# Detection of Catastrophic Faults in Analog Integrated Circuits

LINDA MILOR AND V. VISVANATHAN

*Abstract*—**The IC fabrication process contains several testing stages. Because of the high cost of packaging, the testing stage prior to packaging, called *wafer probe*, is key in reducing overall manufacturing cost. Typically in this stage, specification tests are performed. Even though specification tests can certainly distinguish a good circuit from all faulty ones, they are expensive and many types of faulty behavior can be detected by simpler tests. Hence the construction of a set of measurements which detects many faulty circuits before specification testing is described. Bounds on these measurements are specified, and an algorithm for test selection is presented. An example of a possible simple test is a test of dc voltages (i.e., parametric tests). This type of test is defined rigorously, and the effectiveness of it in detecting faulty circuits is evaluated.**

## I. INTRODUCTION

THE advent of integrated circuit technology has necessitated new approaches to testing of analog circuits. Because of the element density made possible by integrated circuit technology, fault detection can only be done by tests at a limited number of output connections. Furthermore, since most elements are inaccessibly embedded within chips, it is no longer possible to repair or replace them. Consequently faulty circuits are simply discarded and testing procedures do not need to diagnose faulty elements or even determine the location of faults. It is simply necessary to be able to distinguish a faulty circuit from a good one.

Specifications of analog circuits are typically based on the dynamic behavior of a circuit. For example, it may be required that the ac gain over a range of frequencies or the phase margin be bounded. It is possible to distinguish between a good and a faulty circuit by testing all of a circuit's specifications. This is done typically in practice in two stages in the fabrication process (Fig. 1): wafer probe and final test.

Final tests consists of a complete set of specification tests. Since specifications bound both dynamic and dc behavior of a circuit, specification tests are typically categorized as either *parametric* or *functional* tests. Functional tests are those which measure a circuit's dynamic behavior and parametric tests measure dc voltages and currents. *Because packaging and final test are more expensive than all other manufacturing steps* [1], the additional testing stage before packaging, called wafer probe, is added.


Fig. 1. The fabrication process.

Wafer probe also consists of specification tests, with the exception of high frequency measurements for which the high capacitance of the probes make it impossible to obtain accurate results. Performing a complete set of specification tests in the wafer probe stage is still very costly and many circuits have specifications which require expensive testing procedures. It is hence desirable to precede specification tests by simpler tests in wafer probe, particularly if they detect many faulty circuits. For example, a test of dc voltages (that is, additional parametric tests) could be such a test.

The idea of adding additional tests in wafer probe is not original. In some fabrication lines, additional parametric tests are included to discard obviously faulty circuits [1] and furthermore in [1] such an approach is formalized. In [1] the problem of determining bounds on each of the parametric measurements is posed as a statistical optimization problem, where manufacturing profit is maximized. The algorithm can be broken down into three parts:

1) For a sample of possible upper and lower bounds on parametric measurements, estimate wafer probe yield and final test yield, and compute profit from the yield estimates. The estimates of yield can either be done by a Monte Carlo analysis, requiring multiple circuit simulations to determine if the chip passes wafer probe and final tests, or they can be determined by taking measurements from the fabrication process. In the latter case, many circuits will need to be tested with less than optimal bounds on measurements before a large enough sample is built up to obtain a set of good tests.
2) Approximate profit by an analytical function by computing regression coefficients.
3) Maximize the analytical function which approximates profit to determine the optimal lower and upper bounds on measurements.

Manuscript received January 11, 1988; revised September 23, 1988. This work was supported by DARPA under Grant N00039-87-C-0182 and by AT&T Bell Laboratories. The review of this paper was arranged by Associate Editor M. R. Lightner.

L. Milor is with the Department of Electrical Engineering, University of California, Berkeley, CA 94720.

V. Visvanathan is with AT&T Bell Laboratories, Murray Hill, NJ.

IEEE Log Number 8824847.

Obviously, both methods proposed to compute profit in step 1) are nontrivial. Multiple circuit simulations are computationally *very* expensive, and the alternative of computing yield from the fabrication process requires the assembly and testing of a large sample of chips. This second method can be efficient if a very large number of chips need to be tested and if the correct parametric tests are chosen. The paper, however, *assumes* that we know which parametric tests to use and does not present a way of choosing them.

In this paper, like in [1], we aim to prove that adding additional tests in wafer probe can be efficient and effective. However, we go one step beyond [1] by discovering which types of faulty behavior are detectable by these tests and which ones are not. In addition, the work to be presented here exceeds that in [1] by presenting an algorithm to choose the tests to be used. This becomes possible because we do not aim to optimize manufacturing yield to determine bounds on measurements but rather we make reasonable choices of these bounds based on *tolerances on process parameters* (e.g., oxide thickness and substrate doping) which are typically specified by process engineers. By doing so we cut down the computational cost enormously. Our computation requires $n + 1$ circuit simulations for each measurement with typical simulators where $n$ is the number of independent process parameters. Hence, to compute bounds on measurements, our algorithm involves the following steps:

1) Determine a *tolerance box* containing *process parameters*. Typically this is specified by process engineers. However if we desire to optimize manufacturing profit, the regression techniques mentioned in [1] can be used here to determine the optimal size of the tolerance box, but this is not computationally cheap.
2) Map the tolerance box by the sensitivity matrix of the good circuit into the measurement space, each axis corresponding to a measurement to be introduced before specification testing in wafer probe. The measurements may be, but are *not* restricted to parametric tests and the resulting region specifies the *good signature*. It is a polytope [2] and is slightly more complex than the parallelepiped constructed by [1].

The problem of choosing an optimal set of tests which distinguishes between the good circuit and likely faulty ones is similar in formulation to the one of *diagnosis*, with the simplification that *it is only necessary to diagnose the good circuit*. This is because choosing an optimal set of tests is equivalent to finding a set of measurements for which the good circuit can be diagnosed (distinguished from all faulty ones). In the fault diagnosis literature our approach can be categorized as a *simulation-before-test* technique. Such an approach is reasonable for our application because it has been observed that for analog circuits, simulation during design is cheaper than testing time during production.

Much of the work in the area of diagnosis has been done for linear circuits. Among those techniques which deal with nonlinear circuits, typically one evaluates:

$$y = F_i(\bar{\beta}^0)$$

for each fault, where $y$ and $\bar{\beta}^0$ are vectors of measurements and parameters respectively, and $F_i(\ )$ is the faulty circuit equation. The parameters, $\bar{\beta}^0$, are at their nominal values. This vector, $y$, is stored in a dictionary as the fault signature. If we wish to diagnose a fault, a measurement of the circuit being tested is compared with measurements in the dictionary. The fault is identified by determining the closest (e.g., Euclidean norm) simulated fault signature. If our aim is simply to identify the good circuit, then a circuit is determined to be good if its response is close (by some arbitrary measure) to the simulated response of the good circuit with parameters at nominal values [3].

This approach is not robust because good parameters do not lie exactly at their nominal values but are often guaranteed to vary within some prescribed tolerance. Consequently it is unlikely that measurements will have values exactly equal to those stored in the fault dictionary. Furthermore because of tolerance, *some faults may have measurements equal to those of the good circuit when their parameters are within tolerance. These faults belong to the same ambiguity group as the good circuit and cannot be detected by the given set of measurements.* An algorithm which rigorously determines ambiguity groups, solves the diagnosis problem and can be used to find a set of measurements for which the good circuit can be distinguished from all faulty ones.

Attempts have been made to aggregate faults into ambiguity groups. In [4] this aggregation is left to the intuition of the test designer. This is obviously not robust, since the aggregation is not rigorously justified in terms of the variations of parameters. Consider the plot of good and faulty nominal voltages at two outputs of an op amp (which will be examined in greater detail later), shown in Fig. 2. If the ambiguity group containing the good circuit is rigorously determined (as will be done in this paper), it can be seen that those faults marked as undetectable in the example belong to this ambiguity group, while others which do not belong to this ambiguity group are detectable by the two voltage measurements. Consider faults 1 and 2, which have nominal voltages that are equidistant from the nominal voltage of the good circuit. Because of sensitivity to parameter variations, fault 1 is detectable by the two voltage measurements, while fault 2 is not. Furthermore the nominal voltage of fault 4 is far from the nominal voltage of the good circuit, yet because the voltages of this circuit are highly sensitive to process parameter variations, this fault is indistinguishable from the good circuit. Hence, some faults that we will later determine to be detectable by the two voltage measurements have nominal voltages that are very close to those of the good circuit, and some faults which we will rigorously find to be undetectable have nominal voltages far from those of the good circuit. It is consequently impossible to

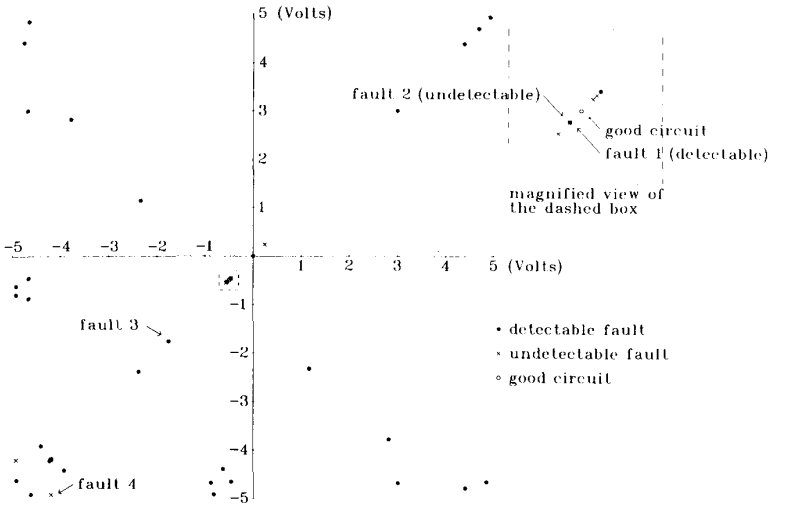


Fig. 2. Voltages at two outputs of an op amp.

identify detectable circuits by just knowing the nominal voltages, and intuition is not sufficient to aggregate ambiguity groups.

Very few results exist which aggregate faults by attempting to approximate the signature of good and faulty circuits, confining parameters within a tolerance box, rather than restricting them to their nominal values. The problem of determining signatures for linear circuits is approached rigorously in [5]. For nonlinear circuits, signatures are used to diagnose parametric faults in [6], while for catastrophic faults (which is our main concern here) one possible solution is variational analysis [7]. In variational analysis, each measurement is considered independently, and for parameters within tolerance, upper and lower bounds on each measurement are computed. In our approach, which we have outlined earlier, we consider all measurements *simultaneously*. This paper is actually the multidimensional extension of the work in [8], [9], where the problem of aggregation is analyzed for nonlinear circuits for the special case of two measurements considered simultaneously. After computing signatures for good and faulty circuits accounting for tolerance, aggregating faults into ambiguity groups can be rigorously justified and it is possible to determine an optimal set of measurements. This is also computationally feasible because the diagnosis problem has been simplified to one of diagnosing only the good circuit.

To illustrate our results, we have studied the special case of adding dc voltage measurements to wafer probe testing. Specifications obviously do not prescribe bounds on these dc voltages tests. Consequently, first bounds are determined on these measurements by the algorithm outlined above. That is, we compute the good signature. Then we present algorithms which will be used to determine an optimal set of tests. This is done essentially by determining signatures for all faulty circuits on a fault list, finding the ambiguity group containing the good circuit, and constructing a set of measurements for which this ambiguity group contains no faults. Finally the effectiveness of dc voltage tests in distinguishing faulty circuits from good one is evaluated. Case studies will show that certain types of faults can be detected by dc voltage measurements.

This paper is organized as follows. In the next section, we describe how to compute the good signature. This specifies a set of measurements for which the circuit passes tests. Obviously, the complement of this set of measurements is a set of measurements for which the circuit is discarded in wafer probe. We then define criteria for detectability, that is, a condition for which the good circuit and a faulty circuit are guaranteed not to be in the same ambiguity group, and algorithms are presented for test selection. So far the algorithm is completely general, and functional tests as well as parametric tests could be used for testing. In Section III the special case of dc voltage measurements (parametric tests) is considered and its effectiveness in detecting faulty circuits is evaluated. To do so, we first discuss the problem of fault modeling. Then we present some examples to determine the fault types which are detectable by dc voltage measurements. In the last section we conclude by summarizing our results.

A list of our notations is as follows.

| | |
|---|---|
| $\{\cdots\}$ | a set |
| $[-1, 1]$ | the closed interval on the real line between $-1$ and 1 |
| $\langle x, y \rangle$ | the inner product of $x$ and $y$ |
| co | convex hull |
| aff | affine hull |
| span $A$ | the subspace spanned by the elements of the set $A$ |
| dim | dimension |
| $N$ | the set of $\{1, 2, \cdots, n\}$ |
| $\mathbb{R}$ | the set of real numbers |
| $\mathbb{R}^n$ | the set of real $n$-tuples |
| $I^c$ | the complement of the index set $I$ in $N$ |

$\binom{n}{m}$ the number of subsets of cardinality $m$ in a set of cardinality $n$

$\operatorname{sgn} x$ $\begin{cases} +1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0 \end{cases}$

## II. Specification of the Test

### 2.1. Problem Formulation

For a given set of measurements to be added in wafer probe, our aim in this section is to determine a set of bounds for these measurements. Measurements made on a good circuit should be within these bounds. As outlined before, computing these bounds essentially involves:

1) Determine a tolerance box (specifying allowed variations of parameters).
2) Map the tolerance box by a sensitivity matrix into the measurement space, defined by the set of tests which are added in wafer probe.

To define the tolerance box, suppose we have a set of $n$ parameters: $\bar{\beta}_1, \cdots, \bar{\beta}_n$. These parameters may be process parameters as for example, oxide thickness and substrate doping; or they may be process disturbances if a statistical process simulator like FABRICSII, coupled to a circuit simulator is available [10]. At any rate, they should have minimal statistical dependence and should be constant across a chip. They can include any variable that is not controlled exactly because of fluctuations during production.

Given $e_1, e_2, \cdots, e_n$ a collection of *linearly independent* vectors in $\mathbb{R}^n$, the tolerance box, $C$, is then defined to be

$$C = \left\{ \sum_{i=1}^{n} \bar{\beta}_i e_i \,\middle|\, \bar{\beta}_i \in [\bar{\beta}_i^L, \bar{\beta}_i^H], \quad i = 1, \cdots, n \right\}$$

where the interval $[\bar{\beta}_i^L, \bar{\beta}_i^H]$ contains the nominal value $\bar{\beta}_i^0$ of $\bar{\beta}_i$.

Suppose we are given an input waveform or dc input voltages, consider the nonlinear circuit equation:

$$y = F(\bar{\beta})$$

where $y \in \mathbb{R}^{\bar{m}}$ is a vector of $\bar{m}$ measurements and $\bar{\beta} \in \mathbb{R}^n$ is a vector of $n$ parameters. Then if the parameters are within tolerance, the set of possible measurements for a good circuit is

$$R = F(C)$$

This region is the good *signature* and we will approximate it.

Since the tolerance box is a small region around nominal parameter values, $\bar{\beta}^0$, we linearize $F$ about $\bar{\beta}^0$ and approximate $R$ by

$$\bar{R} = Y + F(\bar{\beta}^0)$$

where

$$Y = \bar{A}C$$

and

$$\bar{A} = \left[ \frac{\delta F}{\delta \bar{\beta}} (\bar{\beta}^0) \right]$$

$F(\bar{\beta}^0)$ is a vector of $\bar{m}$ nominal measurements for nominal process parameters. $\bar{A}$ is a sensitivity matrix and $Y$ is a centro-symmetric polytope. This mapping is illustrated in Fig. 3, where a true signature is compared with its approximation. The example is the signature of a common-source amplifier with source degeneration. The parameters, $\beta_1$ and $\beta_2$, are normalized resistances which will be defined in the next section. $V_1$ and $V_2$ are the measurements. This example, shown in Fig. 4, will be used throughout this paper to illustrate the computations needed for our algorithm.

In the next subsection we will demonstrate how to define the intervals $[\bar{\beta}_i^L, \bar{\beta}_i^H]$ which contain the parameters $\bar{\beta}_i$ for specified statistical distributions. In the following subsection we will present our algorithm for constructing the centro-symmetric polytope, $Y$. By doing this we have defined the signature and the test. If a measurement is made which is outside of $\bar{R}$ then the circuit is definitely faulty and is discarded. On the other hand, a measurement inside $\bar{R}$ does not necessarily guarantee a good circuit because faulty circuits contained in the ambiguity group of the good circuit may have measurements in $\bar{R}$. However, we will show that if measurements are carefully chosen, many faulty circuits can be distinguished from the good one by dc voltage measurements. Hence the ambiguity group of the good circuit can be made to be small. To do so, in the last subsection we define criteria for which a good circuit is distinguishable from a faulty one for a pregiven set of measurements and we present algorithms which will be used to choose an optimal set of measurements.

### 2.2. Definition of the Tolerance Box

As mentioned in the last subsection, the tolerance box, $C$ is defined to be

$$C = \left\{ \sum_{i=1}^{n} \bar{\beta}_i e_i \,\middle|\, \bar{\beta}_i \in [\bar{\beta}_i^L, \bar{\beta}_i^H], \quad i = 1, \cdots, n \right\}$$

where $\bar{\beta}$ is a vector of $n$ parameters. If the parameters have statistical distributions which are truncated, the tolerance box can be defined to contain all possible combinations of parameters. Then the choice of $\beta_i^L$ and $\beta_i^H$ is obvious. However for typical distributions, parameters may vary from $-\infty$ to $+\infty$ (e.g., the normal distribution). In this case $\bar{\beta}_i^L$ and $\bar{\beta}_i^H$ are specified so that $C$ contains a probability, $p < 1$, usually split equally among parameters. This probability should be close to one (e.g., 0.9). Rigorously $p$ could be chosen to maximize manufacturing profit, which was maximized in [1], but this is beyond the scope of this paper.

If the probability is split equally among $n$ parameters, then the probability of the $i$th parameter being in the $i$th

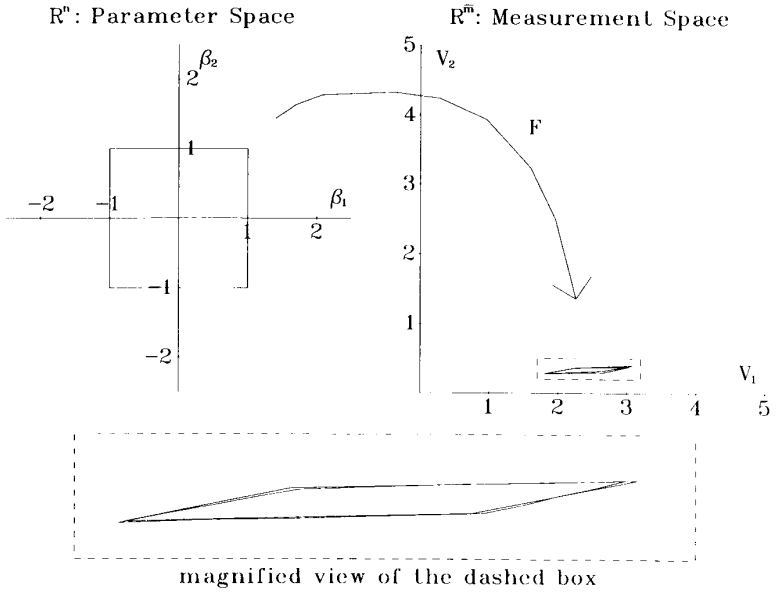


Fig. 3. The good signature and its approximation for the common-source amplifier with source degeneration.

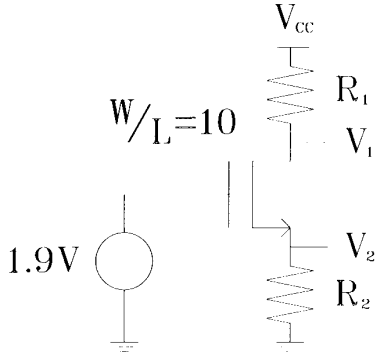


Fig. 4. Common-source amplifier with source degeneration.

interval is

$$p_i = p^{1/n}$$

$C$ is centered at nominal parameter values, $\bar{\beta}^0$. We define $\Delta\bar{\beta}_i$ to be the solution to the following equation:

$$P(\bar{\beta}_i^0 - \Delta\bar{\beta}_i \le \bar{\beta}_i \le \bar{\beta}_i^0 + \Delta\bar{\beta}_i) = p_i.$$

Then

$$\bar{\beta}_i^H = \bar{\beta}_i^0 + \Delta\bar{\beta}_i \quad \text{and} \quad \bar{\beta}_i^L = \bar{\beta}_i^0 - \Delta\bar{\beta}^i$$

We will henceforth assume that the tolerance box has been normalized to a hypercube, centered at the origin with its vertices at various combinations of $\pm 1$. The normalized variables will then be defined as

$$\beta_i = \frac{\bar{\beta}_i - \bar{\beta}_i^0}{\Delta\bar{\beta}_i}.$$

The sensitivity matrix, $\bar{A}$ must consequently also be defined for normalized variables:

$$\bar{A} = \left[\frac{\delta F}{\delta \beta}(0)\right].$$

For illustration, consider the example of a common-source amplifier with source degeneration shown in Figure 4. Suppose $\bar{\beta}_1 = R_1$ and $\bar{\beta}_2 = R_2$. Their nominal values are 7.5K and 1K, respectively. If $p = 0.9$, then $p_i = p^{1/2} = 0.95$. If both parameters have normal distributions, then $\Delta\bar{\beta}_i$ is the solution to

$$\frac{1}{\sigma_i\sqrt{2\pi}} \int_{\bar{\beta}_i^0 - \Delta\bar{\beta}_i}^{\bar{\beta}_i^0 - \Delta\bar{\beta}_i} e^{-(t-\bar{\beta}_i^0/\sigma_i)^2/2}\, dt = p_i$$

where $\sigma_i$ is the standard deviation of $\bar{\beta}_i$ and $\bar{\beta}_i^0$ is its mean value. For $p_i = 0.95$, we find that $\Delta\bar{\beta}_i = 1.94\sigma_i$. Suppose that $\sigma_1 = 0.75k$ and $\sigma_2 = 0.1k$. Then $\Delta\bar{\beta}_1 = 1.455k$ and $\Delta\bar{\beta}_2 = 0.194k$. The normalized variables are, therefore,

$$\beta_1 = \frac{\bar{\beta}_1 - 7.5k}{1.455k} \quad \text{and} \quad \beta_2 = \frac{\bar{\beta}_2 - 1k}{0.194k}.$$

### 2.3. Computing the Good Signature

In this section we will compute the good signature using the approximation

$$\bar{R} = Y + F(\beta^0).$$

Essentially the problem is one of computing the centrosymmetric polytope $Y$ which in mathematical terms is the image of a parallelpiped ($C$) under a linear transformation ($\bar{A}$). In other words,

$$Y = \bar{A}C$$

where the parallelpiped (the tolerance box)

$$C = \left\{ \sum_{i=1}^{n} \beta_i e_i \,\middle|\, \beta \in [-1, 1], \quad i \in N \right\}$$

is mapped by the linear transformation (the sensitivity matrix)

$$\bar{A} = \left[ \frac{\delta F}{\delta \beta}(0) \right]$$

to obtain $Y$.

In the following subsections, we will present an efficient algorithm which determines the image of parallelpiped under a linear transformation. The results presented here are a generalization of the work in [8], [9] where the authors considered the specific case when the range of the linear transformation is a plane. Following the presentation of our algorithm, we will conclude this section with a discussion of the algorithm's computational complexity and an example.

*2.3.1. Introduction*

Geometrically, the problem of determining $Y$ can be described in the following way. Consider a hypercube, $C$, in $\mathbb{R}^n$ centered at the origin, with the coordinates of its vertices, $v_i$, $i = 1, \cdots, 2^n$, being $\pm 1$. Under a matrix transformation $\bar{A}$, from $\mathbb{R}^n$ to $\mathbb{R}^m$, the hypercube becomes a centro-symmetric polytope $Y$, and the problem is to determine $Y$. It is easy to show that $Y$ is the convex hull [2] of the images of the $v_i$'s, i.e.,

$$Y = \text{co}\left\{ \bar{A}v_i, \quad i = 1, \cdots, 2^n \right\}$$

Thus $Y$ can be completely characterized by calculating each of the $\bar{A}v_i$'s; this however is clearly impractical. A more reasonable approach would be to directly identify either the set of vertices of $Y$, which is a subset of the set described above, or its boundary hyperplanes.

For any algorithm which computes $Y$, there are some minor technical problems when the number of rows of $\bar{A}$ is greater than the rank of $\bar{A}$. These are resolved by suitably modifying $\bar{A}$ and restricting the analysis to its range space. Hence our entire analysis will be based on the linear map $A$, which is defined as the restriction of $\bar{A}$ to its range space. Formally,

$$A: \mathbb{R}^n \to \mathbb{R}^m = \bar{A} \,|\, R(\bar{A})$$

and

$$Y = AC \subset \mathbb{R}^m.$$

Note that $m$ is less than or equal to $n$.

A possible approach to determining the vertices of $Y$ is the *random search method*. This method is based on the observation that for any direction $d$ in $\mathbb{R}^m$, the maximum projection of $Y$ on that direction is achieved at a vertex. Further, this vertex of $Y$ can be calculated by mapping into $\mathbb{R}^m$ the vertex of the hypercube whose components have the same sign as the corresponding elements of $d^T A$ (the superscript $T$ denotes transpose). Thus, by using randomly generated search directions, the vertices of $Y$ can be accumulated. The drawback of this method is that unless the number of vertices of $Y$ is known *a priori*, there is no stopping criterion. Further, it has been observed that [11] even if the total number of vertices of $Y$ is known, some of them may be hard to find since they may manifest themselves only for a narrow choice of search directions.

The two problems mentioned above, that of deciding on a search direction and knowing when to stop, are solved in what we call the *directed search method* [12], for the price of iteratively refining the convex hull of the vertices as they are determined. In this method, the vertices of $Y$ that have the maximum projection in the positive and negative coordinate directions in $\mathbb{R}^m$ are identified first. Next, the convex hull of these vertices is formed and more vertices are determined by searching in the directions that are orthogonal to each of the boundary hyperplanes. The list of vertices is then updated and the new boundary hyperplanes are determined. The process continues until there is an entire iteration in which no boundary hyperplanes are broken. The lists now contain the vertices and the boundary hyperplanes of $Y$.

With efficient updating techniques, the complexity of this algorithm is proportional to that of solving the problem in just one cycle. Thus if $v_Y$ is the number of vertices of $Y$ and $f_Y$ the number of boundary hyperplanes, the computational complexity of the directed search method can be determined by analyzing the cost of the following procedure:

1) Determine the $v_Y$ vertices of $Y$ by searching in appropriate directions.
2) Determine the convex hull of the vertices of $Y$, i.e., determine the normals to the $f_Y$ boundary hyperplanes of $Y$.
3) Search along each of the $f_Y$ directions determined in step 2, to verify that all the vertices and boundary hyperplanes have been determined.

Since a search costs $mn$ multiplications, the complexity of steps 1 and 3 is[1] $O(mn(v_Y + f_Y))$. The convex hull problem of step 2 can be solved by the "gift-wrapping" [13] approach [14], [15]. There is some doubt as to its precise worst-case complexity for general polytopes [15]. Swart [15] conjectures that it is $O(f_Y m^{m+4} \log v_Y)$. He also shows that the algorithm works best—in $O(mv_Y f_Y + m^3 f_Y \log f_Y)$ time—when the convex hull is simplicial. As will be obvious shortly, the polytope is not simplicial—however, we will not further pursue the question of the complexity of Swart's algorithm, since even if we assumed the simplicial complexity for step 2, the directed search method is inferior to our algorithm.

The algorithm that we propose is superior to the one above, since we directly determine the boundary hyperplanes of $Y$ (by operating on the columns of the matrix $A$), without worrying about any of its lower dimensional

[1]We use the notation $O(f(n))$ in the usual sense [13], with respect to the number of arithmetic operations.

faces. In order to give a geometric description of the algorithm, we initially limit our description to the case $m = 3$. For this situation, each of the columns of $A$ is a vector in $\mathbb{R}^3$. Every pair of linearly independent columns defines a plane and there can be a maximum of $n(n-1)/2$ of them, this being the case when each pair of columns defines a unique plane. For each of these planes, $P_i$, we flip the columns of $A$ that are not on the plane onto one side of it and add them up to get the point $p_i$. The planes passing through $p_i$ and $-p_i$ and parallel to $P_i$, are boundary hyperplanes of $Y$. Further, there are no other boundary hyperplanes than the ones just described.

For higher values of $m$ the intuitive explanation is essentially the same, except of course that the planes $P_i$ described above generalize to $(m-1)$-dimensional hyperplanes that are determined by sets of $(m-1)$ columns of $A$. Since $m$ is always less than or equal to $n$, the algorithm is valid for arbitrary linear transformations between arbitrary (finite!) dimensions.

In the next subsection we set up the definitions that we require for the presentation of our algorithm. Then we lay out the algorithm and illustrate the computations with a simple two-dimensional example. The formal proof of the correctness of the algorithm is in the appendix. We continue in Section 2.3.3 with a discussion of the computational complexity of the algorithm and the numerical issues that arise in its implementation. In Section 2.3.4 we return to the example of the common-source amplifier with source degeneration and we compare the true signature $R$ with its approximation $\bar{R}$.

*2.3.2. The Algorithm*

*Problem Statement:* Given $e_1, e_2, \cdots, e_n$, a collection of *linearly independent* vectors in $\mathbb{R}^n$ and a linear map $A: \mathbb{R}^n \to \mathbb{R}^m$, determine the set

$$Y = AC = \{Ax \mid x \in C\}$$

where $C$ is the parallelpiped defined as follows:

$$C = \left\{ \sum_{i=1}^{n} \beta_i e_i \,\middle|\, \beta \in [-1, 1], \quad i \in N \right\}.$$

Note that

$$C = \text{co } V$$

where

$$V = \left\{ \sum_{i=1}^{n} b_i e_i \,\middle|\, b_i \in \{-1, 1\}, \quad i \in N \right\}$$

is the set of vertices of $C$.

In order to present our algorithm, a few definitions are first necessary. The primary tool that we need is the normal to a hyperplane $H$ in $\mathbb{R}^m$. To this end, let $L$ be the subspace of $\mathbb{R}^m$ which is parallel to $H$ and has the $m-1$ basis vectors: $y_1, \cdots, y_{m-1}$. From these basis vectors, we can compute the normal, $q$, in the following way [16] ($w \in \mathbb{R}^m$ is arbitrary):

$$\langle w, q \rangle = \det \begin{bmatrix} y_1 \\ \vdots \\ y_{m-1} \\ w \end{bmatrix}.$$

The function det as defined above is an $m$-tensor on $\mathbb{R}^m$. However, the computation can be done by calculating the more familiar determinant of a matrix obtained by setting the $y_i$'s and $w$ as its columns, i.e.,

$$\langle w, q \rangle = \det [y_1 y_2 \cdots y_{m-1} w]$$

The correctness of the computation is easily verified by noting that if $w$ is any linear combination of the $y_i$'s the determinant of the resulting matrix is zero. Further, the $m$ components of the vector $q$ are the $m$ cofactors of the column $w$. Thus to determine $q$, we compute the $m$ cofactors of the column $w$. If each of these cofactors is zero, then the $b_i$'s are linearly dependent. This computation therefore, automatically verifies if a given set of $m-1$ vectors is linearly independent and if so it gives the normal to the linear subspace spanned by them.

Let $A$ denote the set of columns of $A$, i.e.,

$$A = \{a_i = Ae_i \mid i = 1, \cdots, n\}.$$

Then $Q$, the set of nonzero normals defined by $A$.[2] Formally,

$$Q = \{q \in \mathbb{R}^m \mid \langle w, q \rangle = \det [a_{j_1} \cdots a_{j_{m-1}} w], \; a_{j_i} \in A, \; i = 1, \cdots, m-1; \; w \in \mathbb{R}^m\} - \{0\}.$$

We now present our algorithm for computing the polytope $Y$ as the intersection of a set of half-spaces.

*Algorithm Y:*

INPUT $A$
Determine $Q$
For each $q \in Q_n$
  $\alpha_q = \Sigma_{i=1}^{n} (\text{sgn } \langle a_i, q \rangle) \langle a_i, q \rangle$
  OUTPUT the half spaces: $\langle y, q \rangle \leq \alpha_q$ and $\langle y, q \rangle \geq -\alpha_q$
End for

The algorithm constructs two boundary hyperplanes of $Y$ orthogonal to each $q \in Q$, one which contains the point

$$p_q = \sum_{i=1}^{n} (\text{sgn } \langle a_i, q \rangle) a_i$$

and another which contains the point $-p_q$.

As an illustration, consider the matrix:

$$A = \begin{bmatrix} 1 & 1 & 2 \\ -1 & 2 & 0 \end{bmatrix}.$$

[2]When $m = 1$, $Q$ is not defined, hence our approach is inapplicable. However in this case it is a trivial task to determine $Y$.

Obviously $m = 2$ and $n = 3$. The members of the set $A$ are the column vectors:

$$a = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad a_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad \text{and} \quad a_3 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}.$$

They are plotted in Fig. 5. We next proceed to find $Q$. This is done by finding normals to each set of $m - 1$ column vectors. However, since $m = 2$, we find normals to each of the column vectors. Therefore $Q$ has the following members:

$$q_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad q_2 = \begin{pmatrix} -2 \\ 1 \end{pmatrix}, \quad \text{and} \quad q_3 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}.$$

These are the normals to the boundary hyperplanes of $Y$. To find points through which the boundary hyperplanes pass, consider the hyperplane $H_i$, perpendicular to $q_i$ and passing through the origin. All column vectors, $a_i$, not lying in $H_i$ are flipped onto one side of $H_i$ and added up to obtain $p_{q_i}$. The points $p_{q_i}$ and $-p_{q_i}$ lie on the two boundary hyperplanes defined by the perpendicular to $q_i$. For the example we have

$$p_{q_1} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \begin{pmatrix} 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

$$p_{q_2} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} - \begin{pmatrix} 2 \\ 0 \end{pmatrix} = \begin{pmatrix} -3 \\ 1 \end{pmatrix}$$

$$p_{q_3} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} + \begin{pmatrix} 0 \\ 3 \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$$

and $\alpha_q$ is simply computed as

$$\alpha_q = \langle p_q, q \rangle.$$

Now the polytope $Y$ can be drawn and is shown in Fig. 5.

*2.3.3. Computational Complexity and Implementational Issues*

To begin our computation we first need to determine $\bar{A}$, for which one circuit simulation is required if the simulator outputs derivatives of voltages with respect to parameters. Otherwise, we require $n + 1$ simulations, where $n$ is the number of parameters. Here, $\bar{A}$ is computed by perturbation:

$$\bar{A} = [\bar{a}_{ij}] = \frac{(F^i(\bar{\beta}^j) - F^i(\bar{\beta}^{j0}))}{\bar{\beta}_j - \bar{\beta}_j^0} \Delta\bar{\beta}_j.$$

$\bar{\beta}^j$ is a vector of parameters with its $j$th component perturbed from nominal to $\bar{\beta}_j$, i.e.,

$$\bar{\beta}_i^j = \begin{cases} \bar{\beta}_i^0, & \text{if } i \neq j \\ \bar{\beta}_i, & \text{if } i = j \end{cases}$$

$\bar{\beta}_j - \bar{\beta}_j^0$ is the amount of the pertubation and $\Delta\bar{\beta}_j$ is the solution to the probability equation mentioned in Section 2.2:

$$P(\bar{\beta}_j^0 - \Delta\beta_j \leq \bar{\beta}_j \leq \bar{\beta}_j^0 + \Delta\beta_j) = p_i.$$

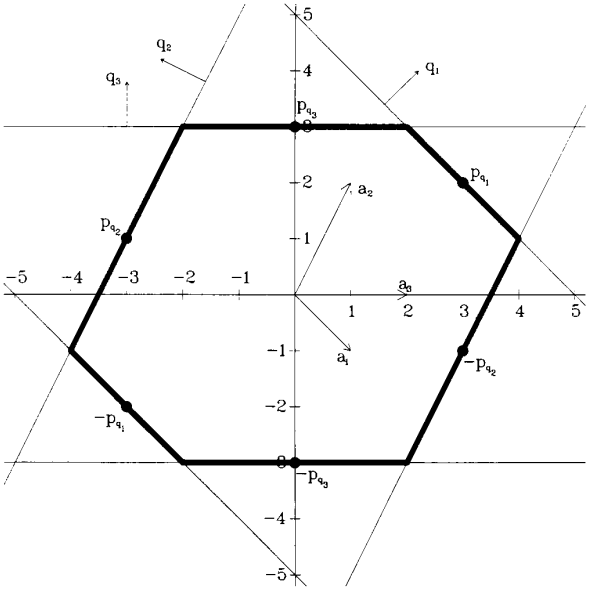


Fig. 5. The geometry of computing $Y$.

In our current implementation, we computed derivatives by perturbation and we have observed that *the computation required to determine Y is dominated by the circuit simulations necessary to obtain* $\bar{A}$.

Given $\bar{A}$, our implementation of the algorithm to compute $Y$ assumes that $\bar{m} = m$, i.e., $\bar{A} = A$. This is typically what happens in practice. The set $A$ inputted to the program is therefore merely the set of columns of the matrix $\bar{A}$ that correspond to the choice of the $e_i$'s as the basis of $\mathbb{R}^m$.

The principal computation is in determining the set $Q$. Recall from its definition that we wish to determine $q$ such that

$$\langle w, q \rangle = \det [a_{j_1} \cdots a_{j_{m-1}} w]$$

Since the $m$ components of the vector $q$ are the $m$ cofactors of the column $w$, in our initial implementation, we calculate each of these $m$ cofactors by computing the determinant of the appropriate $(m - 1) \times (m - 1)$ submatrix of $[a_{j_1} \cdots a_{j_{m-1}}]$. This is done by first computing the LU decomposition and then multiplying the diagonal elements. However, $q$ can be computed more efficiently by solving an $m \times m$ system of linear equations. Thus the complexity of determining each $q$ is $O(m^3)$.

For each $q \in Q$, we also have to compute

$$\alpha_q = \sum_{i=1}^{n} (\operatorname{sgn} \langle a_i, q \rangle) \langle a_i, q \rangle$$

This requires $mn$ multiplications. Finally, it is obvious from the definition of $Q$ that it can have at most $\binom{n}{m-1}$ elements: thus the algorithm requires $O((m^3 + mn)\binom{n}{m-1})$ multiplications and divisions.[3]

[3]For the case $m = 2$ some simplifications to this algorithm result in one that is linear in $n$ [9].

Note that in the worst case of $(m-1) = n/2$, this complexity is exponential in $n$. It appears however, that this exponential complexity is unavoidable since in this situation the number of boundary hyperplanes of $Y$ is indeed exponential in $n$. What is important is that for fixed $m$ and varying $n$, the complexity is polynomial in $n(O(n^{\tilde{m}+1})$, where $\tilde{m} = \min(m-1, n-m+1))$. This is an important feature since in our application, the quantity $m$ (the number of test points) is within our control while $n$ (the number of process parameters) is not.

It is also important to note that we do not need to do the computations for each subset of $A$ of cardinality $(m-1)$. For each $q$ that we determine, we also immediately determine the $a_i$'s orthogonal to it (represented by the set $J_q$ of Definition 2 in the appendix). All subsets of cardinality $m-1$ of this set are then not considered as possible generators of a new $q$. The quantity $O((m^3+mn)\binom{n}{m-1})$ is therefore, the worst case complexity of the algorithm.

One important strength of the algorithm that we have not developed formally is the fact that it is robust in the face of numerical errors. The polytope $Y$ varies continuously with variations in the $a_i$'s, $q$'s and $\alpha_q$'s. This is true even to the extent of a set of $a_i$'s being computed as being linearly independent when in fact, in exact arithmetic they are not. Thus even if we do not compute the "true" set $Q$ (both in terms of the number of elements and their values), the computed $Q$ is approximately the same as the "true" $Q$, and the resulting polytope $Y$ will be approximately the same as the "true" $Y$.

The above observation is one that we wish to exploit as we enhance our implementation. Our initial experience has been that we do get a fairly large number of normals $q$. A possible way to reduce the number of $q$'s and at the same time reduce the computational cost, is to generate only those that correspond to a judiciously chosen subset of the $a_i$'s. Such a subset could consist of the $a_i$'s that when normalized have the maximum projection in a predetermined set of directions that uniformly fill $\mathbb{R}^m$. For example, if $m = 2$, these directions may be along the coordinate axes and 45 degrees from each. The resulting set of $q$'s will then be a subset of $Q$. Then if an $\alpha_q$ is determined as before—with the original set of $a_i$'s—the resulting set of hyperplanes is a subset of the set of boundary hyperplanes of $Y$. Consequently the resulting polytope will encompass the "true" $Y$. Further, the larger the subset of $Q$ used, the closer the computed $Y$ comes to the "true" $Y$. This process suggests a tradeoff between the computational cost and the accuracy of approximating $Y$. In this context it is relevant that for our application, the results developed in this paper represent a linearization of a nonlinear problem and hence even the exact $Y$ is only an approximation of the physical problem.

#### 2.3.4. An Example

In Section 2.2 we defined the tolerance box for a common-source amplifier with source degeneration (Fig. 4), and normalized it to a hypercube centered at the origin. Suppose we consider dc voltage measurements at $V_1$ and $V_2$. For our choice of parameters ($R_1$ and $R_2$), the sensitivity matrix is

$$\bar{A} = \begin{bmatrix} -0.42 & 0.20 \\ -0.011 & 0.04 \end{bmatrix}$$

and $A$ is its column vectors. Then we have the normals:

$$q_1 = \begin{pmatrix} 0.01 \\ -0.42 \end{pmatrix} \quad \text{and} \quad q_2 = \begin{pmatrix} -0.04 \\ 0.20 \end{pmatrix}.$$

It turns out that $\alpha_q = 0.015$ for both vectors, $q_1$ and $q_2$. Therefore, $Y$ has the following boundary hyperplanes:

$$0.011y_1 - 0.42y_2 \le 0.015$$

$$-0.011y_1 + 0.42y_2 \le 0.015$$

$$-0.04y_1 + 0.20y_2 \le 0.015$$

$$0.04y_1 + 0.20y_2 \le 0.015.$$

The resulting centro-symmetric polytope, $Y$, is illustrated in Fig. 6.

The approximation of the signature is then

$$\bar{R} = Y + F(\bar{\beta}^0).$$

That is, $\bar{R}$ is $Y$ centered at nominal measurements:

$$F(\bar{\beta}^0) = \begin{pmatrix} 2.44 \\ 0.34 \end{pmatrix}.$$

Consequently $\bar{R}$ has the following boundary hyperplanes:

$$0.011V_1 - 0.42V_2 \le -0.10$$

$$-0.011V_1 + 0.42V_2 \le 0.13$$

$$-0.04V_1 + 0.20V_2 \le -0.015$$

$$0.04V_1 - 0.20V_2 \le 0.045.$$

$\bar{R}$ is shown in Fig. 3 together with the true signature $R = F(C)$. From this figure we see that a linear map is sufficient to approximate the signature since $\bar{R}$ and $R$ are almost indistinguishable.

### 2.4. The Choice of Measurements

#### 2.4.1. Conditions for Detectability

A set of measurements used to compute $Y$ can be dc voltages made at different nodes of a circuit or at a single node for different dc input voltages; they may be measurements made at several frequencies; or they could be measurements at various time points of a voltage waveform. A combination of all of these measurements could also be used. As mentioned in the introduction, to determine if a particular faulty circuit, say the $i$th faulty circuit, is detectable by a given set of measurements, it is not sufficient to compute the nominal faulty measurements for that circuit. To account for tolerance in good parameters, bounds on a set of possible measurements for the faulty circuit must be found. This set will be called

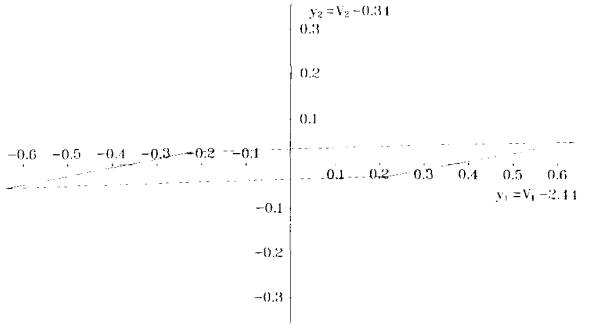


Fig. 6. $Y$ for the common-source amplifier with source degeneration.

the *fault signature*. If the circuit equation for the $i$th fault is

$$y = F_i(\bar{\beta})$$

then the $i$th fault signature is

$$R_i = F_i(C)$$

where $\bar{\beta} \in \mathbb{R}^n$ is the vector of $n$ parameters and $C$ is the tolerance box. Again, since the tolerance box is a small region around nominal parameters, a linear map is a sufficient approximation of $F_i(\ )$. We therefore have the approximation:

$$\bar{R}_i = Y_i + F_i(\bar{\beta}^0)$$

where

$$Y_i = \bar{A}_i C$$

and

$$\bar{A}_i = \left[\frac{\delta F_i}{\delta \bar{\beta}}(\bar{\beta}^0)\right].$$

The fault signatures can be computed by the same algorithm which we used to compute the good signature. But incidently, we have observed that some of the fault signatures have null interiors in $\mathbb{R}^m$. To determine $Y$, it then becomes necessary to begin our computation by determining the rank of $\bar{A}$. This has been done by computing singular values. Since $A$ is defined as

$$A = \bar{A} \mid R(\bar{A})$$

the set $A$ is the columns of $A$, expressed in terms of $e_i$'s, the basis vectors of $R(\bar{A})$.

After computing signatures of a good circuit and the $i$th faulty one, if the signatures are disjoint, i.e.,

$$\bar{R}_i \cap \bar{R}_0 = \phi$$

where $\bar{R}_i$ is the $i$th fault signature and $\bar{R}_0$ is the good signature, then *the* $i$th *fault is detectable by the given set of measurements* and they are not in the same ambiguity group.

*2.4.2. The Algorithm for Test Selection*

Based on the above criteria for detectability, an optimal set of tests will be determined. Suppose we have a list of faults. In the next section we will discuss the construction of such a list. Our aim is now to determine a minimal set of measurements that is able to distinguish as many faulty circuits as possible from the good circuit. In other words, we wish to minimize the ambiguity set which contains the good signature. To this end we will present two algorithms, and iterate between them to determine this optimal set of tests.

The first algorithm begins with a set of tests, $T$, and an ambiguity set, $\Omega$. The ambiguity set contains a list of faults that have not yet been shown to be detectable. Initially this set will contain the entire fault list. Signatures are constructed for the good circuit and each of the faulty circuits in the ambiguity set. If the $i$th fault signature does not intersect the good signature, then the $i$th fault is removed from the ambiguity set. This test is done for each fault in the ambiguity set and the resulting ambiguity set will contain only faults that are not detectable by the measurements in the list of tests. We present the algorithm below:

*Algorithm* $\Omega$*:*

INPUT $\Omega$ and $T$
Compute the good signature: $\bar{R}_0 = Y_0 + F_0(\bar{\beta}^0)$
(See Algorithm Y for computing $Y_0$)
For each $i \in \Omega$
Compute the fault signature: $\bar{R}_i = Y_i + F_i(\bar{\beta}^0)$
(See Algorithm Y for computing $Y_i$)
If $\bar{R}_i \cap \bar{R}_0 = \phi$
$\Omega = \Omega - \{i\}$
End if
End for
OUTPUT $\Omega$

The input of the second algorithm is the ambiguity set, $\Omega$, and the set of possible tests, $T^c$, that are not in the list of tests to be used, $T$. This algorithm selects the next measurement to be added to the list of tests. This is done by a simple heuristic which determines the measurement which deviates the most from the good measurement for all faults listed in the current ambiguity set. First, measurements are computed for the good circuit. Then they are computed for each faulty circuit in the ambiguity set. For the possible test $t$, $\Delta y_i^t$ is the difference between the $i$th faulty measurement and the good measurement. We sum $\Delta y_i^t$ for each measurement, $t$, to get $s_t$. The sums are suitably weighted to account for differences in units and the measurement where the sum is largest is chosen as the next test point. We now present the algorithm.

*Algorithm T:*

INPUT $\Omega$, $T^c$, and parameter $\epsilon$
For each $t \in T^c$
$s_t = 0$
For each $i \in \Omega$
Compute $\Delta y_i^t$

$$s_t = \begin{cases} s_t + \Delta y_i^t & \text{if } \Delta y_i^t < \epsilon \\ s_t = s_t + \epsilon & \text{if } \Delta y_i^t \geq \epsilon \end{cases}$$

End for
End for
OUTPUT $t_{opt} = \arg\max_t s_t$

To determine an optimal set of tests, we begin with an ambiguity set, $\Omega$, that contains all of the faults in the fault list and a list of tests which is empty. We initially use Algorithm T to choose the first test. This test is added to the list of tests. Now the list of tests has a single measurement and 1-dimensional signatures are constructed by Algorithm $\Omega$. In this algorithm we determine whether or not each of the fault signatures intersects with the good one; and if a fault signature and a good signature are disjoint, the fault is dropped from the ambiguity set. Algorithm $\Omega$ determines an ambiguity set which contains only faults that are not detectable by the first test. Then Algorithm T is used to choose the next test based on the new ambiguity set. This measurement is added to the list of tests. Algorithm $\Omega$ constructs two-dimensional signatures and reduces the ambiguity set to contain only faults that are not detectable by these two tests. The next test is then selected. This process continues until the ambiguity set is empty or we feel that we have sufficient fault coverage, $f < 1$. Suppose the desired fault coverage is $f^*$. Then we have the main algorithm:

*The Main Algorithm:*
INPUT $\Omega$ and $T^c$
$f = 0$ and $T = \phi$
Do while $f < f^*$
  Choose the next test and add it to $T$ (Algorithm T)
  Reduce $\Omega$ to contain only faults not detected by $T$ (Algorithm $\Omega$)
  Determine $f$
End do

### 2.4.3. The Example

For illustration, let us return to the example of the common-source amplifier with source degeneration. Suppose this circuit has the following specifications:

1) Magnitude of ac gain $>2.5$;
2) $-3$-dB frequency $>2$ GHz;
3) Small signal output resistance $<7$k.

Our aim is to show that many faulty circuits can be detected without testing any of these specifications. Instead measurements of dc voltages will be used. The possible test points are therefore dc measurements of $V_1$ and $V_2$.

Suppose we have the following faults:

1) Gate–drain short;
2) $W/L = 3$;
3) Drain–source short;
4) $R_2 = 0.5k$.

For measurements at $V_1$ and $V_2$, the two-dimensional fault signatures of each of these circuits are shown in Fig. 7,

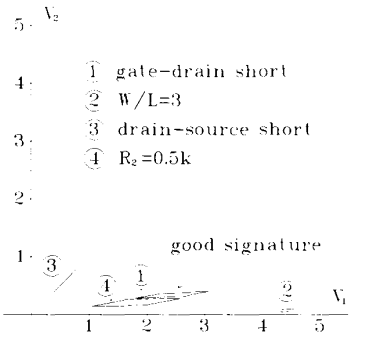


Fig. 7. Good and fault signatures for the common-source amplifier with source degeneration.

together with the good signature. The first test point to be selected is $V_1$, displayed on the horizontal axis. To visualize the one-dimensional signatures, think of these signatures projected along the horizontal axis. We see that the second and third one-dimensional fault signatures do not intersect the good signature and hence these faults can be detected by this test. Then $V_2$ is added to the list of tests and we see that in addition the fourth fault can be detected by dc voltage tests. So we have that three out of four faults can be detected without performing *any* specification tests. In the next section we will see how general this result is.

## III. Fault Coverage of DC Voltage Tests

### 3.1. Fault Modeling

To evaluate our results, in this section we apply our algorithm for test selection to some case studies. For illustration we will limit the set of possible measurements to tests of dc voltages, for a single dc input voltage.

To be able to judge the effectiveness of dc voltage tests, an accurate fault list is needed. For analog circuits, faults can be classified into two categories:

1) Catastrophic faults;
2) Parametric faults.

Catastrophic faults are random defects that cause failures in various components. They may be structural deformations like short and open circuits, or cause large variations in design parameters (e.g., a change in a single transistor's length–width ratio caused by a dust particle on a photolithographic mask). Parametric faults are caused by statistical fluctuations in the manufacturing environment. To define this fault type rigorously, consider a set of process parameters (e.g., oxide thickness and substrate doping), assumed to be constant across a chip but to vary between chips. If the parameters are modeled by unbounded statistical distributions, which is typically the case (e.g., normal distribution), they may vary from $-\infty$ to $+\infty$ and there will be values of parameters for which circuit specifications are not satisfied. The *acceptability region* is defined to be the set of parameters for which the circuit satisfies all specifications. By definition then, the

sets of values of process parameters outside of the acceptability region cause parametric faults. Among such faults are also faults caused by process gradients which produce device mismatch. Since design parameters (catastrophic faults) and process parameters (parametric faults) can take on infinitely many values, there are infinitely many analog faults. We must consequently choose a subset which will lead to the best possible fault list.

Our case studies involve CMOS processes. One approach to fault modeling for CMOS processes is suggested in [17]. Based on the observation that for digital circuits, yield losses in CMOS processes are primarily due to catastrophic faults[4] [17], [18] and that multiple faults are unlikely, this technique generates single catastrophic defects using statistical data from the fabrication process. Three physical defects are implemented as missing or extra material in a given layer in the layout. They are extracted to the circuit-level and cause shorts and breaks in interconnections, new devices, and changes in the behavior of existing devices. Because this methodology requires a large sample of defects to create an accurate fault list and predict fault coverage, for a preliminary study of our application which requires a dc simulation for each fault on the fault list, the simulation time may be excessive.

We have, therefore, chosen to implement an alternative fault list based on empirically observed circuit-level failures. A possible fault model for CMOS circuits which is suggested in [19] contains open circuits in the diffusion and metallization layers and short circuits between adjacent diffusions and metallizations. A more complete model is indicated in [20], where Table I is presented. Unfortunately no indication is given as to the overall likelihood of each of the listed fault types. This is necessary to approximate accurately fault coverage. In the preliminary test of our algorithm we have nevertheless chosen to implement the more common device failures listed in Table I—gate to drain short, gate to source short, drain contact open, and source contact open.

It should be noted that all of the previously mentioned studies have restricted fault lists to catastrophic faults. This is because, as stated in [18], for digital circuits catastrophic faults dominate. There is some controversy as to whether or not this is the case for analog circuits. In [21], it is claimed that 80–90 percent of analog faults involve shorted and open resistors, capacitors, diodes and transistors. In [22], on the other hand, it is argued that this may not necessarily be the case, and instead yield losses in analog circuits are caused by multiple phenomena. In particular, yield losses due to catastrophic faults appear to be insignificant for bipolar IC's [22]. Consequently because of our selection of fault model, in this study we will first evaluate the detectability of catastrophic faults by dc voltage tests using some case studies. Then we will discuss the detectability of parametric faults.

[4]In [17] it is argued that parametric faults are easily detected and therefore are not primary targets of production testing, and in [18] it is stated that 83.5-percent of all chip failures are due to catastrophic faults.

TABLE I
LIKELY CMOS FAULTS

| Class | Device failures | Interconnect failures |
|---|---|---|
| I. Most likely | Gate to drain short.<br>Gate to source short. | Short between diffusion lines. |
| II. Less likely | Drain contact open.<br>Source contact open. | Aluminum polysilicon cross-over broken. |
| III. Least likely | Gate to substrate short.<br>Floating gate. | Short between Aluminum lines. |

### 3.2. *Case Studies*

We have applied our algorithm for determining an optimal set of test points and evaluating fault coverage to two CMOS analog subnetworks. The parameters which define the tolerance box are from the process-oriented CSIM [23] model. They have been assumed to form a statistically independent set and are constant across a chip. These are:

1) Oxide thickness;
2) Substrate doping concentration;
3) Surface mobility;
4) Flatband voltage;
5) Lateral junction depth;
6) Lithographic offset in polywidth;
7) Lithographic offset in diffusion line.

These parameters have been chosen because of being most sensitive for typical designs [23]. Since they are independent for n- and p-type material (except for oxide thickness), for a CMOS process we have a total of 13 process parameters. Limiting the process parameter list to a set with minimal statistical correlation will lead to more realistic testability results by minimizing the size of the signatures.

It has been assumed that faulty circuits contain a single fault. Therefore for each transistor in the circuit, there are four faults in the fault list:

1) Gate to drain short;
2) Gate to source short;
3) Drain contact open;
4) Source contact open.

The CENTER/ADVICE system [24], [25] in which the algorithm has been implemented has facilities for varying parameter values and circuit topology. The fault model has, therefore, been implemented by replacing each transistor by a transistor surrounded by switches, as shown in Fig. 8. Via CENTER, a fault circuit can be obtained from the good one by opening or closing the appropriate switch.

One of the analog subnetworks that was studied was an op amp with 114 faults in the fault list and 22 nodes. The other was a low-pass filter with 52 faults in the fault list and 9 nodes. Our principle findings are presented in Table II.

Clearly a high percentage of catastrophic faults can be detected by very few test points. It also appears that in

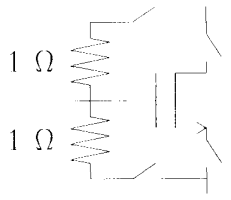


Fig. 8. Fault model implementation.

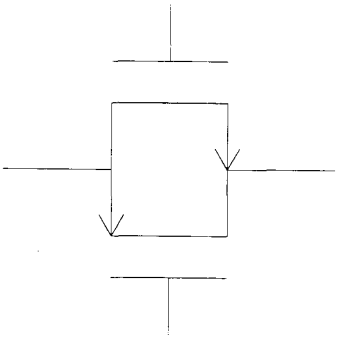

Fig. 9. Circuit fragment.

TABLE II
EXPERIMENTAL RESULTS FOR AN OP AMP AND A LOW-PASS FILTER

| | OP AMP | LOW-PASS |
|---|---|---|
| Number of faults in the fault list | 114 | 52 |
| Number of circuit nodes | 22 | 9 |
| Number of primary outputs | 2 | 1 |
| Percentage of faults detected by primary outputs | 81% | 40% |
| Percentage of faults detected by 20% of the nodes | 99% | 63% |
| Percentage of faults detected by all of the nodes | 100% | 75% |
| Number of test points needed for maximum fault coverage | 7 | 5 |

practice it is best to settle for less than the maximum fault coverage obtainable by dc voltage tests since a few carefully selected test points detect most faulty circuits. It is also interesting that in the case of the low-pass filter, dc measurements at all nodes can detect only 75-percent of the faults. One reason for this is that the filter has circuit fragments as shown in Fig. 9. It turns out that at the dc operating point, the two transistors do not conduct; hence open circuits at the common source and drain cannot be detected by dc testing. Similarly, capacitors may cause open circuit faults to be undetectable by dc voltage tests. This suggests that in most circuits, even though many faults are detectable by dc tests, some additional dynamic tests are necessary.

### 3.3. A Note of the Detectability of Parametric Faults

Since the results of our case studies have been based on a fault list containing only catastrophic faults, our result is that most catastrophic faults can be detected by dc voltage tests. Intuitively this is because the circuit's behavior is highly distorted at least at a local point and a test point close to this point will detect the fault.

Suppose the parameters of a particular circuit have truncated statistical distributions and the tolerance box has been defined to contain 100 percent of the parameters. Furthermore suppose that the circuit has been designed to satisfy specifications when the parameters are within the tolerance box. This may be verified by variational analysis [7]. Then there are no values of parameters for which the circuit is faulty, and hence there are no parametric faults. To test such a circuit, *it is sufficient to find a set of tests that detects catastrophic faults.*

This is typically not the case. Parameters may vary from $-\infty$ to $+\infty$ and there may be values of parameters for which the circuit does not satisfy specifications. By definition, parameters outside of the acceptability region cause parametric faults, and the probability of obtained parameters in the acceptability region is the *parametric yield* of a circuit. The acceptability region does not coincide with the tolerance box, and hence parametric faults may be caused by parameters inside or outside of the tolerance box. See for example Fig. 10 for the common-source amplifier with source degeneration. Obviously there are values of parameters contained in the tolerance box and not contained in the acceptability region.

Since we are mapping the tolerance box into the measurement space to define the signature, dc voltage measurements for circuits with parameters in the tolerance box and outside of the acceptability region will be in the good signature. The circuits with such parameters will pass dc voltage tests. Consequently some parametric faults cannot be detected by our dc voltage testing procedure as implemented. It may be difficult to detect these faults by a computationally efficient dc testing procedure due to the nonlinear nature of the problem and since these faults tend to cause smaller deviations in dc voltage measurements.

It is also clear from Fig. 10 that there may be values of parameters contained in the acceptability region but not in the tolerance box. These parameters correspond to good circuits which will fail dc voltage tests because they are outside of the tolerance box. Note that in Section 2.2 we defined the tolerance box to contain a probability, $p$, of all parameters, and we suggested using $p = 0.9$. Under these conditions, when in the worst case the acceptability region is unbounded in all directions, at most 10 percent of the good circuits will fail tests. This, however, is not the case in practice because the acceptability region is usually bounded and, therefore, if $p = 0.9$, much less than 10 percent of the good circuits will fail tests. Consider, for example, the common-source amplifier with source degeneration. It can also easily be determined by a Monte Carlo-based analysis (which will be elaborated on later) that the parametric yield of this circuit is 75 percent and that 4.5 percent of the *good* circuits fail dc voltage tests because they have parameters outside of the tolerance box.

It is possible to reduce the fraction of good circuits failing tests. To do so, the tolerance box may be enlarged to contain a higher percentage of parameters. For example, if we were to redefine the tolerance box to contain 95 per-

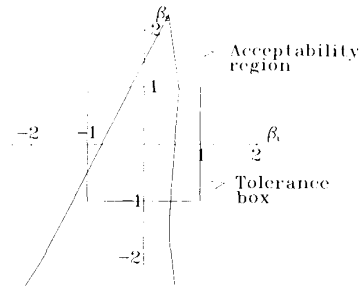


Fig. 10. The tolerance box and acceptability region for the common-source amplifier with source degeneration.

cent of all parameters, then only 2.0 percent of the good circuits would fail dc voltage tests.

It is not necessary to guarantee that no good circuits fail dc voltage tests. On the contrary, as explained in [1], since packaging is expensive, if we want to optimize manufacturing profit, it is sometimes better to fail some good circuits in wafer probe if this simplifies packaging and testing. Nevertheless at least an intuitive and computationally cheap guideline is needed for the assignment of tolerance.

A Monte Carlo simulation, used to determine the number of good circuits failing tests, would be computationally prohibitive and jeopardize the efficiency of our technique. A better approach would be to approximate the acceptability region, possibly along the lines suggested in [26]. Then the number of good circuits failing tests can be determined by a Monte Carlo analysis in the parameter space. This is computationally cheap since we do not require any function evaluations (simulations). Thus, using this approach, the tolerance box can be chosen at the beginning of the analysis so that few good circuits fail tests.

## IV. Conclusions

We have proposed a method of constructing a set of simple tests which detects many faulty circuits before specification testing in the wafer probe stage of production. By adding these tests, we avoid performing expensive specification tests on faulty circuits. In particular we have shown that simple parametric tests used before wafer probe can be effective in detecting faulty circuits containing catastrophic faults and it would be wasteful and expensive to subject these chips to complete specification tests.

To choose the tests to be used, an original algorithm has been presented. This algorithm exceeds those in the diagnosis literature by exploiting a simplification of the problem. That is, typically the diagnosis literature aims to find algorithms which identify arbitrary faulty elements, and in this paper, we have found it necessary just to identify the good circuit. With this simplification, we have found it computationally feasible to compute signatures of good and faulty circuits accounting for tolerance in process parameters. By doing so, we can rigorously justify our aggregation of ambiguity groups. Then to choose an optimal set of tests, it is simply necessary to minimize the ambiguity group containing the good circuit, and an algorithm which does this has been applied successfully to several case studies.

## Appendix
## The Proof of Algorithm $Y$

In this appendix we will reiterate the problem statement, set up the definitions we require for our mathematical analysis, and prove a theorem that establishes the correctness of our algorithm for computing $Y$. The proof of our algorithm essentially consists of two parts—one which establishes that each of the hyperplanes is indeed a boundary hyperplane of $Y$ (Lemma 5), and another which proves that there are no other boundary hyperplanes of $Y$ (Lemmas 2 and 4). This appendix culminates in a single theorem which sums up these results, verifying that our algorithm correctly determines the polytope $Y$.

### A.1. Preliminaries

*Problem Statement:* Given $e_1, e_2, \cdots, e_n$, a collection of *linearly independent* vectors in $\mathbb{R}^n$ and a linear map $\boldsymbol{A}: \mathbb{R}^n \to \mathbb{R}^m$, determine the set

$$Y = \boldsymbol{A}C = \{\boldsymbol{A}x \mid x \in C\} \tag{1.1}$$

where $C$ is the parallelpiped defined as follows:

$$C = \left\{ \sum_{i=1}^{n} \beta_i e_i \mid \beta \in [-1, 1], \quad i \in \boldsymbol{N} \right\}. \tag{1.2}$$

Note that

$$C = \text{co } V \tag{1.3}$$

where

$$V = \left\{ \sum_{i=1}^{n} b_i e_i \mid b_i \in \{-1, 1\}, \quad i \in \boldsymbol{N} \right\} \tag{1.4}$$

is the set of vertices of $C$.

Formally, let $A$ denote the set of images of the $e_i$'s under $\boldsymbol{A}$, i.e.,

$$A = \{a_i = \boldsymbol{A}e_i \mid i = 1, \cdots, n\}.$$

We then have the crucial definition of the set of nonzero normals defined by $A$.

*Definition 1:* We denote by $Q \subset \mathbb{R}^m$, the set of nonzero normals defined by $A$, i.e.,

$$Q = \{q \in \mathbb{R}^m \mid \langle w, q \rangle = \det [a_{j_1} \cdots a_{j_{m-1}} w], \; a_{j_i} \in A, \; i = 1, \cdots, m-1; \; w \in \mathbb{R}^m\} - \{0\}.$$

*Definition 2:* For each $q \in Q$, we define an *index set corresponding to* $q$ as follows:

$$J_q = \{i \in \boldsymbol{N} \mid \langle a_i, q \rangle = 0\}.$$

*Definition 3:* Given $f$, a $k$-dimensional face of $C$ ($k \in \boldsymbol{N}$) which may be described as

$$f = \left\{ \sum_{i \in I_f} \beta_i e_i + \sum_{i \in I_f^c} b_i e_i \mid \beta_i \in [-1, 1], \quad i \in I_f \right\}$$

where for all $i \in I_f^c$, $b_i \in \{-1, 1\}$, the set $I_f$ which is a subset of $N$ of cardinality $k$, is called the *index set corresponding to f*.

*Definition 4:* A normal $q \in Q$ (with index set $J_q$) and a face $f$ of $C$ (with index set $I_f$) are said to correspond to each other if $J_q = I_f$.

*Definition 5:* Given $l$ vertices of $C$,

$$v_j = \sum_{i=1}^{n} b_i^j e_i, \qquad j = 1, \cdots, l$$

where for all $i$ and $j$, $b_i^j \in \{-1, 1\}$, *the minimal face* containing $\{v_j, j = 1, \cdots, l\}$ is the unique face of $C$ which contains $\{v_j, j = 1, \cdots, l\}$ and whose index set $I$ is given by

$$I = \left\{ i \in N \middle| b_i^j \neq b_i^k, \quad j, k \in \{1, \cdots, l\} \right\}.$$

We conclude this section with the following obvious fact which is important for the development of our results.

*Fact 1:* The indices corresponding to any $(m-1)$ linearly independent $a_i$'s are a subset of some $J_q$. Consequently, for any $q \in Q$, the set

$$A_q = \{a_i \in A \mid i \in J_q\}$$

has exactly $(m-1)$ linearly independent vectors.

*A.2 The Proof*

As mentioned before, our algorithm constructs two boundary hyperplanes of $Y$ orthogonal to each $q \in Q$, one which contains the point

$$p_q = \sum_{i \in J_q^c} (\operatorname{sgn} \langle a_i, q \rangle) a_i$$

and another which contains the point $-p_q$. We will prove the correctness of our algorithm with the aid of three principal lemmas. In Lemma 2 we establish that every boundary hyperplane of $Y$ is orthogonal to some $q \in Q$, while in Lemma 4 we prove that $Y$ has exactly two boundary hyperplanes orthogonal to each $q \in Q$. The algorithm therefore enumerates $2|Q|$ ($|Q|$ is the cardinality of $Q$) hyperplanes in $\mathbb{R}^m$ and we prove in Lemma 5, that each of these hyperplanes is a boundary hyperplane of $Y$. We thus establish that our algorithm does indeed determine the polytope $Y$.

*Lemma 1:* $Y$ is an $m$-dimensional polytope and each of its vertices is the image under $A$ of a vertex of $C$.

*Proof:* By (1.1), (1.3), and (1.4),

$$Y = \text{co} \{Av \mid v \in V\}.$$

Thus $Y$ is a polytope and each of its vertices is the image under $A$ of a vertex of $C$. Further, due to (2.1),

$$Y = \left\{ \sum_{i=0}^{n} \beta_i a_i \middle| \beta_i \in [-1, 1] \; i \in N \right\}.$$

Therefore,

$$\text{aff } Y = \text{span } A.$$

Since the $e_i$'s are linearly independent and $m$ is defined as the dimension of the range space of $A$, it follows that

$$\dim (\text{span } A) = m$$

and hence $y$ is $m$-dimensional. ■

*Lemma 2:* Every boundary hyperplane of $Y$ is orthogonal to some $q \in Q$.

*Proof:* Let $f_Y$ be an $(m-1)$-dimensional face of $Y$. Since $f_Y$ is a polytope and each of its vertices is a vertex of $Y$, due to Lemma 1, there exist vertices of $C$,

$$v_j = \sum_{i=1}^{n} b_i^j e_i, \qquad j = 1, \cdots, l$$

such that

$$f_Y = \text{co} \{Av_j, \quad j = 1, \cdots, l\}.$$

Let $I_f$ be the index set of the minimal face containing $\{v_j, j = 1, \cdots, l\}$, i.e.,

$$f_Y = \text{co} \left\{ \sum_{i \in I_f} b_i^j a_i, \quad j = 1, \cdots, l \right\} + \bar{y}$$

where

$$\bar{y} = \sum_{i \in I_f^c} b_i a_i$$

with

$$b_i^j = b_i^k = b_i \quad \text{for all} \quad i \in I_f^c, \; j, k \in \{1, \cdots, l\}.$$

Now note that the polytope

$$Y_f = \left\{ \sum_{i \in I_f} \beta_i a_i \middle| \beta_i \in [-1, 1] \right\} + \bar{y} \tag{2.1}$$

which is the image of the minimal face containing $\{v_j, j = 1, \cdots, l\}$, contains all the vertices of $f_Y$, hence

$$f_Y \subset Y_f \subset Y$$

and since $f_Y$ is $(m-1)$-dimensional

$$\dim Y_f \geq (m-1). \tag{2.2}$$

Let

$$H: \langle y, t \rangle = \gamma$$

denote the boundary hyperplane which is the affine hull of $f_Y$. Since $f_Y$ is a face, $Y_f$ lies in a closed half-space bounded by $H$ and without loss of generality

$$\langle y, t \rangle \leq \langle w, t \rangle = \gamma, \qquad \text{for all } y \in Y_f, \; w \in f_Y$$

Thus, for arbitrary $k \in \{1, \cdots, l\}$

$$\left\langle \sum_{j \in I_f} \beta_j a_j + \bar{y}, t \right\rangle \leq \left\langle \sum_{j \in I_f} b_j^k a_j + \bar{y}, t \right\rangle, \qquad \text{for all } \beta_j \in [-1, 1], j \in I_f.$$

Since,

$$\max_{\beta_j \in [-1,1], j \in I_f} \left\langle \sum_{j \in I_f} \beta_j a_j, t \right\rangle = \sum_{j \in I_f} \operatorname{sgn} (\langle a_j, t \rangle) \langle a_j, t \rangle$$

it follows that

$$\sum_{j \in I_f} \left( b_j^k - \operatorname{sgn} \left( \langle a_j, t \rangle \right) \right) \geq 0$$

which implies that

$$\sum_{j \in I_f} \left[ b_j^k \operatorname{sgn} \left( \langle a_j, t \rangle \right) - 1 \right] \left[ \left( \operatorname{sgn} \langle a_j, t \rangle \right) \langle a_j, t \rangle \right] \geq 0.$$

Since for all $j$, the first term (in square brackets) is always less than or equal to zero, and since $k \in \{1, \cdots, l\}$ was arbitrary, the above inequality can be satisfied only by an equality and that too only if for each $j \in I_f$ the product of the two terms is zero, i.e., either

$$b_j^k = \operatorname{sgn} \langle a_j, t \rangle, \qquad \text{for all } k = 1, \cdots, l$$

or

$$\langle a_j, t \rangle = 0.$$

However, by the definition of the minimal face, for any $j \in I_f$, and $b_j^k$'s cannot be equal, hence

$$\langle a_j, t \rangle = 0, \qquad \text{for all } j \in I_f \tag{2.3}$$

Now from (2.1), aff $Y_f$ is parallel to span $\{a_j \mid j \in I_f\}$ and from (2.2) and (2.3), dim $Y_f = (m - 1)$, and

$$\operatorname{aff} Y_f = \operatorname{aff} f_Y = H.$$

Finally, since there are exactly $(m - 1)$ linearly independent vectors in the set

$$\{a_j \mid j \in I_f\}$$

by Fact 1, there exists $q \in Q$, such that $H$ is orthogonal to $q$. ■

*Lemma 3:* Given $q \in Q$, any face of $C$ corresponding to $q$ maps into an $(m - 1)$-dimensional polytope that is orthogonal to $q$.

*Proof:*

$$f = \left\{ \sum_{i \in J_q} \beta_i e_i + \sum_{i \in J_q^c} b_i e_i \mid \beta_i \in [-1, 1], \quad i \in J_q \right\},$$
$$b_i \in \{-1, 1\}$$

is a face of $C$ corresponding to $q$. The image of $f$ under $\boldsymbol{A}$ is

$$P = \boldsymbol{A}f = \left\{ \sum_{i \in J_q} \beta_i a_i + \sum_{i \in J_q^c} b_i a_i \mid \beta_i \in [-1, 1] \right\},$$
$$b_i \in \{-1, 1\}.$$

$P$ is a polytope since it is the image of a polytope and it is $(m - 1)$-dimensional and orthogonal to $q$ since it is parallel to

$$\operatorname{span} \{a_i \mid i \in J_q\}$$

which by Fact 1 is $(m - 1)$-dimensional. ■

*Lemma 4:* $Y$ has exactly two boundary hyperplanes orthogonal to each $q \in Q$.

*Proof:* Since $Y$ is an $m$-dimensional polytope in $\mathbb{R}^m$ it has a nonempty interior [2]. Hence there are exactly two distinct hyperplanes $H^+$ and $H^-$, orthogonal to each $q$ that support $Y$. Without loss of generality we will consider only $H^+$. $H^+$ must contain $v_Y$ a vertex of $Y$. Let $v_C$ be a vertex of $C$ that maps into $v_Y$. Since the $e_i$'s are linearly independent, there is exactly one face, $f_C$ corresponding to $q$ (index set $J_q$) that contains $v_C$. By Lemma 3, $f_C$ maps into $f_Y$ an $(m - 1)$-dimensional polytope that is orthogonal to $q$. Further since $v_Y \in f_Y$,

$$\operatorname{aff} f_Y = H^+$$

i.e., $H^+$ is a boundary hyperplane of $Y$. ■

*Lemma 5:* Each of the hyperplanes $\langle y, q \rangle = \alpha_q$ and $\langle y, q \rangle = -\alpha_q$ determined by the algorithm is a boundary hyperplane of $Y$.

*Proof:* Consider the face of $C$,

$$f_q = \left\{ \sum_{i \in J_q} \beta_i e_i + \sum_{i \in J_q^c} b_{qi} e_i \mid \beta \in [-1, 1], \quad i \in J_q \right\}$$

corresponding to an arbitrary $q \in Q$, where for all $i \in J_q^c$

$$b_{qi} = \operatorname{sgn} \langle a_i, q \rangle.$$

Note that,

$$\boldsymbol{A}f_q = \left\{ \sum_{i \in J_q} \beta_i a_i + \sum_{i \in J_q^c} b_{qi} a_i \mid \beta_i \in [-1, 1], \quad i \in J_q \right\}.$$

Consider $w \in \boldsymbol{A}f_q$ and note from the definition of $\alpha_q$ in the algorithm that

$$\langle w, q \rangle = \alpha_q.$$

By Lemma 3, $\boldsymbol{A}f_q$ is an $(m - 1)$-dimensional polytope, hence

$$\operatorname{aff} \boldsymbol{A}f_q = H_q \tag{2.4}$$

where $H_q$ is the hyperplane

$$\langle y, q \rangle = \alpha_q.$$

Now consider an arbitrary

$$y \in Y = \boldsymbol{A}C$$
$$= \left\{ \sum_{i \in J_q} \beta_i a_i + \sum_{i \in J_q^c} \beta_i a_i \mid \beta_i \in [-1, 1], \quad i \in N \right\}$$

and note from the definition of $J_q$ and $b_{qi}$ that

$$\langle y, q \rangle = \sum_{i \in J_q} \beta_i \langle a_i, q \rangle + \sum_{i \in J_q^c} \beta_i \langle a_i, q \rangle$$
$$\leq \sum_{i \in J_q^c} b_{qi} \langle a_i, q \rangle = \alpha_q. \tag{2.5}$$

If follows from (2.4) and (2.5) that $H_q$ is a boundary hyperplane of $Y$. We can similarly establish that the hyperplane defined by

$$\langle y, q \rangle = -\alpha_q$$

is also a boundary hyperplane. ■

*Theorem:* $Y$ is precisely the polytope defined as the intersection of the half-spaces outputted by the algorithm.

*Proof:* Due to Lemmas 2, 4, and 5, the boundary hyperplanes of $Y$ are precisely the ones described in Lemma 5. Further the direction of the inequalities in the algorithm are correct since $Y$ contains the origin in $\mathbb{R}^m$ (it being the image of the origin in $\mathbb{R}^n$ which belongs to $C$). ■

## Acknowledgment

The first author would like to thank Prasad Subramaniam, Reddy Penumalli, Hao Nham, and Alberto Sangiovanni-Vincentelli for providing the opportunity to work on this problem at Bell Labs.

## References

[1] W. Maly and Z. Pizlo, "Tolerance assignment for IC selection tests," *IEEE Trans. Computer-Aided Design*, vol. CAD-4, April 1985.
[2] R. T. Rockafellar, *Convex Analysis*, Princeton, NJ: Princeton Univ., 1970.
[3] W. Hochwald and J. D. Bastian, "A DC approach for analog fault dictionary determination," *IEEE Trans. Circuits Syst.*, vol. CAS-26, pp. 523-529, July 1979.
[4] R. E. Tucker and L. P. McNamee, "Computer-aided design application to fault detection and isolation techniques," in *Proc. Int. Symp. Circuits and Systems*, 1977, pp. 684-687.
[5] A. Pahwa and R. A. Rohrer, "Band faults: Efficient approximations to fault bands for the simulation before fault diagnosis of linear circuits," *IEEE Trans. Circuits Syst.*, vol. CAS-29, pp. 81-88, Feb. 1982.
[6] A. J. Strojwas and S. W. Director, "A pattern recognition based method for IC failure analysis," *IEEE Trans. Computer-Aided Design*, vol. CAD-4, pp. 76-92, Jan. 1985.
[7] V. Visvanathan, "Variational analysis of integrated circuits," in *Proc. Int. Conf. Computer-Aided Design*, Santa Clara, pp. 228-231, Nov. 1986.
[8] W. C. Wu, "A robust simulation-based-test technique for nonlinear analog fault diagnosis," M.S. thesis, Univ. of Maryland, 1984.
[9] V. Visvanathan, E. Szeto, and A. Tits, "A robust simulation-before-test technique for DC analog fault diagnosis," in *Proc. Int. Symp. Circuits and Systems*, Montreal, Canada, pp. 689-692, 1984.
[10] S. R. Nassif, A. J. Strojwas, and S. W. Director, "FABRICS II—A statistically based IC fabrication process simulator," *IEEE Trans. Computer-Aided Design*, vol. CAD-3, pp. 40-46, Jan. 1984.
[11] A. L. Tits, private communication.
[12] E. Grosse, private communication.
[13] D. T. Lee and F. P. Preparata, "Computational geometry—A survey," *IEEE Trans. Computers*, vol. C-33, pp. 1072-1101, Dec. 1984.
[14] D. R. Chand and S. S. Kapur, "An algorithm for convex polytopes," *J. Ass. Comput. Mach.*, vol. 17, no. 1, pp. 78-86, Jan. 1970.
[15] G. Swart, "Finding the convex hull facet by facet," *J. Algorithms*, vol. 6, no. 1, pp. 17-48, Mar. 1985.
[16] M. Spivak, *Calculus on Manifolds*. New York: Benjamin, 1965.
[17] J. P. Shen, W. Maly, and F. J. Ferguson, "Inductive fault analysis of MOS integrated circuits," *IEEE Design and Test*, vol. 2, pp. 13-26, Dec. 1985.
[18] C. Stapper, F. Armstrong, and K. Saji, "Integrated circuit yield statistics," *Proc. IEEE*, vol. 71, pp. 453-470, Apr. 1983.
[19] J. Galiay, Y. Crouzet, and M. Vergniault, "Physical versus logical fault models in MOS LSI circuits: impact on their testability," *IEEE Trans. Computers*, vol. C-29, pp. 527-531, June 1980.
[20] P. Banerjee and J. A. Abraham, "Fault characterization of VLSI MOS circuits," in *Proc. IEEE Int. Conf. Circuits and Computers*, New York, pp. 564-568, Sept.-Oct. 1982.
[21] Q. F. Wilson and D. B. Day, "Practical automatic test program generation constraints," in *Proc. Automatic Test Conf. and Workshop*, Apr. 1978.
[22] W. Maly, A. J. Strojwas, and S. W. Director, "VLSI yield prediction and estimation: A unified framework," *IEEE Trans. Computer-Aided Design*, vol. CAD-5, pp. 114-130, Jan. 1986.
[23] S. Liu and L. W. Nagel, "Small-signal MOSFET models for analog circuit design," *IEEE J. Solid-State Circuits*, vol. SC-17, pp. 983-998, Dec. 1982.
[24] K. Singhal, C. C. McAndrew, F. R. Nassis, and V. Visvanathan, "The CENTER design optimization system," *AT&T Tech. J.*, to be published.
[25] L. W. Nagel, "ADVICE for circuit simulation," in *Proc. Int. Symp. Circuits and Systems*, Houston, Texas, 1980.
[26] D. E. Hocevar, M. R. Lightner, and T. N. Trick, "An extrapolated yield approximation technique for use in yield maximization," *IEEE Trans. Computer-Aided Design*, vol. CAD-3, pp. 279-287, Oct. 1984.

*

**Linda Milor** received the B.S. degree in engineering physics from U.C. Berkeley in 1982. She is currently working towards the Ph.D. degree in the Department of Electrical Engineering at U.C. Berkeley.

She spent the summer of 1985 and the spring of 1986 working at AT&T Bell Laboratories, Murray Hill. Her research interests include testing, design, and analysis of analog circuits.

*

**V. Visvanathan** received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, New Delhi, India, the M.S.E.E. degree from the University of Notre Dame, IN, and the Ph.D. degree in electrical engineering and computer sciences from the University of California at Berkeley.

He has taught at the University of Maryland at College Park and the University of California at Berkeley. He has also worked at the IBM T. J. Watson Research Center, Yorktown Heights, NY. He is currently with AT&T Bell Laboratories, Murray Hill, NJ, where he is a Member of Technical Staff. His research interests are in various aspects of computer-aided analysis, design and testing of analog integrated circuits.