}{2!} + \ldots + \frac{x^{n-1} f^{(n-1)}(a)}{(n-1)!} \qquad (13)$$

where a represents the nominal value of a model parameter, $f(a)$ represents the value of the output when that model parameter is at its nominal value, and x represents the amount by which that model parameter deviates from its nominal value because of manufacturing nonidealities. We wish to estimate $f(a+x)$.

A first-order Taylor series approximation is a reasonably accurate model for many common analog systems with parameters that do not deviate significantly from their nominal values. This is the model used by Stenbakken and Sounders [16], and our discussion of it here will be brief. Dropping the higher-order terms and generalizing to multiple dimensions, the expansion becomes

$$f(a+x) = f(a) + \bigtriangledown f(a) x \qquad (14)$$

$$= f(a) + \frac{\partial f}{\partial a_1} x_1 + \frac{\partial f}{\partial a_2} x_2 + \ldots \qquad (15)$$

where a_i is the nominal value of the i^{th} model parameter and x_i is the deviation in that parameter. The basis functions for this system are thus $\{f(a), \frac{\partial f}{\partial a_1}, \frac{\partial f}{\partial a_2}, \ldots\}$. $f(a)$ is the nominal system performance, and each of the partial derivatives represents an error signature for a particular type of manufacturing defect which can occur. The error signatures are computed by finding the sensitivity of the output to the parameters of interest at each point on the response surface. Note that these error signatures could represent either catastrophic faults, such as shorts and disconnections, or parametric faults, such as small deviations in capacitance values or process parameters.

Once the basis vectors $\{g_1, g_2, \ldots, g_n\}$ are identified, we compute their null space to verify that they are all independent. The parameters associated with error signatures that are linearly dependent are said to belong to the same *ambiguity group*, since variations in those parameters are indistinguishable at the system output. Ambiguity groups reduce the number of basis vectors needed to model the response surface and hence the number of test vectors which must be applied to fully characterize a system.

Let U be the matrix formed from these basis vectors, where g_i is the i_{th} column of U. Suppose U has dependent columns, then its null space is non-empty such that

$$UN = 0 \qquad (16)$$

where $N \in R^{m \times r}$ is a matrix with r independent column vectors that spans the null space of U. Non-zero entries in N indicate that the corresponding components are in ambiguity groups. A component i belongs to an ambiguity group if and only if row i of N has a non-zero entry. Furthermore, components i and j are in the same ambiguity group if rows i and j of N are non-zero and not orthogonal to each other [17]. It follows that the components fall into the same group if their corresponding row vectors of N are non-zero and not orthogonal.

The null space of U can be computed using singular value decomposition (SVD) or Gaussian elimination. In the case of SVD, we first compute $U^T U$, followed by SVD

$$U^T U = X_1 X_2 N^T \qquad (17)$$

where N spans the null space of $U^T U$. Since $U^T U N = 0$, $UN = 0$, so N is the null space of U also. The reason for computing $U^T U$ in (17) is that U often has many more rows than columns, so computing N for a smaller matrix $U^T U$ is more efficient. Furthermore, note that computing N and checking the rows of N for pairwise orthogonality can be performed in polynomial time.

The final set of basis vectors which will be used to characterize the system, then, is

1. all of the basis vectors which are not in any ambiguity group, combined with
2. one vector from each ambiguity group.

Once this set is formed, the I-optimality routines, as described in Section 4, are executed to find a good set of test vectors.

Once the test vectors have been applied, the measured responses are used to estimate $\hat{\beta}$, the vector of coefficients for each of the basis vectors. For the special case when the number of test points is equal to the number of basis functions, $\hat{\beta}$ is found by solving

$$X\hat{\beta} = Y \qquad (18)$$

for $\hat{\beta}$, where X is the design matrix as output by the I-optimality routine and Y is the vector of measured responses. When the number of test points is greater than the number of basis functions, $\hat{\beta}$ is found by solving

$$X^T X \hat{\beta} = X^T Y \qquad (19)$$

for $\hat{\beta}$, which is a linear regression.

A 99% confidence interval for the $\hat{\beta}$ estimates over the entire response surface is computed by applying Equations 4 and 5 from Section 4.1, followed by

$$CI(\hat{\beta}) = \left(\hat{\beta} - 2.576\sqrt{var(\hat{\beta})}, \hat{\beta} + 2.576\sqrt{var(\hat{\beta})}\right), \qquad (20)$$

where the factor 2.576 is used because it is the 99.5% quantile of a normal distribution, which leaves only a 1% probability that the actual value of the response at that function falls outside of the confidence interval.

Note that we have assumed that the variance σ^2 of the error term ϵ is known by the designer. If this is not the case, then S^2, an estimator of σ^2, can be derived from the observations Y as

$$S^2 = \frac{\sum_{i=1}^{n} [Y_i - \hat{\mu}(x_i)]^2}{(n-d)} \qquad (21)$$

where Y_i is the i^{th} observation, $\hat{\mu}(x_i)$ is the predicted value of Y_i, based on the model, n is the number of measurements, and d is the dimensionality of the model (the number of independent basis vectors). When S^2 is used instead of σ^2 then the confidence intervals are constructed by using the t-distribution quantiles with $n-d$ degrees of freedom instead of the normal quantiles. Note that the t-distribution approaches the normal distribution for large n (e.g. $n > 30-40$), so the normal distribution can often be used as an approximation to the appropriate t-distribution when generating the confidence intervals from S^2 instead of from σ^2.

6 Results

In this section we describe some practical examples of analog systems on which our ATPG algorithms have been run. The first is a bandpass filter with center frequency of 24.5 kHz [18], which was analyzed using SPICE sensitivity analysis. The second is a single MOS transistor, which was analyzed in SPICE with a level three transistor model. The third is a 6-bit Nyquist-rate D/A converter which was analyzed using explicit behavioral equations [13].

6.1 Bandpass Filter

Figure 3 shows a linear model for a bandpass filter. As basis functions we select the constant function, the nominal frequency

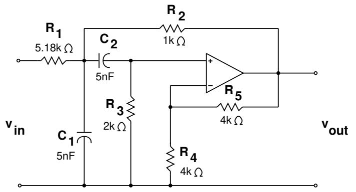


Figure 3: Bandpass filter with center frequency at 24.5 kHz.

response, and the sensitivities of the nominal response with respect to R_1, C_1, R_2, C_2, R_3, R_4, and R_5.

Running the ambiguity algorithm reveals that many of the component sensitivities are linearly dependent; the constant function, the nominal response, and the sensitivities with respect to C_2, C_1, and R_1 are sufficient to fully characterize the filter.

Since there are five basis functions in the model, at least five test frequencies will be needed to estimate the system response. We impose a constraint that the test frequencies lie between 15kHz and 40kHz, since that is the region of the response in which we are interested, and run the I-optimality algorithm. The five test frequencies which the algorithm selects are shown in Table 2. Note that the fifth test point is pushed to the user-imposed limit of 40kHz, while the remaining test points sample the response at intervals of approximately 3 kHz near the nominal center frequency. Furthermore, if the constant function is eliminated from the model, then the test point at 40.00 kHz is no longer needed and disappears from the test set.

Frequency	| Output |
19.32 kHz	0.914
22.57 kHz	1.65
24.89 kHz	1.99
28.42 kHz	1.30
40.00 kHz	0.482

Table 2: Test frequencies chosen for bandpass filter.

Applying the five selected test frequencies to a simulated circuit produces the estimated output and 99% confidence intervals shown in Figure 4. According to the testing algorithm outlined in Section 4, these confidence intervals would be compared against the filter specifications to determine whether the component should be accepted or rejected, or whether additional test vectors should be applied to tighten the confidence intervals.

6.2 MOS Transistor

Suppose we wish to test an MOS transistor to verify that its drain current I_{DS} falls within certain bounds over all values of V_{GS} and V_{DS}. The manufacturer has provided a level 3 SPICE model for the device with the parameter values shown in Figure 5.[1] As basis functions, we select the constant function, the nominal performance, and the sensitivities with respect to V_{T0}, k', γ, t_{ox}, and θ. The normalized basis functions are shown graphically for three values of V_{GS} in Figure 6. Although only three values of V_{GS} are shown, both V_{DS} and V_{GS} are treated as continuous variables, so the response surface is 2-dimensional and continuous.

Running the ambiguity group algorithm, we find that k' and the nominal performance are linearly dependent on the entire response surface, which can be seen in Figure 6(a) and 6(c). So the k' vector is dropped from the model, and there are 6 remaining independent basis functions. Therefore we will need at least 6 test points to characterize the device.

[1] This model is based on the HP CMOS26B 0.8 μm process.

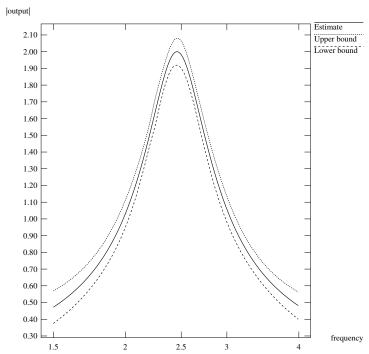


Figure 4: Estimated output and 99% confidence intervals for bandpass filter.

```
.MODEL nom NMOS LEVEL=3 PHI=0.600000 TOX=2.0300E-08 XJ=0.150000U
+ TPG=1 VTO=0.7333 DELTA=9.4450E-01 LD=1.0000E-09 KP=1.2964E-04
+ UO=762.1 THETA=5.2460E-02 RSH=2.3650E+00 GAMMA=0.4481
+ NSUB=1.7500E+16 NFS=2.3560E+12 VMAX=1.4870E+05 ETA=1.4850E-01
+ KAPPA=9.5100E-02 CGDO=2.5516E-12 CGSO=2.5516E-12
+ CGBO=3.0108E-10 CJ=1.1962E-04 MJ=0.4398 CJSW=4.6935E-10
+ MJSW=0.123994 PB=0.800000
```

Figure 5: SPICE model for MOS transistor.

To prevent the I-optimality algorithm from selecting unreasonable test points, we impose constraints on the inputs V_{GS} and V_{DS} such that $0.1V \leq V_{DS} \leq 10.0V$ and $2.0V \leq V_{GS} \leq 5.0V$. We then run the I-optimality algorithm; it selects the test points shown in Table 3.

V_{GS}	V_{DS}	I_{DS}
5.0	1.0	19.04910×10^{-3}
3.2	0.9	10.00540×10^{-3}
5.0	0.1	2.66950×10^{-3}
5.0	10.0	33.62560×10^{-3}
2.0	10.0	6.24432×10^{-3}
5.0	2.2	27.45550×10^{-3}

Table 3: Test points chosen for MOS transistor.

Figure 7(a) shows the estimated response curves for three values of V_{GS} after applying the indicated 6 test vectors to a device, along with the 99% confidence intervals for those estimates. The confidence intervals are based upon a measurement accuracy of 0.1%. The expected value of the model error at $V_{GS} = 3.5V$ is shown in Figure 7(b), from which we conclude that our estimates are least accurate near $V_{GS} = V_T$. This fact is not surprising, since that region of transistor operation is difficult to model.

6.3 Nyquist-rate D/A Converter

A 6-bit Nyquist-rate D/A converter based on binary-weighted current sources is shown in Figure 8, which is similar to the simple example presented in Section 2 except that in this case the current sources are binary-weighted instead of unit-weighted. The basis vectors for the system are chosen to be $\{1,\ x_5,\ x_4,\ x_3,\ x_2,\ x_1,\ x_0\}$, where the constant function 1 is used to model the converter offset. Since there are seven independent basis functions in the model, at least seven tests must be performed to fully characterize the system. The I-optimal design is shown in Table 4, along with

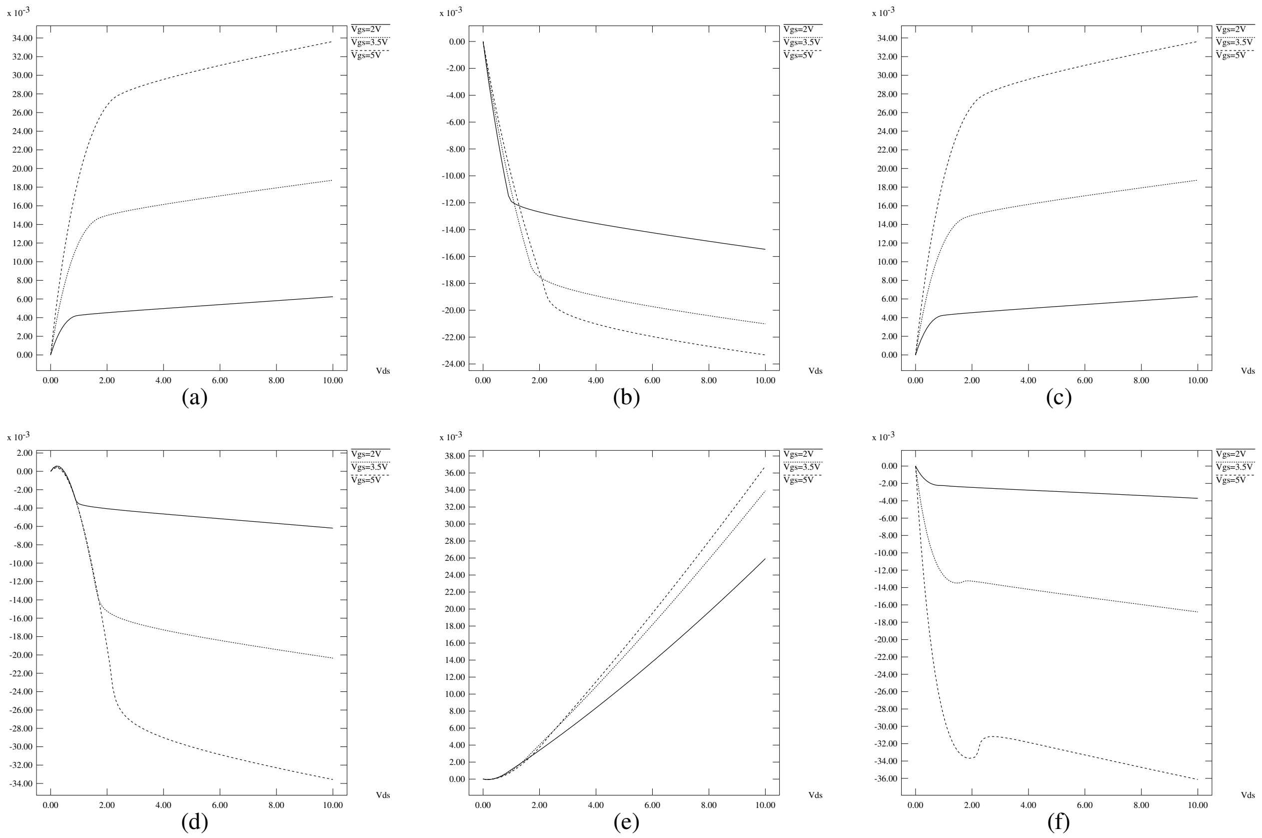


Figure 6: (a) Nominal I_{DS} vs. V_{DS} transistor curves. (b) Sensitivity w.r.t. V_{T0}. (c) Sensitivity w.r.t. k'. (d) Sensitivity w.r.t. γ. (e) Sensitivity w.r.t. t_{ox}. (f) Sensitivity w.r.t. θ.

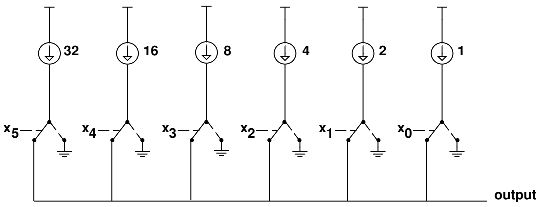


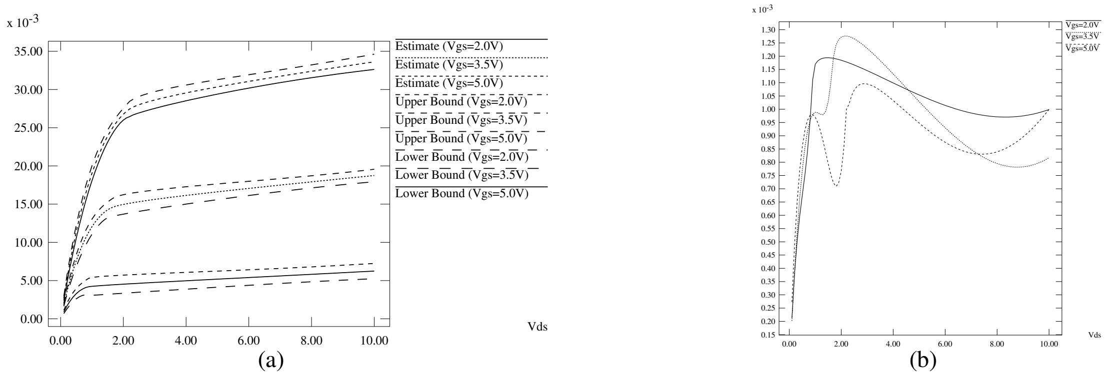
Figure 8: 6-Bit binary-weighted current source D/A converter.

the next seven extra points which would be chosen, in succession, to tighten the confidence intervals on the estimated performance.

Application of the seven initial test vectors to a simulated DAC produces the results shown in Figure 9. The upper and lower confidence intervals illustrate how the entire performance of the DAC can be modeled quite accurately after the application of only seven well-chosen test vectors. Furthermore, we may be able to draw some conclusions regarding the acceptability of this DAC, depending upon the INL specification. If the INL specification is greater than 0.2 LSB, then the DAC should be accepted with no further tests. If the INL specification is less than 0.1 LSB, then the DAC should be rejected with no further tests. If the INL specification falls between these bounds, then additional test vectors must be applied to tighten the confidence intervals.

Code	**Inputs**						**I-Value**
	x_5	x_4	x_3	x_2	x_1	x_0	
8	0	0	1	0	0	0	
15	0	0	1	1	1	1	
21	0	1	0	1	0	1	
22	0	1	0	1	1	0	
35	1	0	0	0	1	1	
44	1	0	1	1	0	0	
59	1	1	1	0	1	1	1.27778
61	1	1	1	1	0	1	1.12500
38	1	0	0	1	1	0	0.97619
1	0	0	0	0	0	1	0.83333
48	1	1	0	0	0	0	0.70000
48	1	1	0	0	0	0	0.58333
26	0	1	1	0	1	0	0.55263
10	0	0	1	0	1	0	0.52222
All 64 codes							0.10938

Table 4: Test vectors chosen for D/A converter.



(a) (b)

Figure 7: (a) Estimated response and confidence intervals for MOS transistor from seven test points. (b) Standard error of estimate.

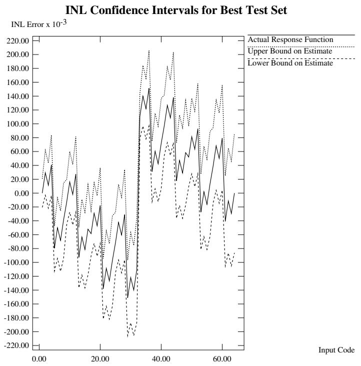


Figure 9: Upper and lower bounds on INL error from seven test vectors.

7 Conclusions

We have presented a new CAD algorithm which automatically generates a minimal set of test vectors for characterizing a general class of analog circuits, namely those circuits which can be efficiently modeled as an additive summation of user-defined basis functions. The algorithm chooses the set of test vectors so as to minimize the average prediction variance of the model. Applying the minimal set of test vectors to a circuit produces an estimate of the circuit's performance for all possible input vectors and, more importantly, confidence intervals on those estimates which can be used to determine whether the component should be passed or failed, or whether additional test vectors should be applied to tighten the confidence intervals.

Because these techniques generate the tightest possible confidence intervals after a minimum number of test vectors, they represent the most efficient way of fully characterizing system performance. Tight confidence intervals will lead to reduced testing time for analog systems because more components will be fully verifiable, to a desired confidence level, with the minimum number of test vectors. We have applied the algorithm to several analog systems and have shown it to be efficient and effective.

References

[1] L. Bonet, J. Ganger, J. Girardeu, C. Greaves, M. Pendelton and D. Yatim, "Test features of the MC145472 ISDN U-transceiver", *Proc. 1990 International Test Conf.*, pp. 68–79, 1990.

[2] *Semiconductor Industry Technology Workshop Conclusions*, Semiconductor Industry Association, 1993.

[3] M. H. Schultz, E. Trischler and T. M. Sarfert, "SOCRATES: a highly efficient automatic test pattern generation system", *IEEE Trans. on CAD*, vol. Vol. 7,No. 1, pp. 126–137, 1988.

[4] A. Ghosh, S. Devadas and A. Richard Newton, *Sequential Logic Testing and Verification*, Kluwer, Boston, 1992.

[5] G. Stenbakken and T. Souders, "Test-Point Selection and Testability Measures via QR Factorization of Linear Models", *IEEE Transactions on Instrumentation and Measurement*, June 1987.

[6] T. Souders and G. Stenbakken, "Cutting the high cost of testing", *IEEE Spectrum*, pp. 48–51, March 1991.

[7] L. Milor and A. L. Sangiovanni-Vincentelli, "Minimizing production test time to detect faults in analog circuits", *IEEE Trans. on CAD*, vol. Vol. 13,No. 6, pp. 796–813, June 1994.

[8] J. Kiefer, *Collected Papers III: Design of Experiments*, Springer-Verlag, New York, 1985.

[9] G. E. P. Box and N. R. Draper, *Empirical Model-Building and Response Surfaces*, Wiley, New York, 1987.

[10] S. B. Crary, "Optimal design of experiments for sensor calibration", *Proc. IEEE International Conference on Solid-State Sensors and Actuators*, pp. 404–407, 1991.

[11] N. K. Nguyen and A. J. Miller, "A review of some exchange algorithms for constructing discrete D-optimal designs", *Computational Statistics and Data Analysis*, vol. Vol. 14, pp. 489–498, 1992.

[12] R. H. Hardin and N. J. A. Sloane, "A new approach to the construction of optimal designs", *Journal of Statistical Planning and Inference*, vol. Vol. 37, pp. 339–369, 1993.

[13] E. Liu, A. Sangiovanni-Vincentelli, G. Gielen and P. Gray, "A behavioral representation for nyquist rate A/D converters", in *Proc. IEEE ICCAD*, pp. 386–389, November 1991.

[14] A. Vladimirescu, A. R. Newton and D. O. Pederson, *SPICE Version 2G.1 User's Guide*, Dept. of Electrical Engineering and Computer Sciences, Univ. of California, Berkeley, CA, 1980.

[15] R. Hooke and T. A. Jeeves, "'Direct Search' solution of numerical and stastical problems", *J. Assoc. Comp. Machinery*, vol. Vol. 8, pp. 212–229, 1961.

[16] G. N. Stenbakken and T. M. Sounders, "Linear error modeling of analog and mixed-signal devices", *Proc. IEEE International Test Converence*, pp. 573–581, 1991.

[17] E. W. Y. Liu, W. Kao, E. Felt and A. L. Sangiovanni-Vincentelli, "Analog testability analysis and fault diagnosis using behavioral modeling", *Proc. IEEE Custom Integrated Circuits Conference*, 1994.

[18] T. M. Sounders and G. N. Stenbakken, "A comprehensive approach for modeling and testing analog and mixed-signal devices", *Proc. IEEE International Test Converence*, pp. 169–174, 1990.