Nonlinear Decision Boundaries for Testing Analog Circuits

Haralampos-G. D. Stratigopoulos, *Student Member, IEEE*, and Yiorgos Makris, *Member, IEEE*

***Abstract*—A neural classifier that learns to separate the nominal from the faulty instances of a circuit in a measurement space is developed. Experimental evidence, which demonstrates that the required separation boundaries are, in general, nonlinear, is presented. Unlike previous solutions that build hyperplanes, the proposed classifier is capable of drawing nonlinear hypersurfaces. A new circuit instance is classified through a simple test, which examines the location of its measurement pattern with respect to these hypersurfaces. The classifier is trained through an algorithm that probably converges to the optimal separation boundary. Additionally, a feature selection algorithm interacts with the classifier to identify a discriminative low-dimensional measurement vector. Despite employing only a few measurements, the test criteria established by the neural classifier are strongly correlated to the performance parameters of the circuit and do not rely on a presumed fault model.**

***Index Terms*—Analog test, artificial intelligence, circuit classification, implicit functional test.**

I. Introduction

THE conventional approach to classification of analog circuits as nominal or faulty employs empirical tests to measure directly the macro performance parameters. Given the large variety of specifications of modern analog circuits, as well as the multiple repetitions necessary to moderate measurement error, this procedure is lengthy and cost ineffective [1]. Compounding this problem, circuit specifications refer to particular operational conditions, which require expensive and specialized test equipment to be reproduced [2]. In order to alleviate the burden of specification testing, several structural test methods have been explored [3]. Such methods typically employ test criteria in the form of a threshold that is set a priori based on the effects of faults within an underlying fault model. Therefore, no one-to-one mapping is established between compliance to these test criteria and accordance with the specifications of the circuit. As a result, test criteria are usually biased, impacting either the yield, if they are too strict, or the fault coverage, if they are too lenient [4]. To overcome the limitations of both specification and fault-model-based test methods, several methods [5]–[9] have been developed along a promising new direction termed as implicit functional testing [10].

Manuscript received July 19, 2004; revised November 7, 2004. This paper was recommended by Associate Editor K. Chakrabarty.

H.-G. D. Stratigopoulos is with the Department of Electrical Engineering, Yale University, New Haven, CT 06520-8285 USA (e-mail: haralampos-g.stratigopoulos@yale.edu).

Y. Makris is with the Departments of Electrical Engineering and Computer Science, Yale University, New Haven, CT 06520-8285 USA (e-mail: yiorgos.makris@yale.edu).

Digital Object Identifier 10.1109/TCAD.2005.855835

Implicit functional test methods employ prior knowledge collected from a representative sample set of circuit instances. This knowledge is used to train a system that is specified by some type of intelligence to classify new instances through a simple and inexpensive test. The learned test implicitly captures the actual joint distribution of all process control and layout parameters and correlates it to the corresponding drift of the performance parameters. By virtue of this correlation and despite its simplicity, the learned test provides comparable accuracy to the conventional functional tests.

Along these lines, the authors in [8] utilized a regression technique to approximate the function

$$f_{x\pi} : \tilde{x} \mapsto \tilde{\pi} \tag{1}$$

that maps the measurement vector, $\tilde{x}$, to the performance parameter vector, $\tilde{\pi}$. In this manner, the performance parameters are predicted by carrying out an adequate set of measurements and are subsequently compared to the specification tolerances, in order to classify the circuit. The extracted measurements constitute a set of samples of the output response to a transient input stimulus, which is optimized based on a genetic algorithm. It is noted that an explicit derivation of $f_{x\pi}$ would be feasible only for simple circuits [11] because it is, in general, nonlinear. Even if the circuit itself is linear, a feedback connection generates a nonlinear $f_{x\pi}$. An alternative approach, developed in [9], is a heteroscedastic probabilistic neural network that realizes a Bayesian classifier. The neural network outputs the class-conditional probabilities for the nominal circuit and for several fault clusters, which are modeled as a mixture of normal kernel functions. A circuit is then mapped through its measurement pattern to the class that wins the probability comparison.

Along a different direction, the method proposed in this paper develops a classification system that establishes a mapping

$$f'_{x\pi} : \tilde{x} \mapsto Y \tag{2}$$

where Y is a boolean variable that takes its value based on whether the measurement pattern $\tilde{x}$ stems from a circuit that satisfies all its specifications or not. Essentially, this mapping translates into an adequate number of hypersurfaces that separate the distributions of nominal and faulty circuits in the measurement space. The resulting test criterion for a circuit instance is to examine on which side of the separation hypersurfaces its measurement pattern falls.

More specifically, the proposed classification system is based on an artificial neural network that learns how to allocate these nonlinear decision boundaries in the measurement space. Learning is achieved through a set of measurements carried out on a representative sample of circuit instances that are functionally tested. Subsequently, the trained network will correctly classify new circuits, since their measurement patterns are produced by the same statistical mechanism as the training ones.

For a given set of measurements, the effectiveness of such classification methods depends on the accuracy with which the drawn boundaries separate the distributions of nominal and faulty circuits. Previously reported classifiers [5]–[7], establishing the mapping in (2), separate the nominal and faulty populations by drawing hyperplanes in the measurement space. However, as will be illustrated in Section III, the boundaries are nonlinear in practice, with curvatures that cannot be approximated by hyperplanes. Conversely, the network described herein is capable of drawing the necessary nonlinear separation boundaries and, thus, it reciprocates very well even in the presence of complex distributions. As an ancillary benefit of the flexibility provided by nonlinear boundaries, the proposed classifier requires fewer measurements to solve the separation problem.

The remainder of the paper is organized as follows. Section II refers to related work in detail. Section III presents experimental evidence that the required separation surfaces are indeed nonlinear. Section IV discusses the structure of the proposed neural network and its training algorithm, which allocates the separation hypersurfaces. Section V presents a feature selection algorithm that interacts with the neural network in order to identify the most useful measurements within an initial set. Finally, the classification capacity of the proposed network is illustrated in Section VI, using two example circuits, a Butterworth filter and an operational amplifier.

II. Related Work

Before proceeding to an in-depth analysis of the proposed classification system, previously reported classifiers [5]–[7] that build the mapping of (2) are briefly reviewed. The common characteristic of these methods is that they separate the nominal and faulty populations by drawing linear boundaries (i.e., hyperplanes) in the measurement space.

In [5], an analog circuit is modeled as a digital system by embedding digital-to-analog and analog-to-digital converters. Samples of the impulse response of the circuit are used as candidate signatures for distinguishing the distributions of the fault-free and faulty circuits. The impulse response is constructed by computing the cross correlation between the input pseudorandom digital sequence and the output response. Fisher's linear discriminant is used to separate the two distributions in the signature space and, therefore, the resulting multidimensional boundaries are linear. In [6], the authors use dc, ac, and I_{dd} measurements as candidate signatures for the allocation of the boundaries. Logistic discrimination analysis is used as the vehicle for the separation of the distributions. This type of analysis assumes that both distributions are normal with equal covariance matrices. In practice, this theoretical assumption is not satisfied.[1] Nevertheless, the method essentially inserts separating linear surfaces into the signature space. The authors in [7] propose the use of a two-layer neural network to allocate the separation boundaries. The network is trained to map a sampled version of the impulse response to pass or fail decisions. Since the hidden threshold units operate on linear combinations of samples, this network generates linear decision boundaries in the signature space. Moreover, its learning algorithm does not guarantee convergence to an optimal separation of the two distributions.

III. Nonlinear Decision Boundaries

In practice, the separation surfaces between nominal and faulty circuits can be very complex, and linear approximations using hyperplanes, such as in [5]–[7], may lead to high misclassification rates. Fig. 1 illustrates a few distribution examples in a two-dimensional (2-D) measurement space $x_1 - x_2$.[2] These scatter plots were drawn by performing a Monte Carlo simulation, where various parameters of the circuit follow a normal distribution centered at the nominal conditions. Each pattern in the measurement space represents a circuit instance. The circuits are denoted as nominal or faulty depending on whether they satisfy a single specification. The solid curves show the linear decision boundaries b_1 drawn by a neural network similar to the one described in [7], as well as the nonlinear decision

[1] Let $\tilde{p}$ be the circuit parameter vector and $f_{px} : \tilde{p} \mapsto x$. Consider the Taylor expansion of the measurement pattern x around the expectation $\tilde{\mu}$ of $\tilde{p}$, $\tilde{\mu} = E[\tilde{p}]$

$$\begin{aligned} x &= \sum_{j=0}^{\infty} \frac{1}{j!} \left[(\tilde{p} - \tilde{\mu}) \cdot \nabla_{\tilde{p}'} \right]^j f_{px}(\tilde{p}')|_{\tilde{p}'=\tilde{\mu}} \\ &= f_{px}(\tilde{\mu}) + (\tilde{p} - \tilde{\mu}) \cdot \nabla_{\tilde{p}'} f_{px}(\tilde{p}')|_{\tilde{p}'=\tilde{\mu}} \\ &\quad + \text{higher order terms.} \end{aligned}$$

If it is assumed that the elements of $\tilde{p}$, denoted by p_i, are independent and normally distributed, $p_i \sim N(\mu_i, \sigma_i)$, it can be shown that

$$\begin{aligned} &f_{px}(\tilde{\mu}) + (\tilde{p} - \tilde{\mu}) \cdot \nabla_{\tilde{p}'} f_{px}(\tilde{p}')|_{\tilde{p}'=\tilde{\mu}} \\ &\qquad \sim N\left(f_{px}(\tilde{\mu}), \nabla_{\tilde{p}'} f_{px}(\tilde{p}')|_{\tilde{p}'=\tilde{\mu}}^T \cdot \Sigma \cdot \nabla_{\tilde{p}'} f_{px}(\tilde{p}')|_{\tilde{p}'=\tilde{\mu}} \right) \end{aligned}$$

where Σ is the covariance matrix of the random vector $\tilde{p} : \Sigma_{ii} = \sigma_i^2$ and $\Sigma_{ij} = 0$ for $i \neq j$. The distribution of the higher order terms depends on the form of f_{px}. From (3), it can be seen that x behaves as a normal random variable only for $\tilde{p}$ close to the nominal vector $\tilde{\mu}$, where the higher order terms are negligible. In order to claim that x is normal, this behavior should be extended from $-\infty$ to ∞. However, as $\tilde{p}$ deviates from $\tilde{\mu}$, the higher order terms become increasingly significant, distorting the normal distribution. Dropping these terms would lead to a misleading distribution for x. For example, consider $x = p^2$ and $p \sim N(0, 1)$. Dropping the second term of the Taylor expansion yields $x \sim N(0, 1)$. However, it is known that p^2 has a chi-square distribution with one degree of freedom [12]. It should also be noted that, if the nominal and faulty populations were normal with equal covariance matrices, then the optimal decision boundary would be a hyperplane [13].

[2] These measurements were taken on the two circuits that appear in Section VI, a low-pass Butterworth filter [Fig. 1(a): x_1 is a sample of the impulse response and x_2 is a sample of the response for an ac input signal of frequency 16 kHz and magnitude 1 V; Fig. 1(b): x_1 and x_2 are both samples of the impulse response] and an operational amplifier [Fig. 1(c): x_1 is a dc signal and x_2 is a sample of the I_{dd} measurement by ramping the supply voltage]. Fig. 1(c) is normalized since x_2 is several orders of magnitude smaller than x_1.

boundaries b_2 produced by the higher order neural network that will be described in the subsequent sections.

Fig. 1(a) and (b) demonstrates the nonlinear nature of the ideal decision boundaries. A linear boundary will inevitably result in misclassifying a large number of circuits. Moreover, there are cases where a linear boundary will make completely misguided decisions. For example, consider the distribution of Fig. 1(c). The regions where the population of faulty circuits lie are nonconvex and disjoint. The classification system is not cognizant of the disjoint character of the faulty distribution and tries to fit a single boundary, as the examined specification is single ended. In this case, the linear boundary is allocated onto a space that is free of patterns, as any other choice would lead to even higher misclassification. All the patterns appear on one side of the line and, therefore, all the nominal circuits are erroneously classified as faulty since the number of patterns that correspond to nominal circuits is smaller. Such a biased decision is avoided when a higher order model is used. The figure shows that the neural network that was developed draws two nonlinear separating boundaries at the two sides of the distribution of nominal circuits. This neural network is capable of producing disjoint decision boundaries since, as will be illustrated in the next section, it essentially fits a second-order polynomial to separate the distributions on the space of measurements. Therefore, in the case of the 2-D scatter plot of Fig. 1(c), every value on the perpendicular axis corresponds to two real solutions on the horizontal axis.

The method proposed in this work is inherently extendible to boundaries of any order. While the option of polynomials of higher degree was considered, all the decision boundaries that came up in the experiments were accurately expressed by second-order polynomials. A higher order polynomial could potentially provide higher resolution for the data that is in the close proximity of the second-order boundary. However, such a fine-grained decision boundary runs the danger of overfitting the data and does not necessarily lead to better classification of previously unseen patterns, since there is always some degree of overlap between the distributions of nominal and faulty circuits.

The flexibility of nonlinear boundaries facilitates the discrimination of the two populations in a lower dimensional measurement space. It is true that, by adding more measurements, the point where the two populations are linearly separable can eventually be reached. In particular, in high dimensions, the measurement patterns are sparsely distributed, leaving a wide empty space between the two populations, where a linear boundary can fit. This might create the misperception that linear boundaries are adequate, provided that the input dimensionality is sufficiently large. The fact is, however, that the system, by construction, covers the entire space and, thus, the output assignment to an empty subspace will be random. As a result, patterns that fall within a subspace, which was empty during the learning phase, will be randomly labeled during the classification phase. Thus, increasing the number of measurements to the point where patterns are sparsely distributed has an adverse impact on classification of previously unseen patterns. This undesirable phenomenon has been termed as curse of dimensionality [13], [14].

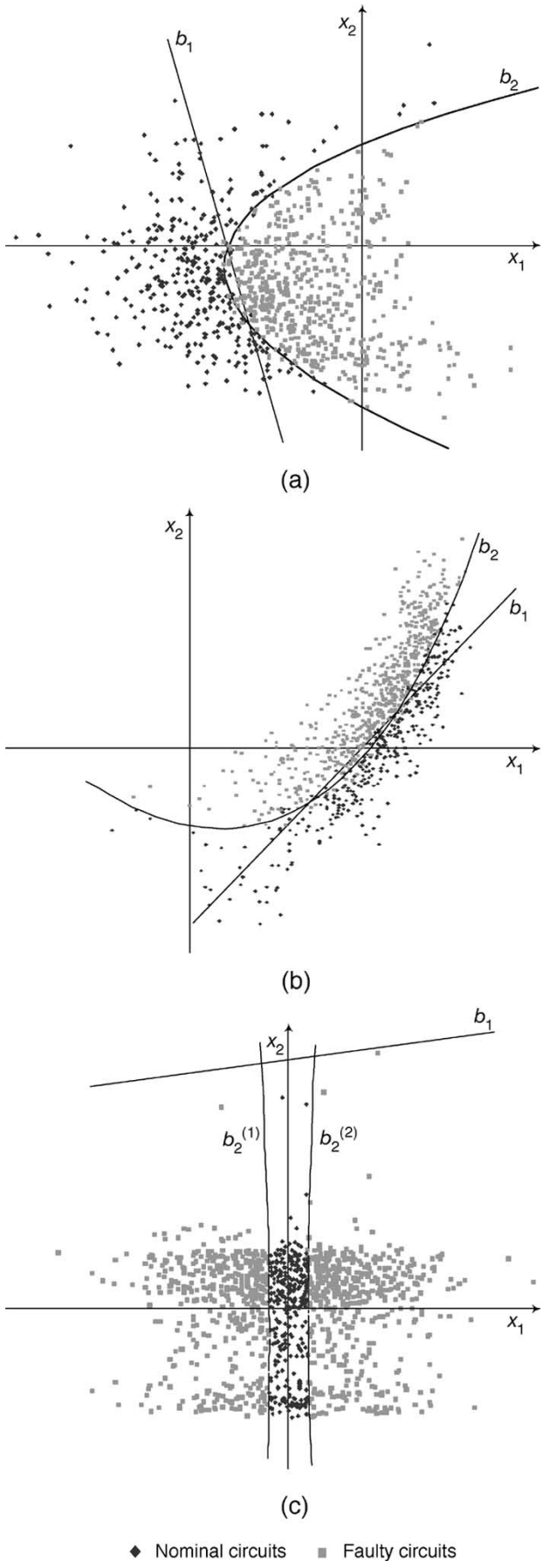

Fig. 1. Distributions of nominal and faulty circuits.

IV. Neural Network Topology and Learning

The neural network that is developed in this work provides a general framework for representing nonlinear functional mappings between a set of measurements and a pass/fail decision for a circuit. In this section, the prior knowledge required to train the proposed neural network is defined, its structure is analyzed, and how the network learns to classify previously unseen circuit instances is shown.

A. Starting Point

The topology of the neural network is solely defined by the number of single-ended specifications describing the design, denoted herein by M. The knowledge used to train the neural network is collected from a representative set of N_t circuits and comprises two elements, namely, the functional test results and a set of d measurements. The functional tests are encoded into an M-dimensional Boolean vector $\tilde{t}$, whose elements indicate whether the circuit satisfies the corresponding specifications or not. The values of the d measurements obtained on each of the N_t instances are gathered in d-dimensional vectors $\tilde{x} \in \Re^d$. The N_t patterns $[\tilde{x}^n, \tilde{t}^n]$, where $n = 1, \ldots, N_t$ labels the pattern, constitute the training set

$$T = \left\{(\tilde{x}^1, \tilde{t}^1), \ldots, (\tilde{x}^{N_t}, \tilde{t}^{N_t})\right\} \tag{3}$$

which is used to optimize the adaptive parameters of the neural network. The cost function for this optimization is the classification rate J_{N_t} for this particular set of circuit instances

$$J_{N_t} = \frac{1}{N_t} \sum_{i=1}^{N_t} \left(1 - h^{\mathrm{EC}}(\tilde{x}^i)\right) \tag{4}$$

where h^{EC} is the error counting function: $h^{\mathrm{EC}}(\tilde{x}^i) = 1$ if the pattern $\tilde{x}^i$ is misclassified and $h^{\mathrm{EC}}(\tilde{x}^i) = 0$ otherwise.

The objective in building a reliable classification system is not to fit a decision boundary that separates the training populations perfectly. Rather, the objective is to develop a system that gives accurate predictions for previously unseen instances, in other words, a system that exhibits good generalization. In order to assess the generalization capabilities of the proposed neural network, a second independent set of N_v circuits is used, which is called the validation set

$$V = \left\{(\tilde{x}^{N_t+1}, \tilde{t}^{N_t+1}), \ldots, (\tilde{x}^{N_t+N_v}, \tilde{t}^{N_t+N_v})\right\}. \tag{5}$$

The generalization rate $\hat{P}_{N_t}$ is defined similarly to (4)

$$\hat{P}_{N_t} = \frac{1}{N_v} \sum_{i=N_t+1}^{N_t+N_v} \left(1 - h^{\mathrm{EC}}(\tilde{x}^i)\right). \tag{6}$$

This method of generalization assessment is called hold-out. A number of other methods such as resubstitution, cross-validation, and bootstrap exist in the literature. The interested reader is referred to [15] for a detailed discussion.

B. Neural Network Topology

The proposed neural network has d inputs, one for each measurement x_1 through x_d. Its basic component is Frank Rosenblatt's perceptron [13], [14], which is illustrated in Fig. 2. A perceptron is a single-layer[3] network where the input row is transformed through a fixed set of processing elements ϕ_j, followed by adaptive weights w_j and a threshold activation function g of the form

$$g(\alpha) = \begin{cases} -1, & \text{for } a < 0 \\ +1, & \text{for } a \geq 0 \end{cases}. \tag{7}$$

Here, the d-dimensional measurement vector $\tilde{x}$ activates all the fixed processing units ϕ_j for $j = 1, \ldots, k$. The output of the extra processing element ϕ_0 is permanently set to $+1$, $\phi_0(\tilde{x}) = +1$, with a corresponding parameter w_0. The $(k+1)$-dimensional vector $\tilde{w} = [w_0, w_1, \ldots, w_k]^T$ shall be referred to as the weight vector and the parameter w_0 as the bias. The output of the perceptron is, therefore, given by

$$y = g\left(\sum_{j=0}^{k} w_j \phi_j(\tilde{x})\right) = g(\tilde{w}^T \tilde{\phi}) \tag{8}$$

where $\tilde{\phi} = [\phi_0, \ldots, \phi_k]^T$.

In the complete neural network, a perceptron represents a single-ended specification of the circuit. Suppose that the perceptron examined herein corresponds to specification s_μ, $\mu \in \{1, \ldots, M\}$. Let C_n^μ denote the class of circuits that comply to this specification and C_f^μ denote the class of circuits that violate it. As mentioned earlier, each pattern $\tilde{x}^n$ is associated with a variable t_μ^n

$$t_\mu^n = \begin{cases} -1, & \text{for } \tilde{x}^n \in C_f^\mu \\ +1, & \text{for } \tilde{x}^n \in C_n^\mu. \end{cases} \tag{9}$$

If the $+1$ and -1 in (7) correspond to the pass and fail decisions respectively, then, from (7) and (8), the conditions $\tilde{w}^T\tilde{\phi} > 0$ for all $\tilde{x}^n \in C_n^\mu$ and $\tilde{w}^T\tilde{\phi} < 0$ for all $\tilde{x}^n \in C_f^\mu$ should hold. Therefore, the inequality

$$\tilde{w}^T \left(\tilde{\phi}(\tilde{x}^n) t_\mu^n\right) > 0 \tag{10}$$

should hold for all measurement patterns in order to achieve optimal classification for the training set.

The output expression (8) has a simple geometrical representation. The perceptron divides the input space by a hypersurface, such that the activation function is $y = +1$ on one side of the hypersurface and $y = -1$ on the other side. Therefore, the populations of nominal and faulty circuits are divided in the measurement space by a boundary composed of the set of solutions to the equation $\tilde{w}^T\tilde{\phi} = 0$. The neural network learning process adjusts the weights in $\tilde{w}$ so that the allocation of the boundary will result in the lowest possible misclassification rate for the patterns in the training set. Since the training set reflects the underlying systematic aspects of the

[3]The term "single-layer" implies one layer of adaptive weights.

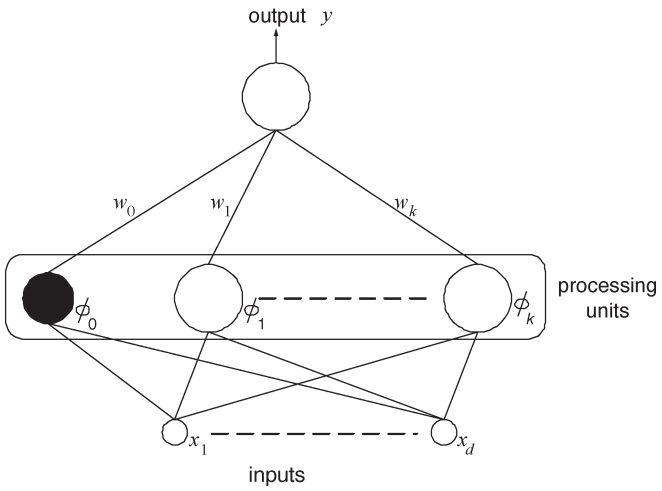

Fig. 2. Perceptron network.

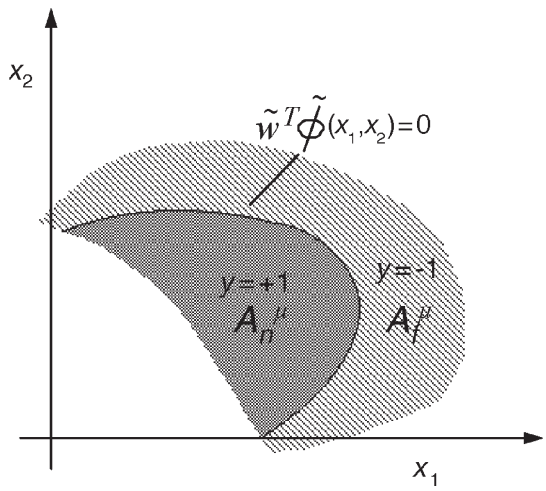

Fig. 3. Example decision boundary for a single perceptron in a 2-D measurement space.

data, this boundary is expected to classify correctly a previously unseen pattern. A possible decision boundary in a 2-D space is shown in Fig. 3. Circuits with measurement patterns that fall into the area A_n^μ are classified as nominal with respect to specification s_μ, while circuits with measurement patterns that fall into A_f^μ are classified as faulty.

The shape of the boundary depends on the selection of the fixed processing units. Clearly, if $\phi_j(\tilde{x}) = x_j$ and $k = d$, (8) reduces to the conventional linear case. For a second-order decision boundary, the fixed processing units ϕ_j must be chosen such that

$$\tilde{w}^T\tilde{\phi} = \sum_{i=0}^{d} w'_i x_i + \sum_{i_1=1}^{d}\sum_{i_2=1}^{d} w'_{i_1 i_2} x_{i_1} x_{i_2} = \sum_{i_1=0}^{d}\sum_{i_2=0}^{d} w'_{i_1 i_2} x_{i_1} x_{i_2} \tag{11}$$

where $x_0 = +1$ and $w'_{i0} = w'_{0i} = w'_i$.

Therefore, according to (11), the vector $\tilde{\phi}$ should be

$$\tilde{\phi} = \left[1, x_1, \ldots, x_d, x_1^2, x_1x_2, \ldots, x_1x_d, x_2^2, x_2x_3, \ldots, x_2x_d, \ldots\right]^T.$$

The summations in (11) can be constrained to allow for the permutation symmetry of the terms. Based on this remark and on the expression of $\tilde{\phi}$, it turns out that each perceptron should comprise $\binom{d+2}{2} = ((d+1)(d+2)/2)$ fixed processing units. The right-hand side of (11) then becomes

$$\sum_{i_1=0}^{d}\sum_{i_2=0}^{d} w'_{i_1 i_2} x_{i_1} x_{i_2} = \sum_{j=1}^{\frac{(d+1)(d+2)}{2}} w_j \phi_j(\tilde{x}). \tag{12}$$

The complete topology of the proposed neural network is shown in Fig. 4. Each perceptron p_μ corresponds to the single specification μ; thus, a total of M perceptrons is needed. The output of p_μ indicates whether the input pattern derives from a circuit that satisfies μ or from a circuit that does not. Specifically, the perceptron p_μ has activation function $y_\mu = +1$ if the circuit passes the specification μ and $y_\mu = -1$ if it fails it. The output unit has the threshold activation function of (7), with α being the sum of the outputs of all perceptrons and the output bias. By setting the bias of the output unit to $-M$, it computes the logic AND of the outputs of the M perceptrons: $\alpha = \sum_{\mu=1}^{M} y_\mu - M$. In other words, the neural network outputs $Y = +1$ if and only if the examined circuit satisfies all its specifications, and $Y = -1$ if at least one specification is violated. An example of decision boundaries for a circuit with five specifications in a 2-D measurement space is shown in Fig. 5. The intersection of all five individual acceptance regions is

$$A_n = \bigcap_{\mu=1}^{M=5} A_n^\mu.$$

Note that such a network can generate more complex decision boundaries as well, such as the piecewise boundaries shown in Fig. 1(c), where the regions of faulty circuits are disjoint.

C. Learning

In this section, the procedure through which the neural network learns to allocate the decision boundaries in the measurement space is discussed.

1) Training Algorithm: In order to minimize circuit misclassification, (10) suggests that weight values that minimize the following error function known as the perceptron criterion be selected

$$E^{\text{perc}}(\tilde{w}) = -\sum_{\tilde{x}^n : y(\tilde{x}^n) \neq t_\mu^n} \tilde{w}^T\left(\tilde{\phi}(\tilde{x}^n) t_\mu^n\right). \tag{13}$$

Here, the summation is over all patterns in the training set that are misclassified by the current weight vector $\tilde{w}$. The error function is the sum of a number of positive terms and is equal to zero if all patterns are correctly classified. The search in the

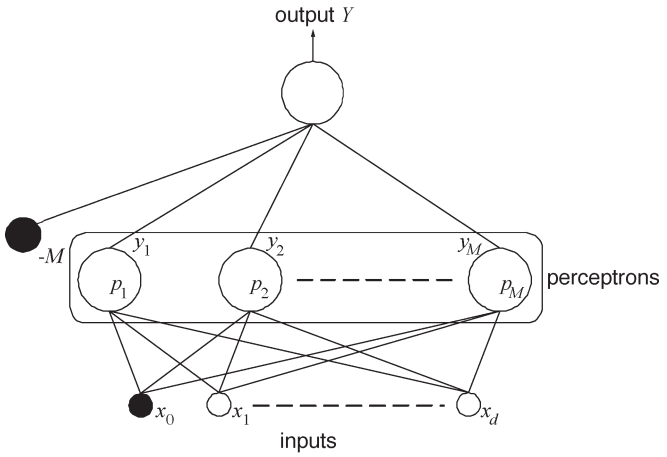

Fig. 4. Structure of the proposed neural network. Each p unit corresponds to a perceptron.

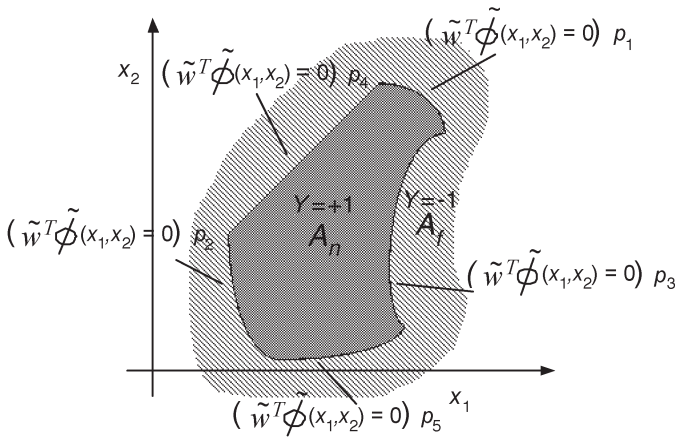

Fig. 5. Example acceptance region in a 2-D measurement space for a circuit with five specifications. $(\cdot)_{p_i}$ denotes the boundary allocated by perceptron i.

$(k+1)$-dimensional vector space of weights is performed by applying the gradient descent rule in a pattern by pattern fashion

$$w_j^{(t+1)} = w_j^{(t)} + \eta\phi_j(\tilde{x}^n)t_\mu^n \tag{14}$$

where $\eta > 0$. This corresponds to a simple learning procedure: Cycle through all patterns in the training set and test each pattern in turn using the current set of weight values. If the pattern is correctly classified then proceed to the next, otherwise, add $\eta\phi_j(\tilde{x})$ to the current weight vector if the pattern belongs to the nominal class, or subtract $\eta\phi_j(\tilde{x})$ if the pattern belongs to the faulty class. This procedure successively reduces the error since

$$\begin{aligned} -\tilde{w}^{(t+1)T}\left(\tilde{\phi}(\tilde{x}^n)t_\mu^n\right) &= -\tilde{w}^{(t)T}\left(\tilde{\phi}(\tilde{x}^n)t_\mu^n\right) \\ &\quad -\eta\left\|\tilde{\phi}(\tilde{x}^n)t_\mu^n\right\|^2 \\ &< -\tilde{w}^{(t)T}\left(\tilde{\phi}(\tilde{x}^n)t_\mu^n\right). \end{aligned} \tag{15}$$

In the case where the patterns $\phi_j(\tilde{x})$ are linearly separable, i.e., they can be perfectly separated by a hyperplanar decision boundary, the above learning rule is guaranteed to find a solution in a finite number of steps [16]. More specifically, let $R = \max_{0\le i\le N}\|\tilde{\phi}(\tilde{x}^i)\|$ and suppose that there exists a weight vector $\tilde{w}^*$ such that $t_\mu^i(\tilde{w}^{*T}\cdot\tilde{\phi}(\tilde{x}^i)) \ge \gamma$ for $0 \le i \le N$. Then, the number of mistakes made by the perceptron algorithm on the training set S is at most

$$2\left(\frac{R}{\gamma}\right)^2\left(\|\tilde{w}^*\|^2+1\right). \tag{16}$$

However, due to the continuous nature of analog signals, the problem of classifying analog circuits is nonseparable, at least in the lower dimensional space. For such nonseparable problems, the described perceptron rule is not well behaved. While the search will eventually visit an optimal set of weights, it will never terminate. Furthermore, the algorithm may pass from an optimal set of weights to a significantly worse set in one iteration, regardless of the number of iterations that have been previously performed. Indeed, the reduction of the cost function in each iteration does not necessarily imply that fewer patterns are misclassified by the new weight vector. Thus, the algorithm is stopped arbitrarily, and no guarantee can be provided for the quality of classification achieved by the identified weight vector on the training set and, by extension, on previously unseen data.

To address this issue, Pan and Cheng [7] used $\eta = K/\tau$, where K is a constant and τ is the step number. Hence, corrections to the weight vector become gradually smaller. However, this heuristic still does not guarantee convergence to an optimal boundary. To overcome this problem, a perceptron-based learning algorithm, called the pocket algorithm [17], is implemented, which probably converges to an optimal decision boundary with probability close to unit (for a general proof, see [18]). The basic idea is that the training samples are visited randomly and, once a weight vector that has a longer run of consecutive correct classifications is found, it replaces the current weight vector, and it is kept "in the pocket." The pocket weights are the outputs of the algorithm. Note that no upper time bound is known for this to occur, yet experience in the field [17], [19] indicates that very good weights are produced using reasonable computational effort. As a termination rule, the suggestion in [17] is followed. The algorithm starts with $i = 1000$ iterations. If weights are changed after the first $i/5$ iterations, then at the end of this set of iterations, another set of $1.5i$ iterations is applied. The algorithm continues in this manner until a set of iterations is ran with no weight replacement occurring during the last $0.8i$ iterations.

As an example, Fig. 6 illustrates the movement of the decision boundary in a 2-D space as training progresses. Here, the classifier learns the separation boundary of the two populations shown in Fig. 1(a). In Fig. 6, boundary $b_2^{(i)}$ corresponds to the ith pocket vector. Table I shows the iterations in which the pocket vector changes, as well as the classification rate that each particular pocket vector achieves. It can be seen that the pocket vector is replaced seven times and remains unchanged after 168 iterations. Thus, the algorithm stops after 800 iterations.

2) Small Faulty Population: Small training sets are commonly encountered in practical pattern classification problems. A known application where this occurs is the short-term forecasting of the demand for electric power from an electric utility [20]. A manufacturing process typically has high yield, with nominal circuits outnumbering significantly the faulty

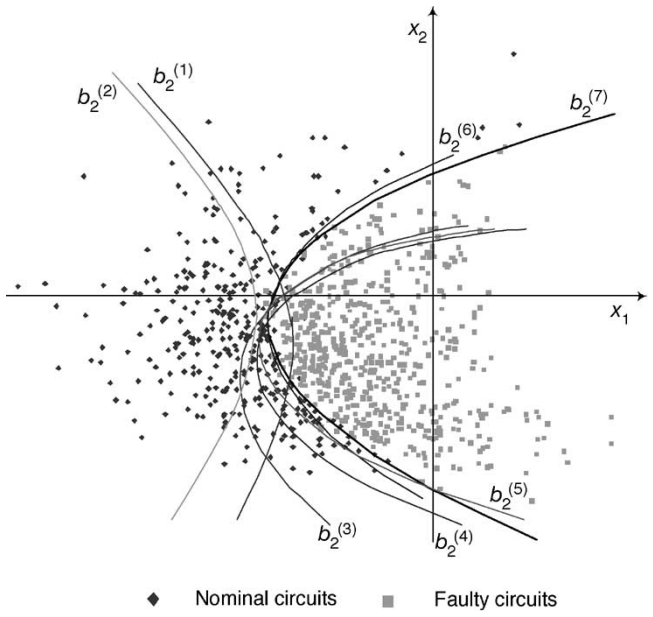

Fig. 6. Movement of the decision boundary as training progresses.

TABLE I
ITERATIONS IN WHICH THE POCKET VECTOR CHANGES AND THE CLASSIFICATION RATE THAT THE RESULTING BOUNDARY ACHIEVES. THE TRAINING SET CONSISTS OF 1000 PATTERNS

	$b_2^{(1)}$	$b_2^{(2)}$	$b_2^{(3)}$	$b_2^{(4)}$	$b_2^{(5)}$	$b_2^{(6)}$	$b_2^{(7)}$
Iteration	4	7	84	93	118	137	168
Rate	834	853	866	873	897	941	964

ones. Thus, the prior set $\{T, V\}$ typically contains a small faulty population. Before discussing the implications of such an unbalanced prior set and list a number of possible remedies, the difference between a small faulty population and a nonrepresentative faulty population is emphasized. The fundamental assumption of implicit functional test methods [5]–[9] is that both populations are representative, such that the estimated statistics of the classes are accurate. Thus, in this section, the problem of using a small, yet representative, faulty population in the training process is examined.

If the faulty population is small, the resulting decision boundary depends upon the partition of the prior set [21]. In particular, the more faulty patterns included in the training set, the more representative the decision boundary will be. Thus, as a first step, all faulty patterns in the training set need to be included. Even then, part of the region where the faulty population lies might be overshadowed by the dominant nominal patterns. As a result, the decision boundary will be pushed inside the faulty population region, possibly misclassifying a large portion of it. Finally, the curse of dimensionality will emerge faster when dealing with a small faulty population. In particular, as explained in Section III, the decision boundary will be set imprecisely if the ratio N_f/d is small, where N_f is the number of faulty patterns and d is the dimensionality.

In the case examined herein, where the distributions of the two populations are unknown and unequal and the classifier is nonlinear, it is not possible to estimate the number of faulty patterns needed. The authors in [22] give an empirical recommendation that is based on the ratio

$$r = \frac{\hat{P}^{\wp} - \hat{P}^{\ell}}{2 - (\hat{P}^{\wp} + \hat{P}^{\ell})}$$

where $\hat{P}^{\wp}$ and $\hat{P}^{\ell}$ are the generalization rates after training the classifier with the resubstitution and the leave-one-out methods, respectively.[4] If r is small, it can be concluded that the size of the faulty population is sufficient to achieve a desired level of learning accuracy. If the size of the faulty population is not sufficient, there are two remedies that can be followed to mitigate the effect. The first one relies on generating new faulty patterns as spherical normal distributed noise around the available ones

$$\tilde{x}_{ji} = \tilde{x}^j + \tilde{\xi}_{ji} \tag{17}$$

where $j = \{j | \tilde{x}^j \in C_f\}$ and each $\tilde{\xi}_{ji}$ is governed by a normal probability distribution centered at $\tilde{x}^j$. It has been shown both experimentally and theoretically that noise injection greatly improves the generalization ability [13], [23], [24]. Practically, it fills the space across the decision boundary, smoothing the generalization error and stabilizing the training procedure. The second remedy relies on training the classifier with the available information and then using the newly classified instances to retrain the classifier and update the decision boundary. The classified instances are referred to as semilabeled instances. It can be shown that learning with semilabeled patterns achieves comparable performance to learning with a large sample size, when the two populations are relatively separated [25]. To reduce the number of training epochs, the classifier can be retrained only if the classified pattern falls within a zone around the current decision boundary. Since no knowledge of the classification accuracy of each new instance exists, a control factor that is related to the likelihood of a class can be attached to it. Of course, this is not needed if each pattern that falls within this zone can be tested exhaustively. Experiments indicate that this adaptive classifier design quickly evolves into an optimal classifier with high generalization performance [25].

3) Biased Decisions: The training algorithm described in Section IV-C-1 produces an unbiased classifier, in the sense that it minimizes classification error without showing preference to one of the two classes. In essence, the classifier attempts to optimize both fault coverage and yield. If misclassifying faulty circuits as nominal (test escapes) is unacceptable, then the classifier should be biased towards rejecting faulty circuits. This constraint can be easily incorporated in the training process of the classifier. In particular, as explained in Section IV-C-1, the algorithm cycles through the training instances in a random fashion and corrects the decision boundary if a selected instance

[4]In the resubstitution method, all patterns are used to design the classifier and used again to estimate its generalization performance. In the leave-one-out method, $\binom{N}{N-1}$ classifiers are designed. Each classifier is designed by choosing $N-1$ of the N patterns as a training set, and its classification rate is evaluated on the left-out pattern. This process is repeated for all distinct choices of $N-1$ patterns and the average of the classification rates is computed.

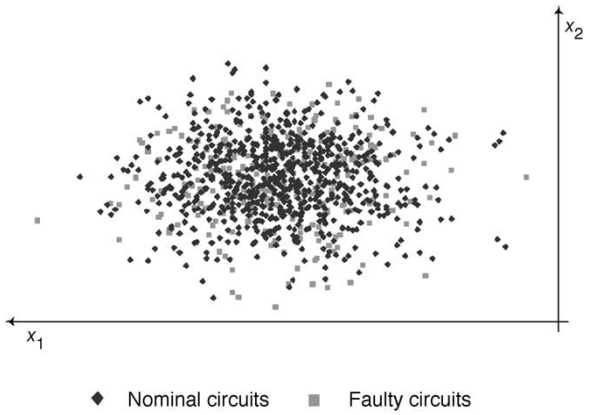

Fig. 7. Addition of measurement x_2 does not improve discrimination.

$(\tilde{x}^k, t_\mu^k)$ is misclassified. According to (10) and (15), this correction forces the dot product $\tilde{w}^T(\tilde{\phi}(\tilde{x}^k)t_\mu^k)$ to become more positive and likely to surpass the threshold of zero. This is equivalent to saying that the new boundary settles closer to the pattern that caused the change and is now likely to classify it correctly. Thus, if the training set is augmented by using multiple copies of the faulty patterns, the classifier will favor corrections of the decision boundary due to misclassified faulty circuits. As a result, the final decision boundary will classify correctly all faulty patterns in the training set and, thus, the probability of test escapes for unseen instances will be very small. Also note that within this constraint of eliminating test escapes, the training process still attempts to maximize yield.

V. Selection of Measurements

Increasing the dimensionality of the measurement space does not necessarily improve the efficiency of the classification system. For instance, an additional measurement may be strongly correlated to the current set of measurements and may not provide further classification ability. This is illustrated in Fig. 7, where nominal and faulty circuits overlap significantly in the space of the two selected measurements.[5] In essence, the two populations fall upon each other and, hence, adding measurement x_2 does not provide additional discrimination ability over measurement x_1. Moreover, as discussed in Section III, increasing the dimensionality of the measurement space may have an adverse effect on the generalization performance of the classifier due to the curse of dimensionality. Therefore, the separation problem should be solved in a measurement space that is both highly discriminative and low dimensional.

The problem of selecting the most effective d' measurements from a given set of d measurements, $d' < d$, is called feature selection. Feature selection methods are typically based on two components: 1) a criterion by which subsets of measurements are compared to each other; and 2) a systematic procedure for searching through the number of candidate subsets of measurements. The solution proposed in [26] chooses individual measurements, such that the difference between the means of the nominal and faulty classes is large and, at the same time, the within-class scatter is as small as possible. However, although the omitted measurements are of relatively little importance for the representation of the data itself, they may be crucial for the subsequent classification phase. For example, this method will not include two measurements, such as the ones illustrated in Fig. 1(b), where each one provides little discrimination by itself, yet their combination is very effective. Therefore, subsets of measurements should be considered, rather than individual measurements. The computational complexity of this approach is exponential in the number of measurements. Therefore, several efficient heuristics have been developed and a taxonomy of the most well-known algorithms can be found in [27].

Among them, a method called floating search was chosen to be implemented [28]. A comparative study [27] using data from synthetic aperture radar satellite images shows that this method yields near-optimal results, yet it is much faster than other feature selection approaches based on branch-and-bound algorithms [29]. Each measurement subset X_i is evaluated by training the neural network and computing the classification rate $J(X_i)$ achieved on the training set. A subset X_i is deemed better than another subset X_j if and only if $J(X_i) > J(X_j)$. This comparison is meaningful only for subsets of the same cardinality. The algorithm uses two basic procedures, the sequential forward selection (SFS) and the sequential backward selection (SBS). Given a subset of measurements X_k, SFS selects from the remaining measurements and includes the most significant one x_i with respect to the following subset:

$$x_i = \left\{ x_i | \max_i \left(J(X_k + x_i) - J(X_k) \right) \right\}. \tag{18}$$

Similarly, SBS selects from the current subset of measurements X_k and excludes the least significant one x_i with respect to the following subset:

$$x_i = \left\{ x_i | \min_i \left(J(X_k) - J(X_k - x_i) \right) \right\}. \tag{19}$$

The algorithm is a bottom-up search procedure that starts with an empty feature set and includes new features by means of applying the basic SFS procedure. The feature inclusion phase is followed by a series of successive conditional exclusions of the worst feature in the newly updated set through the SBS procedure, provided that a further improvement can be made to previous sets of lower cardinality. The algorithm records the best identified subsets $S^{d'}$, $d' \in \{1, \ldots, d-1\}$, and updates the list when better ones are found. Upon termination, the algorithm reports the subsets $S^{d'}$ for every $d' \in \{1, \ldots, d-1\}$. A simplified flowchart of the algorithm is shown in Fig. 8. It is emphasized that, unlike methods such as in [30], [31], where measurements are selected based on fault coverage and yield corresponding to a particular fault model, the proposed method selects subsets of measurements based on actual classification rates achieved on a representative sample of naturally labeled instances.

A snapshot of the algorithm running on an example with five measurements is shown in Fig. 9: At some point, the algorithm

[5]These measurements were taken on the operational amplifier that appears in Section VI. x_1 is the Fourier coefficient of the first harmonic and x_2 is a dc signal.

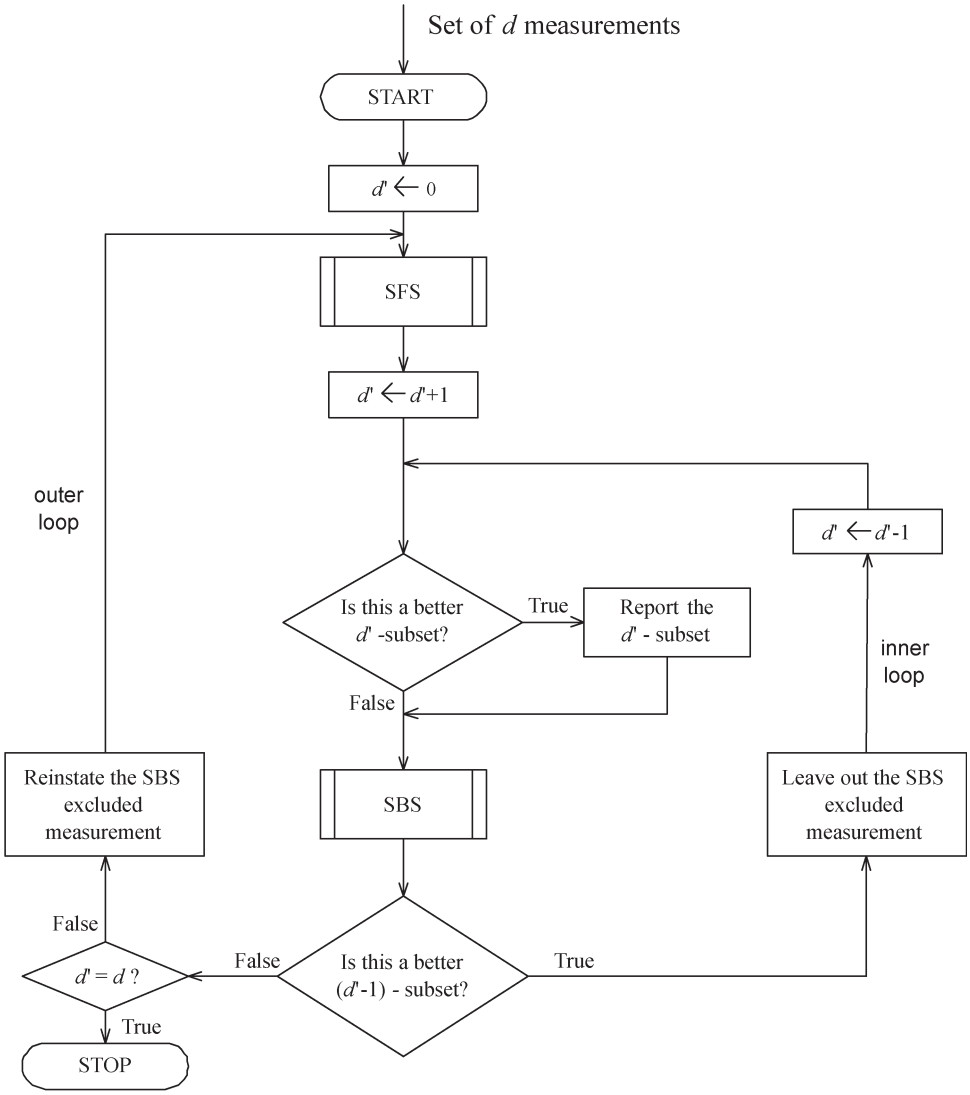

Fig. 8. Flowchart for the measurement selection method.

examines $\{x_1, x_3, x_4\}$ and evaluates $J(\{x_1, x_3, x_4\}) > J(S^3)$. Thus, it updates the list $S^3 = \{x_1, x_3, x_4\}$. It then determines that the least significant measurement with respect to S^3 is x_4. However, $J(\{x_1, x_3\}) < J(S^2)$, leaving the list unchanged. Then, x_4 is reinstated and the algorithm selects the most significant measurement with respect to S^3. This inclusion gives a better subset of cardinality four. In the last step shown, a better subset of cardinality three is identified.

VI. Experimental Results

In this section, the proposed classification method is evaluated on two example circuits, a Butterworth low-pass filter and an operational amplifier. The two circuits are shown in Fig. 10(a) and (b), respectively, and their list of specifications is given in Tables II and III. The experiments illustrate the following:

1) Nonlinear decision boundaries provide better separation of the nominal and faulty populations than their linear counterparts.
2) Nonlinear decision boundaries generalize better than their linear counterparts. Thus, the resulting test criterion is more accurate. Furthermore, the maximum generalization is achieved in a low-dimensional space, which points to a criterion that is simple to evaluate.
3) The classification rate on the training set increases monotonically with the number of measurements, i.e., $J(X^+) \geq J(X)$, where X and X^+ denote sets of measurements, with $|X| \leq |X^+|$. However, for the selected best subsets of measurements, monotonicity is not necessarily satisfied on the validation set. This verifies the existence of the curse of dimensionality.

A. Experimental Setup

The course of the experiment is as follows:

1) First, an initial set of ten measurements is selected. This set was obtained by applying a realistic statistical simulation of the circuit and identifying test stimuli responses (and their respective sampling times), which are the most sensitive to process variations [32]. Various types of

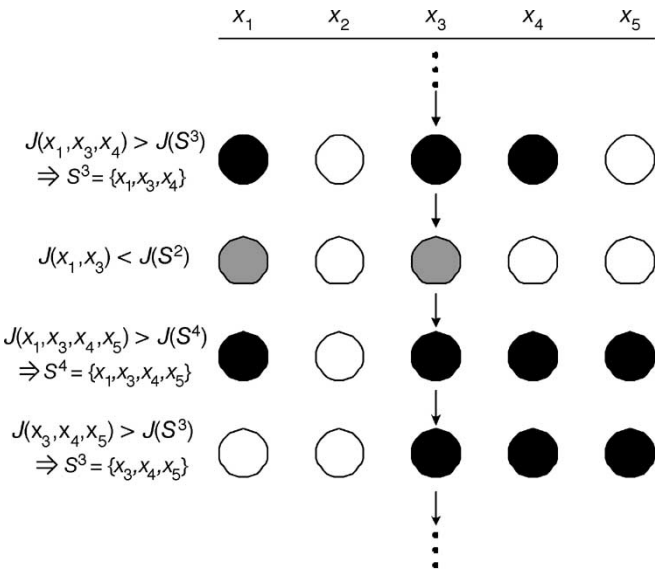

Fig. 9. Snapshot of the measurement selection method.

measurements are examined in this analysis: ac, dc, impulse response [33], Fourier transformation of the power supply current [34], and power supply ramping [35]. It is noted that the classification rate that the proposed method can achieve is bounded by the discrimination ability of this initial set of measurements. While more advanced test generation methods could probably improve this upper bound, such improvement is not related to the quality of the classifier and is out of the scope of this paper.

2) The circuit instances that constitute the training and the validation set for the classifier are then obtained. In a production environment, the training set will comprise a small number of circuits that will be functionally tested and on which the selected measurements will be carried out. For the purpose of this experiment, these circuit instances are generated through a Monte Carlo simulation with 2000 runs. Half of them are used as the training set and the other half as the validation set. All amplifiers in the active filter are used in a negative feedback configuration and, thus, the performance of the circuit is dominated primarily by variations in the values of the resistors and capacitors. Therefore, only disturbances of the resistors and capacitors connected externally to the operational amplifiers are considered. For the operational amplifier, transistor geometries L and W, oxide thickness $t_{\rm ox}$, threshold voltage V_T, body effect coefficient γ, junction capacitances, and the capacitor C_c are considered. For both circuits, all parameters are modeled by a normal distribution centered at their nominal value with a deviation of 10%. It is emphasized that these faults are only used for the purpose of generating the circuit instances in the training set. In practice, the proposed classification system does not rely on any assumption about an underlying fault model.
3) The vector $[\tilde{x}, \tilde{t}\,]$ is computed for each circuit instance. Individual measurements might have typical values that differ by several orders of magnitude. For example, the Fourier coefficient of the second harmonic can be considerably smaller than a voltage measurement. As a result, the neural network might assign larger weights to measurements with larger typical values. However, the typical sizes of the measurements do not reflect their relative importance in the classification problem. Therefore, all measurements are normalized before they are presented to the network. Each measurement is treated independently. Its mean, $\bar{x_i}$, and its variance, σ_i^2, are calculated with respect to the training set using

$$\bar{x_i} = \frac{1}{N}\sum_{n=1}^{N} x_i^n$$

$$\sigma_i^2 = \frac{1}{N-1}\sum_{n=1}^{N} \left(x_i^n - \bar{x_i}\right)^2.$$

The set of normalized measurements is then defined as

$$x_i^{n'} = \frac{x_i^n - \bar{x_i}}{\sigma_i}.$$

4) The normalized vectors $[\tilde{x}_i^{n'}, \tilde{t}^{\,n}]$, for all patterns n that belong to the training set, serve as input to the feature selection algorithm of Section V, which searches for efficient measurement subsets with regards to the classification rate that they achieve. Upon completion, the algorithm reports the best subset of d' measurements that it has identified for every $d' \in \{1, \ldots, 9\}$, along with the corresponding weight vector for the adaptive parameters of the neural network and the classification rate achieved on the training set.
5) For each best subset of d' measurements and for every $d' \in \{1, \ldots, 10\}$, the classification rate is also computed on the validation set, in order to examine the generalization performance of the trained neural network.
6) For comparison purposes, steps 4 and 5 are repeated for a linear neural network that has perceptrons with fixed processing elements $\phi_j(\tilde{x}) = x_j$, $j = 1, \ldots, d$. Since the training algorithm probably yields an optimal decision boundary [17], [18], the linear boundaries that this network produces are at least as good as the ones produced by the classification systems in [5] and [7]. Note also that the feature selection algorithm may produce different best subsets than in the nonlinear case.

Next, the results for each circuit are discussed, providing examples of separating boundaries and classification rates for both training and validation sets.

B. Butterworth Low-Pass Filter

Fig. 11 displays the distributions of nominal and faulty circuits projected onto the space of the best pair of measurements. Here, the best 2-D subset of measurements is identical for both the linear and the second-order network. Each decision boundary b_j^{μ}, where $\mu \in \{1, 2\}$, $j = 1$ for the linear network

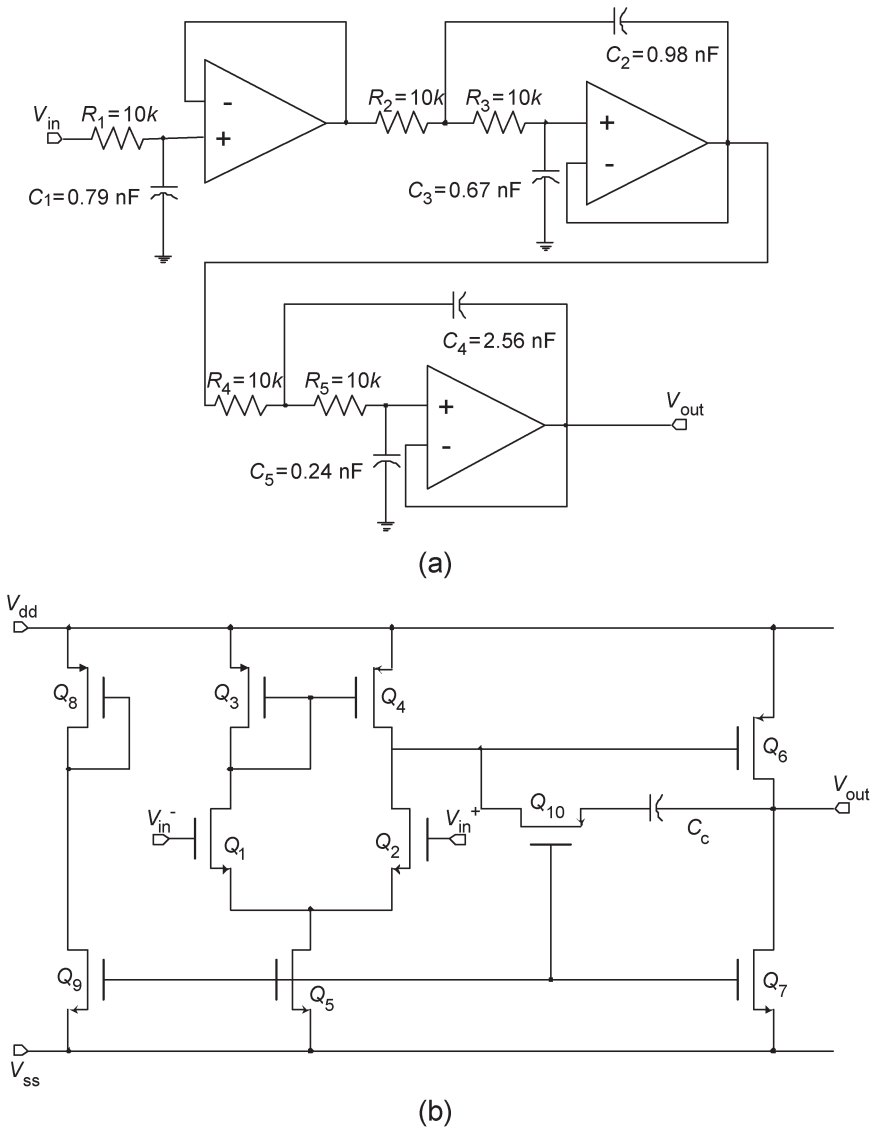

Fig. 10. Schematics of (a) Butterworth low-pass filter and (b) complementary metal oxide semiconductor (CMOS) operational amplifier.

TABLE II
SPECIFICATIONS OF THE LOW-PASS BUTTERWORTH FILTER CIRCUIT. $\alpha = -A$, WHERE A IS THE GAIN IN DECIBELS

$\alpha < 0.55$dB at $f = 16$kHz
$\alpha > 21$dB at $f = 32$kHz

TABLE III
SPECIFICATIONS OF THE OPERATIONAL AMPLIFIER

Low frequency gain	$A_0 \geq 82\,\mathrm{dB}$
Unity-gain frequency	$f_0 \geq 4.8$ MHz
Slew-rate	$S_r \geq 1.7\,\mathrm{V/\mu s}$
Common-mode rejection ratio	$CMRR \geq 113\,\mathrm{dB}$
Phase margin	$\phi M > 80^{\mathrm{o}}$

and $j = 2$ for the proposed network, divides the measurement space into two regions $A_n^{\mu}(j)$ and $A_f^{\mu}(j)$. Circuits that fall into $A_n^{\mu}(j)$ are classified as nominal, while circuits that fall into $A_f^{\mu}(j)$ are classified as faulty with respect to specification s_{μ}. Therefore, the entire measurement space is divided into four decision regions: circuits that fall into $A_f^1(j) \bigcap A_f^2(j)$ violate both specifications, circuits that fall into $A_n^1(j) \bigcap A_f^2(j)$ satisfy only the first specification and so forth. As can be observed, the acceptance region defined by the two nonlinear boundaries, $A_n^1(2) \bigcap A_n^2(2)$, approximates the area of nominal circuits better than the acceptance region defined by the two linear boundaries, $A_n^1(1) \bigcap A_n^2(1)$.

Fig. 12 shows graphically the classification rates for the training and validation set for both types of networks. The proposed network can be trained using five measurements to classify correctly all patterns in the training set. The linear network does not reach this optimum rate even if all ten available measurements are utilized. This shows that the nonlinear boundaries separate the populations more effectively. Furthermore, the linear network requires seven measurements to reach the same rate achieved by the proposed network with only two measurements. Hence, the separation is also achieved

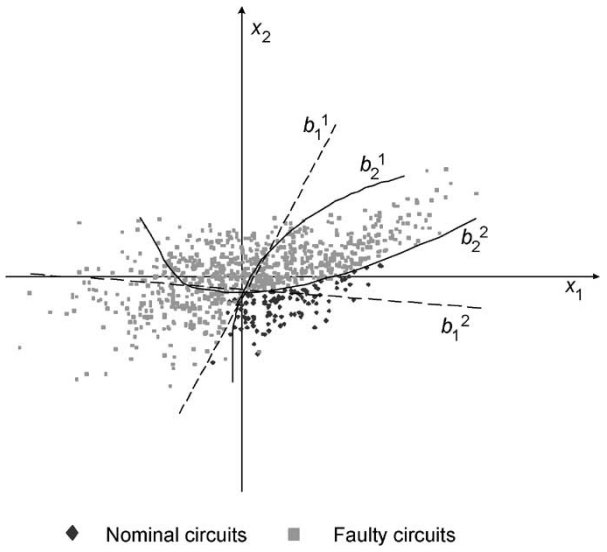

Fig. 11. Distributions in the space of the best pair of measurements for the Butterworth low-pass filter; both measurements are samples of the filter's output response for an ac input signal of frequency 16 kHz and magnitude 1 V.

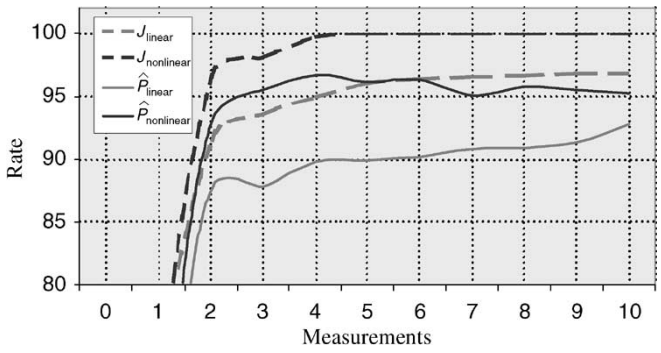

Fig. 12. Performance on the training set J and generalization performance $\hat{P}$ of the linear and nonlinear boundaries for the Butterworth low-pass filter.

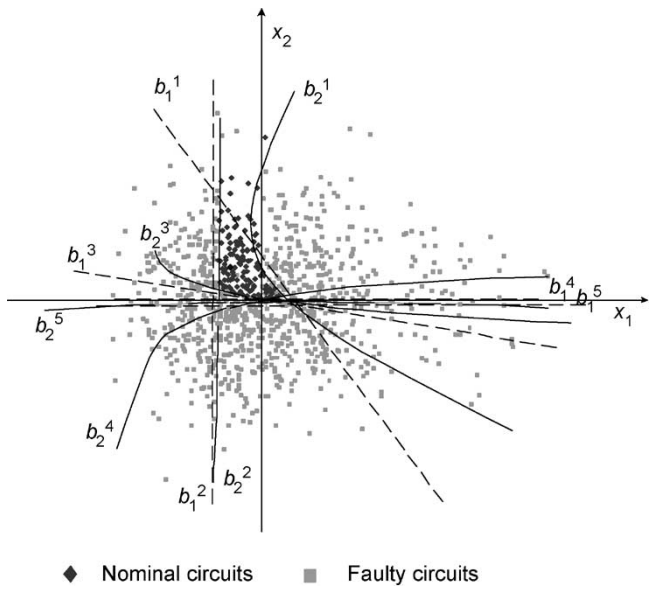

Fig. 13. Distributions in the space of the best pair of measurements for the operational amplifier; x_1 is a sample of the response for an ac signal of frequency 100 kHz and magnitude 1 V, when the amplifier is configured in a differentiating topology [6], and x_2 is the dc response for an input of 100 mV, when the amplifier is configured in an inverting topology.

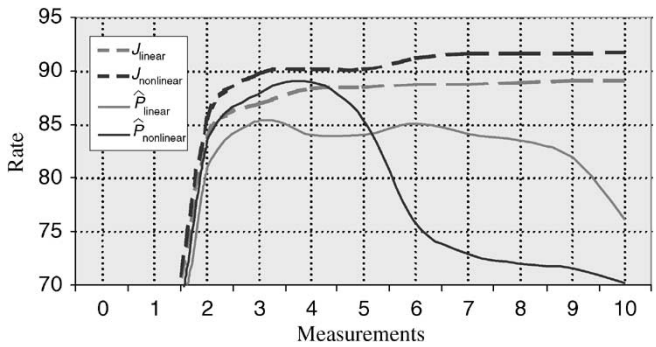

Fig. 14. Performance on the training set J and generalization performance $\hat{P}$ of the linear and nonlinear boundaries for the operational amplifier.

in a lower dimensional space. On the validation set, the linear network reaches the classification rate of 92.8% when all ten measurements are used, which is achieved by only two measurements for the second-order network. In addition, this rate is much smaller than the best rate of 96.7% for the second-order network, which is achieved for four measurements. Thus, the second-order network not only exhibits better generalization performance, but it also reaches a specified generalization rate using fewer measurements.

In Fig. 12, it is observed that the misclassification for the training set is reduced by using a larger set of measurements, but the rate of improvement decreases as the number of measurements increases. This suggests that the search of the feature selection algorithm in low-dimensional spaces is crucial in order to find the most relevant measurements. The curse of dimensionality is slightly evident on the validation set for the second-order network. The best generalization performance is achieved for four measurements and monotonicity is not always satisfied for larger sets of measurements. This phenomenon is not observed in the linear network. The fact that the curse is slightly evident for the second-order network and absent for the linear one implies that the core of the measurement patterns remains tight in this particular ten-dimensional space.

C. Operational Amplifier

Fig. 13 shows the two distributions projected onto the space of the best measurement pair with regards to the nonlinear neural network. The feature selection algorithm produced a different best pair for the linear network. Nevertheless, the respective linear boundaries for the best pair with regards to the nonlinear case are also shown in the figure in order to illustrate the acceptance regions established by the two networks. Once again, it can be observed that the acceptance area defined by the five nonlinear boundaries, $\bigcap_{\mu=1}^{5} A_n^{\mu}(2)$, approximates very effectively the actual area of nominal circuits.

The classification capabilities for all best subsets of measurements and for both types of networks are illustrated graphically in Fig. 14. As in the previous example, the improvement of the classification rate for the training set decreases as the number of measurements increases. With respect to the training set, the proposed network achieves a better classification rate using three measurements than the rate produced by the linear network using the complete set of measurements. The best generalization performance for the proposed network is achieved using four measurements and is appreciably larger than the respective best generalization performance for the linear network, which occurs for the best subset of cardinality three.

The curse of dimensionality is evident for both types of networks. Specifically, concerning the proposed network, the generalization rate reaches a peak and then decreases monotonically. Similarly, for the linear network, the monotonicity of the generalization rate is not satisfied after the third measurement.

VII. Conclusion

An implicit functional test method for analog circuits that relies on a nonlinear artificial neural classifier was discussed. The classifier reaches a decision on the acceptability of the performance parameters of the circuit based on a small set of measurements. This assessment is independent of the fault origins that cause violation of specifications and, thus, it does not rely on a particular fault model. A feature selection algorithm interacts with the neural network to choose the most useful measurements for the classification task. Experimental results show that the proposed network provides a substantially better generalization performance than previously reported methods that establish linear separation boundaries in the measurement space. It was also demonstrated that the proposed network achieves higher classification rates using a smaller number of measurements. Furthermore, through the experiments, the occurrence of the curse of dimensionality was pointed out, which suggests that, for this type of test methods, an efficient combination of measurements is desired, rather than a large set.

References

[1] R. J. Van Rijsinge, A. A. R. M. Haggenburg, C. de Vries, and H. Wallinga, "From specification to measurement: The bottleneck in analog industrial testing," in *IEEE Int. Test Conf.*, Washington, DC, 1990, pp. 177–182.

[2] W. M. Lindermeir, "Design of robust test criteria in analog testing," in *ACM/IEEE Int. Conf. Computer-Aided Design*, San Jose, CA, 1996, pp. 604–611.

[3] M. Soma, "Automatic test generation algorithms for analogue circuits," *IEE Proc., Circ. Devices Syst.*, vol. 143, no. 6, pp. 366–373, 1996.

[4] B. Vinnakota and R. Harjani, "DFT for digital detection of analog parametric faults in SC filters," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 19, no. 7, pp. 789–798, Jul. 2000.

[5] C. Y. Pan and K. T. Cheng, "Pseudorandom testing for mixed-signal circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 16, no. 10, pp. 1173–1185, Oct. 1997.

[6] W. M. Lindermeir, H. E. Graeb, and K. J. Antreich, "Analog testing by characteristic observation inference," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 18, no. 9, pp. 1353–1368, Sep. 1999.

[7] C. Y. Pan and K. T. Cheng, "Test generation for linear time-invariant analog circuits," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 5, pp. 554–564, May 1999.

[8] P. N. Variyam, S. Cherubal, and A. Chatterjee, "Prediction of analog performance parameters using fast transient testing," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 21, no. 3, pp. 349–361, Mar. 2002.

[9] Z. R. Yang, M. Zwolinski, C. D. Chalk, and A. C. Williams, "Applying a robust heteroscedastic probabilistic neural network to analog fault detection and classification," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 19, no. 1, pp. 142–151, Jan. 2000.

[10] C. Y. Pan and K. T. Cheng, "Implicit functional testing for analog circuits," in *IEEE VLSI Test Symp.*, Princeton, NJ, 1996, pp. 489–494.

[11] J. Savir and Z. Guo, "Test limitations of parametric faults in analog test," *IEEE Trans. Instrum. Meas.*, vol. 52, no. 5, pp. 1444–1454, Oct. 2003.

[12] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. New York: McGraw-Hill, 1991.

[13] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.

[14] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.

[15] G. T. Toussaint, "Bibliography on estimation of misclassification," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 4, pp. 472–479, Jul. 1974.

[16] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*. Reading, MA: Addison-Wesley, 1991.

[17] S. I. Gallant, "Perceptron-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 1, no. 2, pp. 179–191, Jun. 1990.

[18] M. Muselli, "On convergence properties of pocket algorithm," *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 623–629, May 1997.

[19] S. I. Gallant, "Optimal linear discriminants," in *8th Int. Conf. Pattern Recognition*, Paris, France, 1986, pp. 849–852.

[20] J. L. Yuan and T. L. Fine, "Neural-network design for small training sets of high dimension," *IEEE Trans. Neural Netw.*, vol. 9, no. 2, pp. 266–280, Mar. 1998.

[21] C. Kirsopp and M. Shepperd, "Making inferences with small numbers of training sets," *IEE Proc., Softw.*, vol. 149, no. 5, pp. 123–130, 2002.

[22] S. J. Raudys and A. K. Jain, "Small sample size effects in statistical pattern recognition: Recommendations for practitioners," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 3, pp. 252–264, Mar. 1991.

[23] J. Sietsma and R. J. F. Dow, "Creating artificial neural networks that generalize," *Neural Netw.*, vol. 4, no. 1, pp. 67–79, 1991.

[24] M. Skurichina, S. Raudys, and R. P. W. Duin, "K-nearest neighbors directed noise injection in multilayer perceptron training," *IEEE Trans. Neural Netw.*, vol. 11, no. 2, pp. 504–511, Mar. 2000.

[25] Q. Jackson and D. A. Landgrebe, "An adaptive classifier design for high-dimensional data analysis with a limited training data set," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 12, pp. 2664–2679, Dec. 2001.

[26] Z. Wang, G. Gielen, and W. Sansen, "Probabilistic fault detection and the selection of measurements for analog integrated circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 17, no. 9, pp. 862–872, Sep. 1998.

[27] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 2, pp. 153–158, Feb. 1997.

[28] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," *Pattern Recogn. Lett.*, vol. 15, no. 11, pp. 1119–1125, 1994.

[29] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Trans. Comput.*, vol. 26, no. 9, pp. 917–922, Sep. 1977.

[30] L. Milor and A. L. Sangiovanni-Vincentelli, "Minimizing production test time to detect faults in analog circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 13, no. 6, pp. 796–813, Jun. 1994.

[31] G. Devarayanadurg, M. Soma, P. Goteti, and S. D. Huynh, "Test set selection for structural faults in analog IC's," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 18, no. 7, pp. 1026–1039, Jul. 1999.

[32] Z. Jaworski, M. Niewczas, and W. Kuzmicz, "Extension of inductive fault analysis to parametric faults in analog circuits with application to test generation," in *IEEE VLSI Test Symp.*, Monterey, CA, 1997, pp. 172–176.

[33] Y. Maidon, B. W. Jervis, P. Fouillat, and S. Lesage, "Using artificial neural networks or Lagrange interpolation to characterize the faults in an analog circuit: An experimental study," *IEEE Trans. Instrum. Meas.*, vol. 48, no. 5, pp. 932–938, Oct. 1999.

[34] G. Gielen, Z. Wang, and W. Sansen, "Fault detection and input stimulus determination for the testing of analog integrated circuits based on power-supply current monitoring," in *ACM/IEEE Int. Conf. Computer-Aided Design*, Santa Clara, CA, 1994, pp. 495–498.

[35] S. S. Somayajula, E. Sanchez-Sinencio, and J. Pineda de Gyvez, "Analog fault diagnosis based on ramping power supply current signature clusters," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 43, no. 10, pp. 703–712, Oct. 1996.

Haralampos-G. D. Stratigopoulos (S'02) received the Diploma in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 2001, and the M.S. degree in electrical engineering from Yale University, New Haven, CT, in 2003. He is currently working toward the Ph.D. degree in electrical engineering at Yale University.

His research interests include analog circuit testing, analog design, pattern recognition, and modeling of the metal oxide semiconductor (MOS) transistor.

Yiorgos Makris (S'96–M'02) received the Diploma in computer engineering and informatics from the University of Patras, Patras, Greece, in 1995, and the M.S. and Ph.D. degrees in computer science and engineering from the University of California, San Diego, in 1997 and 2001, respectively.

Since 2001, he has been an Assistant Professor of Electrical Engineering and Computer Science at Yale University, where he leads the Testable and Reliable Architectures (TRELA) research group. His research interests include test and reliability of digital and analog circuits and systems.