

# Error Moderation in Low-Cost Machine-Learning-Based Analog/RF Testing

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

**Abstract**—Machine-learning-based test methods for analog/RF devices have been the subject of intense investigation over the last decade. However, despite the significant cost benefits that these methods promise, they have seen a limited success in replacing the traditional specification testing, mainly due to the incurred test error which, albeit small, cannot meet industrial standards. To address this problem, we introduce a neural system that is trained not only to predict the pass/fail labels of devices based on a set of low-cost measurements, as aimed by the previous machine-learning-based test methods, but also to assess the confidence in this prediction. Devices for which this confidence is insufficient are then retested through the more expensive specification testing in order to reach an accurate test decision. Thus, this two-tier test approach sustains the high accuracy of specification testing while leveraging the low cost of machine-learning-based testing. In addition, by varying the desired level of confidence, it enables the exploration of the tradeoff between test cost and test accuracy and facilitates the development of cost-effective test plans. We discuss the structure and the training algorithm of an ontogenic neural network which is embodied in the neural system in the first tier, as well as the extraction of appropriate measurements such that only a small fraction of devices are funneled to the second tier. The proposed test-error-moderation method is demonstrated on a switched-capacitor filter and an ultrahigh-frequency receiver front end.

**Index Terms**—Alternate testing, analog circuits, circuit testing, machine learning, RF circuits.

## I. INTRODUCTION

THE CURRENT practice for testing analog/RF devices is specification (or parametric) testing, which involves direct measurement of the performance parameters (e.g., gain, integral nonlinearity, noise figure, third-order intercept point, etc.). While specification testing is highly accurate, it often incurs a very high cost. Indeed, testing the analog/RF functions of a mixed-signal integrated circuit (IC) is typically responsible for the majority of the total cost despite the fact that the vast majority of the IC is digital [1]. In particular, the base cost per second of automatic test equipment (ATE) escalates rapidly when incorporating mixed-signal and RF features. Compounding this problem, specification testing involves long test times. Specifically, during its course, the device is con-

secutively switched to numerous test configurations, resulting in long setup and settling times. In each test configuration, measurements are performed multiple times and averaged in order to moderate thermal noise and crosstalk. In addition, this elaborate procedure is repeated under various operational modes such as temperatures, voltage levels, and output loads.

In recent years, machine learning inspired a new test paradigm, wherein the results of specification testing are inferred from a few simple measurements that are rapidly obtained using an assortment of low-cost test equipment [2]–[20].<sup>1</sup> The idea is to use a training set of device instances, on which both the specification tests and the low-cost measurements are performed, in order to derive the underlying mapping. The rationale is that the training set reflects the statistical mechanisms of the manufacturing process, and therefore, the learned mapping exhibits good generalization for new device instances produced by this process.

Despite offering a low-cost alternative to specification testing, the accuracy of machine-learning-based test methods is not up to par due to the following reasons: 1) When the populations of nominal and faulty devices are projected in a space of simple measurements, they may overlap to some extent, unlike the space of performance parameters, wherein they are cleanly separated by the design specifications; 2) the finite number of devices in the training set may result in learning a poor representation of the actual mapping; and 3) certain methods [2]–[5] pose restrictions on the order of the mapping; thus, they reciprocate poorly when the actual mapping is more complex.

The aim of this paper is to bridge the accuracy of specification testing and machine-learning-based testing. In particular, we propose to first test all fabricated devices through the low-cost machine-learning-based testing, assess the confidence in the test outcome, and, in case this confidence is deemed insufficient, retest the device through more expensive specification testing. To support this two-tier test scheme, we design a neural system comprising a committee of ontogenic neural networks. In a training phase, the neural system allocates guard bands<sup>2</sup> to partition a measurement space into regions wherein test decisions can be made with confidence or wherein ambivalence prevails. Thus, the neural system learns to either infer the results

Manuscript received July 17, 2006; revised November 26, 2006, February 5, 2007, and June 5, 2007. This paper was recommended by Associate Editor Prof. K. Chakrabarty.

H.-G. Stratigopoulos is with the TIMA Laboratory/CNRS, 38031 Grenoble, France (e-mail: haralampos.stratigopoulos@tima.fr).

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

Digital Object Identifier 10.1109/TCAD.2007.907232

<sup>1</sup>Machine learning was first used for fault-diagnosis purposes [21]–[26]; however, this field of application suffers from the lack of representative and widely accepted analog fault models.

<sup>2</sup>Guard banding is the practice of adjusting specification limits (pass/fail criteria) to account for uncertainty in the measurement system. In this paper, guard bands are in the form of decision hypersurfaces allocated in a measurement space.

of specification testing or defer a decision that involves risk. In the latter case, the neural system forwards the device to specification testing in order to reach an accurate decision. Overall, with appropriate extraction of measurement spaces, the number of devices that need to be retested is minimized; thus, the average test cost per device is lower than the cost of specification testing. Moreover, by exploring various measurement spaces and by varying the desired levels of confidence, this two-tier method enables the exploration of the tradeoff between test accuracy and test cost, and therefore, it can be used to develop cost-effective test plans.

The remaining parts of this paper are organized as follows. In the next section, we review the machine-learning-based test methods, and we explain their inherent limitations in more detail. In Section III, we provide an overview of the proposed two-tier test scheme. In Section IV, we introduce the use of guard bands to assess the confidence of the machine-learning-based test decisions. In Section V, we present the topology of an ontogenic neural network, its training algorithm, and its utilization for the allocation of effective guard bands. In Section VI, we show the structure of the complete neural system. In Section VII, we discuss the extraction of measurement spaces and the subsequent selection of effective subspaces using a genetic algorithm (GA). Experimental results are provided in Section VIII, demonstrating the methodology on two devices—a switched-capacitor filter and an ultrahigh-frequency (UHF) receiver front end.

## II. MACHINE-LEARNING-BASED TESTING

Test methods based on machine learning explore two directions. In the first direction [9]–[20], the training set is used to derive the functions that map the pattern of low-cost measurements to the performance parameters of the device. The functions are approximated using multivariate adaptive regression splines (MARS) [27] or multilayer-perceptron networks [28]. In the second direction [2]–[8], the devices in the training set are projected in the space of low-cost measurements, and a hypersurface is allocated to separate the nominal ( $A^n$ ) from the faulty ( $A^f$ ) region. The hypersurface is used as a decision boundary for testing a new device: If the footprint of its measurement pattern falls in  $A^n$  ( $A^f$ ), then it is classified as nominal (faulty). In essence, this test hypersurface encodes the specification tests and performs them in parallel.

The scatter plot in Fig. 1 shows the projection of a training set of devices in a 2-D measurement space  $x_1 - x_2$ . Each data point represents the measurement pattern  $(x_1, x_2)$  of one device in the set. Suppose now that a test hypersurface is learned in  $x_1 - x_2$ , as shown in Fig. 1(a). For the reasons explained next, and regardless of the machine-learning technique that is employed to learn the test hypersurface, this test approach is bound to have a nonzero test error.

First, as shown in Fig. 1(a), there exist areas around the test hypersurface wherein the distributions of nominal and faulty devices overlap. Overlapping occurs due to the following: 1) the intricate correlation between the measurements and the performance parameters (a closed-form relation does not exist); 2) noise; and 3) drift in the test equipment that is used to



Fig. 1. (a) Test hypersurface and (b) guard-band allocation in a 2-D measurement space  $x_1 - x_2$  extracted from the response of the switched-capacitor filter shown in Fig. 6 when it is driven by band-limited white noise.

characterize the devices. Overlapping areas cannot be assigned with confidence to any of the two classes; thus, new devices whose pattern falls in such areas are subject to misclassification. Moreover, it can be observed that there exist areas wherein the measurement patterns are sparsely distributed. In sparse subspaces, the segments of the hypersurface are randomly shaped since there is little information to guide the curvature. As a result, devices whose pattern falls close to the test hypersurface, in subspaces that were empty during training, are also subject to misclassification. Sparse subspaces could be a side effect of the following: 1) a finite-sized training set; 2) a nonrepresentative training set; or 3) the projection of a finite training set in a high-dimensional measurement space (this phenomenon is commonly referred to as curse of dimensionality [28]).

Similar arguments can be made when regression is used to map the measurement pattern to the performance parameters of the device. In this case, the overlap of the nominal and faulty patterns in a measurement space corresponds to a large

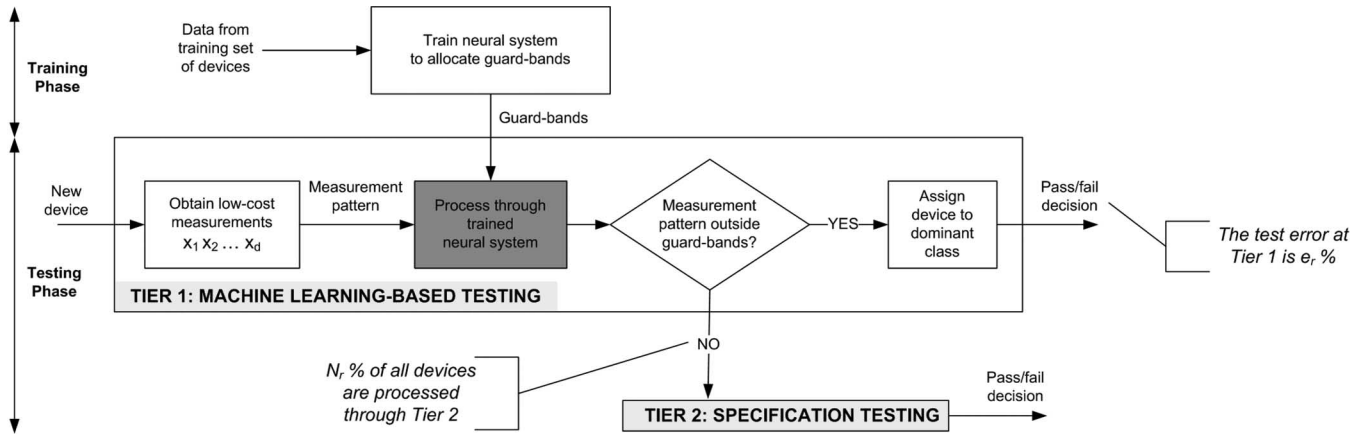


Fig. 2. Flow diagram of the proposed two-tier test scheme.

variance in performance-parameter values for similar measurement patterns, and the sparsely distributed subspace problem is equivalent to having few samples available for regression.

In extensive simulations with various devices and measurement spaces, the test error (or misclassification) of hypersurface-based tests has never been reported to be below 2% [2]–[8]. In regression-based tests [9]–[20], the error is defined in terms of the prediction accuracy of the regression functions and the correlation coefficients between the measured and predicted performance-parameter values. The error of the subsequent classification step, where the predicted values are compared to the specification limits promised in the data sheet, is usually omitted. In [20], it is shown by using real data from a high-performance WLAN 802.11a/b/g radio transceiver that the test error can be as high as 14% when considering the predicted value of the chain gain parameter. Regression-based test methods have the comparative advantage that they provide a prediction of the individual performance parameters, thus allowing diagnosis and multibinning.

However, if go/no-go testing, which is essentially a binary classification problem, is the primary objective, then solving an intermediate harder problem (i.e., regression) entails a possible loss of pertinent information available in the training data. Thus, intuitively, when regression is used as the underlying learning method, it is expected that the test error will be of at least similar magnitude to the error of a test hypersurface. A comparative study to examine the test error when using various machine-learning methods is outside the scope of this paper.

In short, to date, it has not been possible to identify a low-cost measurement pattern for any analog/RF device, which, when processed by even the most powerful learning machines, results in an acceptable test-error rate for industrial standards. Thus, additional effort is necessary in order to capitalize on the low cost of machine-learning-based testing and to make this approach competitive, in terms of test accuracy, to specification testing.

### III. OVERVIEW OF TWO-TIER TEST SCHEME

In this paper, we propose to locate the ambivalent areas in the measurement space in order to identify the devices for

which the machine-learning-based test decisions are prone to error. In particular, we propose to allocate guard bands such that the measurement space is partitioned into three regions: two regions of predominantly nominal and faulty devices, respectively, and a zone interjected in between that contains a mixed distribution. Fig. 1(b) shows a possible allocation of guard bands in the measurement space  $x_1 - x_2$ , with the gray-shaded area representing the guard-banded zone. The nominal (faulty) guard band has the entire nominal (faulty) population on one side, i.e., it guards the nominal (faulty) population.

The guard bands facilitate a two-tier test scheme, as shown in Fig. 2. All fabricated devices go through the first tier, where the low-cost pattern of measurements is obtained. The position of the guard bands in the corresponding measurement space is encoded in the neural system through a training routine, which is executed prior to the testing phase and only once for any production run. During the testing phase, the neural system examines the relative location of the footprint of the measurement pattern with respect to the guard-banded zone. If it falls outside the guard-banded zone, then the device is assigned to the respective class, i.e., the neural system infers the results of specification testing from the measurement pattern, with low test error  $\epsilon_r$ . Otherwise, if it falls in the guard-banded zone, the device is deemed suspect to misclassification, and the neural system suggests that further action be taken. In this case, the device is directed to the second tier, where it is retested through the standard specification testing in order to reach an accurate decision.

By enlarging the area of the guard-banded zone, the test error  $\epsilon_r$  of the first tier is reduced at the expense of retesting more devices. In the two limits, the guard-banded zone contains the entire device distribution, or the guard bands merge onto the test hypersurface of Fig. 1(a). Thus, if  $\epsilon_r'$  denotes the error of the test hypersurface and  $N_r$  denotes the percentage of devices that go through the second tier, then  $\epsilon_r$  drops from  $\epsilon_r'$  to zero as  $N_r$  increases. In practice, in a discriminative measurement space, the guard bands can be allocated such that  $\epsilon_r$  approaches zero when a small fraction  $N_r$  of devices are retested.

Now, let  $C_i$  and  $T_i$  denote the test cost per second and the test time of the  $i$ th tier, respectively. Let also  $C_s$  and  $T_s$  denote the test cost per second and the test time, respectively, when

the standard specification testing is applied to every fabricated device. The average test cost per device for the proposed approach can be modeled as

$$C = C_1 \cdot T_1 + C_2(N_r) \cdot T_2(N_r). \quad (1)$$

By using a first-order Taylor approximation,  $C_2(N_r)$  can be written as

$$\begin{aligned} C_2(N_r) &= C_2(N_r = 1) + \left. \frac{dC_2(N_r)}{dN_r} \right|_{N_r=1} \cdot (N_r - 1) \\ &= C_s + C_h + \left. \frac{dC_2(N_r)}{dN_r} \right|_{N_r=1} \cdot (N_r - 1) \\ &= C_s + C_h + C^*(N_r) \end{aligned} \quad (2)$$

where  $C_h$  is the test cost per second overhead from handlers that are used to transfer the devices to the tester in the second tier, and

$$C^*(N_r) = \left. \frac{dC_2(N_r)}{dN_r} \right|_{N_r=1} \cdot (N_r - 1) \leq 0. \quad (3)$$

The test time of the second tier can be expressed as

$$\begin{aligned} T_2(N_r) &= N_r \cdot T_s \\ &= N_r \cdot (T_s^e + T_h) \end{aligned} \quad (4)$$

where  $T_s^e$  is the electrical specification test time, and  $T_h$  is the handling time spent to transfer the devices to the tester in the second tier. Equation (1) now becomes

$$C = C_1 \cdot T_1 + N_r \cdot (C_s + C_h + C^*(N_r)) \cdot (T_s^e + T_h). \quad (5)$$

Note that  $C_h = T_h = 0$  if the devices do not need to be transferred to another tester to undergo specification testing.

Therefore, if the inequality

$$\frac{C_1 \cdot T_1 + N_r \cdot (C_h + C^*(N_r)) \cdot (T_s^e + T_h)}{1 - N_r \cdot \frac{T_s^e + T_h}{T_s}} < C_s \cdot T_s \quad (6)$$

holds, then

$$C < C_s \cdot T_s \quad (7)$$

which means that the test cost is reduced while maintaining the accuracy of the specification testing.

As previously hinted, given a measurement space,  $\epsilon_r$  and  $N_r$  can be traded off by varying the area of the guard-banded zone. Thereby, and through the exploration of various measurement spaces, a tradeoff curve between  $\epsilon_r$  and  $C$  is drawn. This curve allows test engineers to devise cost-effective test plans that target different test-quality objectives.

#### IV. GUARD-BAND ALLOCATION

Guard banding has been mentioned in [11] as also an option for the regression-based methods. Therein, guard bands are defined as a percentile deviation from the device-specification limits, but no experimental data is reported regarding the resulting percentage of retested devices. In addition, within the

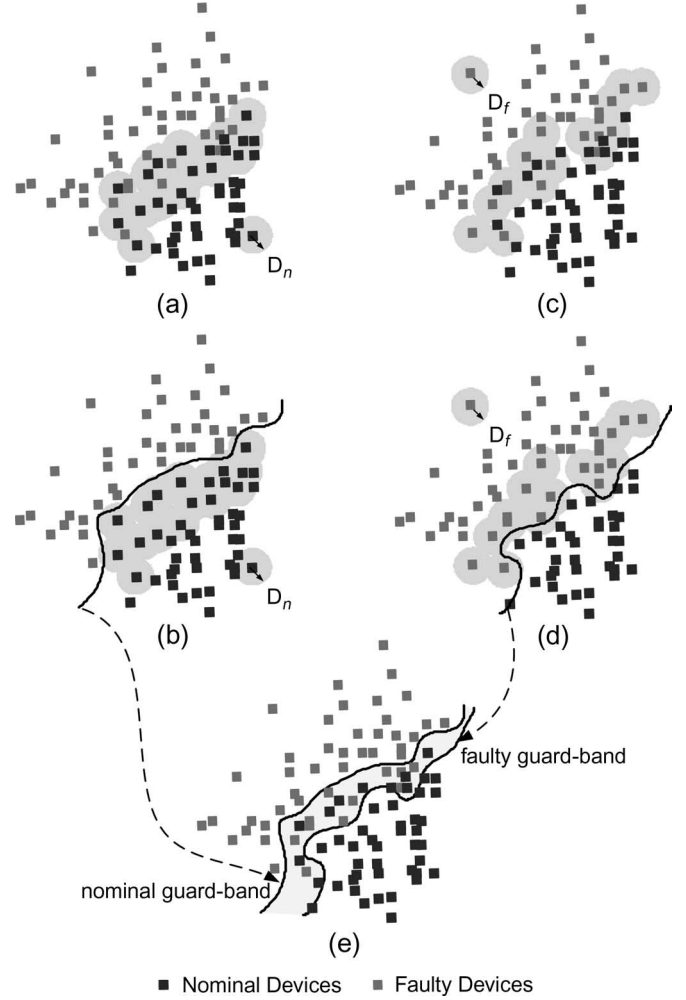


Fig. 3. Steps in the guard-band allocation.

context of specification-test compaction [29], guard bands are allocated by perturbing the entire test hypersurface by a predefined distance, thus creating a guard-banded zone of constant width. This rigidity of the guard-band allocation method might inadvertently enclose areas with nonoverlapping populations, resulting in an unnecessarily large percentage of the retested devices. Instead, in the proposed method, the guard bands are viewed as independent decision boundaries and, thus, are allocated regardless of the position of the test hypersurface.

Each guard band is allocated separately to perfectly classify all the training patterns of the guarded class and, under this constraint, to provide an optimum classification for the training patterns of the opposite class. Without loss of generality, consider the allocation of the nominal guard band, which is shown on the left-hand side of Fig. 3. First, we draw hyperspheres of radius  $D_n$  centered at nominal training patterns, as shown in Fig. 3(a). The radius  $D_n$  is defined as

$$D_n = \frac{1}{N_n} \sum_{q \in C_n} \min_{p \in C_f} \|\bar{x}^q - \bar{x}^p\| \quad (8)$$

where  $\bar{x}^k$  is the measurement pattern of training instance  $k$ ,  $C_n$  and  $C_f$  denote the nominal and faulty classes, respectively,  $\|\cdot\|$  is the Euclidian norm, and  $N_n$  is the number of nominal patterns

in the training set. Faulty training patterns  $p$  are successively paired with all nominal training patterns  $q$ , and if they lie within distance  $D_n$  from the nominal training patterns, i.e., if the inequality

$$\|\bar{x}^q - \bar{x}^p\| < D_n \quad (9)$$

holds, then they are temporarily excluded from the training set, as shown in Fig. 3(b). After the faulty training patterns have been cleared out of the overlapping areas, the ontogenic neural network described in detail in Section V is employed to allocate the nominal guard band, which is shown in Fig. 3(b). The dual procedure, shown in Fig. 3(c) and (d), is followed to allocate the faulty guard band using a distance  $D_f$ , which is defined similarly to  $D_n$  in (8). The guard-banded zone is the area enclosed between the two guard bands, which is gray-shaded in Fig. 3(e).

## V. ONTOGENIC NEURAL NETWORK

The guard bands are, in essence, decision boundaries for testing new devices. In Section V-A, we refer to previous work on this topic, pointing out the limitations that motivate our choice of the ontogenic neural network. Next, in Section V-B, we present the topology of the ontogenic neural network. In Section V-C, we discuss its training algorithm, and in Section V-D, we show a heuristic inductive principle to achieve optimal generalization.

### A. Decision Boundaries for Testing Devices

In the past, decision boundaries have been allocated using Fisher's linear discriminants [2], logistic discrimination analysis [4], linear-perceptron networks [3], [5], feedforward neural networks with sigmoidal hidden units [6], [7], and polynomial kernel transformations [8].

In [2]–[5], the problem is decomposed into  $M$  two-class separation problems that are solved individually, where  $M$  is the number of single-ended device specifications. In particular, as shown in Fig. 4(a), for each single-ended specification  $\mu$ , a hyperplane  $b_\mu$  is allocated in the measurement space such that the maximum possible separation between the nominal and faulty instances in the training set (with respect to specification  $\mu$ ) is obtained. In essence, each hyperplane  $b_\mu$  creates two regions  $A^n(\mu)$  and  $A^f(\mu)$ : Instances that fall into  $A^n(\mu)$  ( $A^f(\mu)$ ) are classified as nominal (faulty) with respect to specification  $\mu$ . Note that this approach assumes single convex decision regions, which, as discussed in [8], is not always the case. The overall acceptance region  $A^n$  is then approximated by the union  $\bigcup_{\mu=1}^M A^n(\mu)$ , which is bounded by the hyperplane segments. However, the decision boundaries are, in general, nonlinear [for example, in Fig. 4(a), the optimal decision boundary is an ellipsoid], and thus, a crude approximation with hyperplanes results in an error. Furthermore, there is an additional error factor resulting from individually optimizing the location of each of the  $M$  decision boundaries. In particular, each decision boundary is allocated such that it minimizes misclassification throughout the entire distribution of measurement patterns.

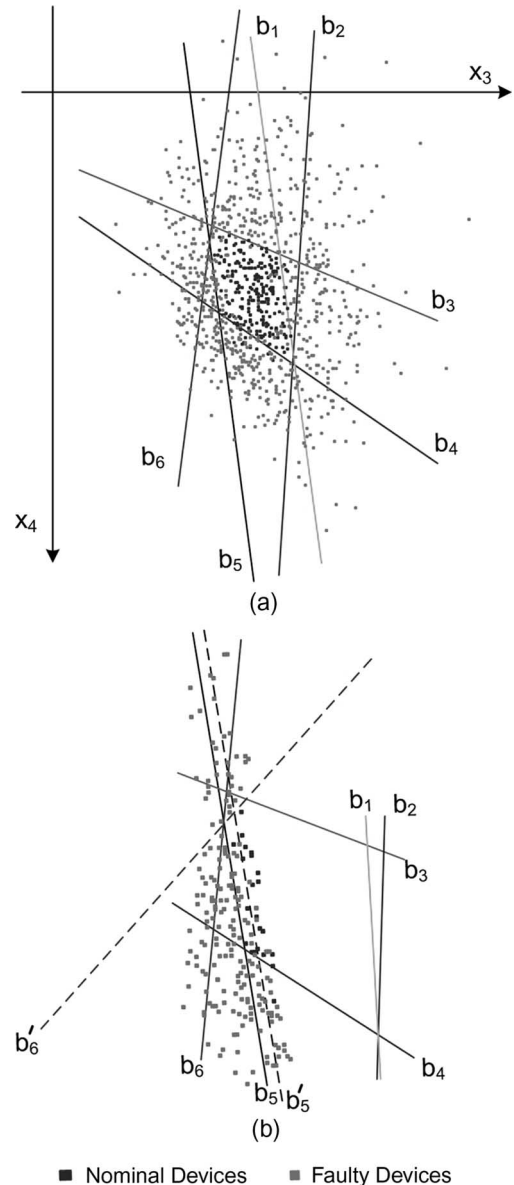


Fig. 4. Projection of instances of a state variable filter with six single-ended specifications ( $M = 6$ ) on a 2-D measurement space  $x_3 - x_4$ . (a) Piecewise linear approximation of the decision boundary. (b) Individual allocation of decision boundaries induces error.

Thus, it also tries to minimize misclassification in areas that are distant from the acceptance region, which is interpreted as mapping the measurement patterns into the proper faulty class. This, however, is unnecessary, and in addition, it affects the positioning of the segments of the hyperplanes around which the unique nominal class is separated from the  $2^M - 1$  faulty classes. To view this, consider Fig. 4(b), where Fig. 4(a) is redrawn showing the boundary of the acceptance region and the measurement patterns distributed across boundaries  $b_5$  and  $b_6$  only. It can be seen that there exist boundaries  $b'_5$  and  $b'_6$  which would yield a better classification with regard to all specifications. Yet,  $b_5$  and  $b_6$  are chosen in place of  $b'_5$  and  $b'_6$  because, with respect to individual specifications, they provide a better classification throughout the measurement space.

In [8], polynomial hypersurfaces are allocated in the measurement space instead of hyperplanes, but the error resulting

from the individual allocation is still not addressed. In [6] and [7], a feedforward neural network is used that can potentially draw decision boundaries of arbitrary order without the need to decompose the problem. However, the network topology (i.e., the number of hidden units) that generalizes well on new device instances is not known *a priori* and can only be identified empirically by a trial-and-error procedure that requires significant computational effort.

As opposed to [6] and [7], the ontogenic neural network that we chose to use is constructed successively, in a way that enables it to adaptively acquire the necessary connectivity that produces a decision boundary of appropriate order.

### B. Topology

The neural network is trained using data from a set of device instances, which is denoted by  $S_t$ . The performance parameters of each device  $k \in S_t$  are measured explicitly in order to associate it with a status bit  $t^k$ , where  $t^k = +1$  if  $k$  is nominal, i.e.,  $k \in C_n$ , and  $t^k = -1$  if  $k$  is faulty, i.e.,  $k \in C_f$ . Each instance  $k \in S_t$  is also associated with a  $d$ -dimensional measurement pattern  $\vec{x}^k \in \mathbf{R}^d$ . The training set  $(\vec{x}^1, t^1), (\vec{x}^2, t^2), \dots, (\vec{x}^{|S_t|}, t^{|S_t|})$  is used to optimize the adaptive parameters of the neural network. The classification error on the training set is defined as

$$E_{S_t} = \frac{1}{|S_t|} \sum_{k \in S_t} h^{\text{EC}}(\vec{x}^k) \quad (10)$$

where  $h^{\text{EC}}$  is the error counting function:  $h^{\text{EC}}(\vec{x}^k) = 1$  if the pattern  $\vec{x}^k$  is misclassified, and  $h^{\text{EC}}(\vec{x}^k) = 0$  otherwise.

As explained in Section V-A, since the order of the decision boundary is not known *a priori*, assuming a fixed network topology limits the range of feasible boundaries. Instead, the proposed neural network learns the boundary constructively, starting with the input layer and dynamically adding layers (ontogenicity) until it matches the intrinsic complexity of the problem at hand. A comprehensive discussion on constructive algorithms for neural networks can be found in [30].

In particular, the proposed neural network is constructed using the 1-pyramid algorithm [31] that successively places layers of single neurons above the existing ones. The first neuron  $y_1$  receives inputs from the  $d$  measurements. Each successive neuron  $y_i$  receives inputs from the  $d$  measurements and from each neuron below itself. In order for the algorithm to handle the real-valued measurements, each neuron above the first layer also receives an extra attribute that is the projection of the  $d$ -dimensional measurement vector onto a parabolic surface

$$x_{d+1} = \sum_{i=1}^d x_i^2. \quad (11)$$

Each newly added neuron takes over the role of the output neuron, and the network growth continues until a satisfactory solution for the learning problem is found. The complete architecture of the network is shown in Fig. 5.

The neuron model used herein is an  $\ell$ -input threshold logic unit, also known as perceptron [28], that computes the thresh-

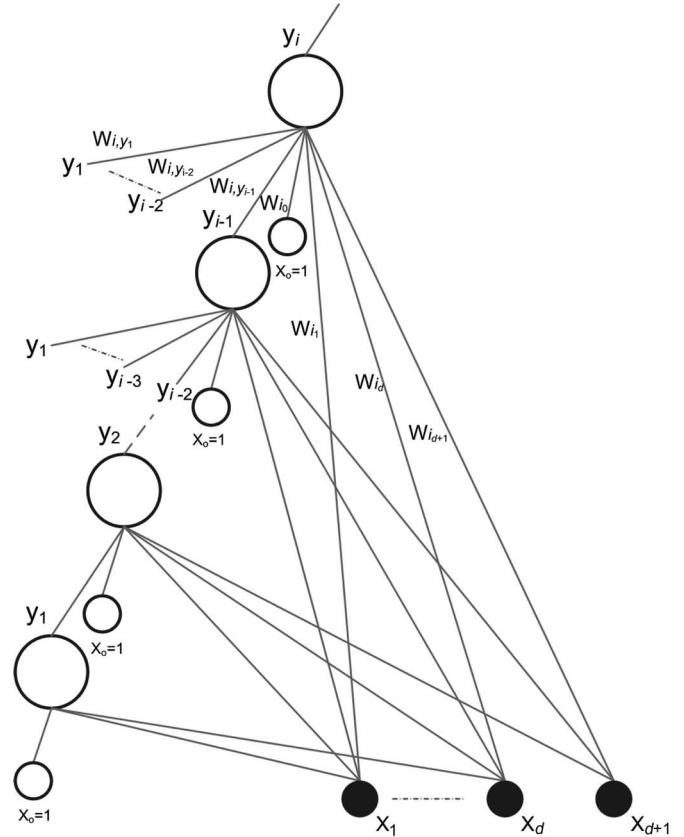


Fig. 5. Topology of the ontogenic neural network.

old function of the weighted sum of its inputs  $\vec{v}_i \in \mathbf{R}^\ell$ :  $y_i(\vec{v}_i) = -1$  for  $\vec{w}_i^T \vec{v}_i < 0$  and  $y_i(\vec{v}_i) = +1$  for  $\vec{w}_i^T \vec{v}_i \geq 0$ .  $\vec{w}_i^T = [w_{i_0}, w_{i_1}, \dots, w_{i_\ell}]$  is the adaptive weight vector, and the weight  $w_{i_0}$  is referred to as the bias. Here,  $\vec{v}_1 = \vec{x}$ ,  $\vec{w}_1^T = [w_{1_0}, w_{1_1}, \dots, w_{1_d}]$  and  $\vec{v}_i = (\vec{x}, x_{d+1}, y_{i-1}, \dots, y_1)$ ,  $\vec{w}_i^T = [w_{i_0}, \dots, w_{i_{d+1}}, w_{i, y_{i-1}}, \dots, w_{i, y_1}]$  for  $i > 1$ . Since  $\vec{v}_i$  is a function of  $\vec{x}$ , in the following, we use  $y_i(\vec{x}^k) = y_i(\vec{v}_i^k(\vec{x}^k))$  to denote the output of the neural network at layer  $i$  when the measurement pattern  $\vec{x}^k$  is applied at its inputs. Now, let  $y_i(\vec{x}^k) = +1$  and  $y_i(\vec{x}^k) = -1$  refer to pass and fail decisions, respectively, for instance  $k$ . Then, we want to select weights such that  $\vec{w}_i^T \vec{v}_i^k < 0$  for all  $\vec{x}^k \in C_f$  and  $\vec{w}_i^T \vec{v}_i^k \geq 0$  for all  $\vec{x}^k \in C_n$ .

The perceptron has a simple geometrical representation. It divides linearly its input space by a hyperplane, which is composed by the set of solutions to equation  $\vec{w}_i^T \vec{v}_i = 0$ , such that its output  $y_i$  is  $+1$  on one side of the hyperplane and  $-1$  on the other side. Because of the extra attribute  $x_{d+1}$  in (11) and the input from the preceding neurons, this hyperplane translates into a nonlinear hypersurface, denoted by  $f_i$ , when it is projected in the original  $d$ -dimensional space of measurements. Therefore, nonlinear decision boundaries are formed by training a sequence of linear perceptrons. This property is very useful since, as we discuss in Section VII, it allows the use of this network for a fast evaluation of measurement spaces and selection of subspaces in an optimization framework.

*Theorem 1:* Let  $f$  be the optimal decision boundary at which  $E_{S_t} = 0$ . The aforementioned constructive algorithm produces

a sequence of decision boundaries  $\{f_i\}$  that, in the limit, converges to  $f$ , i.e.,  $\lim_{i \rightarrow \infty} \|f - f_i\| = 0$ .

*Proof:* For each pattern  $\bar{x}^p$ , define

$$\varepsilon_p = \frac{1}{2} \cdot \min_{q \neq p} \sum_{i=1}^d (x_i^p - x_i^q)^2$$

$$k = \max_{p,q} \sum_{i=1}^d (x_i^p - x_i^q)^2 > \varepsilon_p.$$

Suppose that pattern  $\bar{x}^p$  is misclassified at layer  $(i-1)$ , i.e.,  $y_{i-1}(\bar{x}^p) = -t^p$ . Then, if we select the following weights

$$w_{i_0} = t^p \left( k + \varepsilon_p - \sum_{j=1}^d (x_j^p)^2 \right)$$

$$w_{i_j} = 2t^p x_j^p, \quad j = 1, \dots, d$$

$$w_{i_{d+1}} = -t^p$$

$$w_{i, y_{i-1}} = k$$

$$w_{i, y_j} = 0, \quad j = i-2, i-3, \dots, 1$$

the net input of the  $i$ th neuron is

$$\begin{aligned} \vec{w}_i^T \vec{v}_i^p &= w_{i_0} + \sum_{j=1}^{d+1} w_{i_j} x_j^p + \sum_{j=1}^{i-1} w_{i, y_j} y_j(\bar{x}^p) \\ &= t^p \left( k + \varepsilon_p - \sum_{j=1}^d (x_j^p)^2 \right) + \sum_{j=1}^d 2t^p (x_j^p)^2 \\ &\quad - t^p \sum_{j=1}^d (x_j^p)^2 + k y_{i-1}(\bar{x}^p) \\ &= t^p \varepsilon_p. \end{aligned}$$

Since  $\varepsilon_p > 0$ , the pattern  $\bar{x}^p$  is correctly classified by the new layer  $i$ . Consider now any pattern  $\bar{x}^q \neq \bar{x}^p$  that is correctly classified at layer  $(i-1)$ , i.e.,  $y_{i-1}(\bar{x}^q) = t^q$ . Then

$$\begin{aligned} \vec{w}_i^T \vec{v}_i^q &= w_{i_0} + \sum_{j=1}^{d+1} w_{i_j} x_j^q + \sum_{j=1}^{i-1} w_{i, y_j} y_j(\bar{x}^q) \\ &= t^p \left( k + \varepsilon_p - \sum_{j=1}^d (x_j^p)^2 \right) + \sum_{j=1}^d 2t^p x_j^p x_j^q \\ &\quad - t^p \sum_{j=1}^d (x_j^q)^2 + k y_{i-1}(\bar{x}^q) \\ &= t^p (k + \varepsilon_p - \varepsilon') + k t^q \\ &= t^q \left( \frac{t^p}{t^q} k' + k \right) \end{aligned}$$

where  $\varepsilon' = \sum_{i=1}^d (x_i^p - x_i^q)^2 > \varepsilon_p$ , and  $k' = k + \varepsilon_p - \varepsilon'$ . Since  $0 < k' < k$ , the pattern  $\bar{x}^p$  continues to be classified correctly after the addition of layer  $i$ . Therefore, there exist

weights that will reduce  $E_{S_t}$  whenever a new layer is added to the network. Since the number of training patterns is finite, eventual convergence to  $E_{S_t} = 0$  is guaranteed. In the following section, we discuss a training algorithm that generates such weights. ■

### C. Training a Layer

The distributions of nominal and faulty training measurement patterns are separable at layer  $i$  if the following condition holds:

$$(\vec{w}_i^T \vec{v}_i^k) t^k > 0 \quad \forall k. \quad (12)$$

In order to reduce  $E_{S_t}$  at layer  $i$ , (12) suggests that we select a weight vector  $\vec{w}_i$  that minimizes the following error function, which is known as perceptron criterion:

$$E^{\text{perc}}(\vec{w}_i) = - \sum_{k \in S_t: y_i(\bar{x}^k) \neq t^k} (\vec{w}_i^T \vec{v}_i^k) t^k. \quad (13)$$

Here, the summation is over all patterns in the training set, which are misclassified by the current weight vector  $\vec{w}_i$ . 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 space of weights is performed by applying the thermal perceptron learning rule [32]

$$w_{i_j}^{(\tau+1)} = w_{i_j}^{(\tau)} + \frac{\alpha}{2} \vec{v}_j^k (t^k - y_i(\bar{x}^k)) e^{-\frac{|\vec{w}_i^T \vec{v}_i^k|}{T}} \quad (14)$$

where  $\alpha > 0$ . This corresponds to a simple learning procedure: We cycle through all patterns in the training set and test each pattern, in turn, using the current set of weight values. If the pattern  $\bar{x}^k$  is correctly classified, then we proceed to the next; otherwise, we add  $\alpha \vec{v}_j^k e^{-|\vec{w}_i^T \vec{v}_i^k|/T}$  to the current weight vector if  $\bar{x}^k \in C_n$ , or we subtract  $\alpha \vec{v}_j^k e^{-|\vec{w}_i^T \vec{v}_i^k|/T}$  if  $\bar{x}^k \in C_f$ . This procedure successively reduces the error in (13) [28].

The exponential tail in (14) controls the correction of weights based on the location of the misclassified pattern  $\bar{x}^k$  with respect to the decision boundary.  $|\vec{w}_i^T \vec{v}_i^k|$  is a measure of this distance. In turn, the temperature  $T$  controls how strongly the changes are attenuated for large values of  $|\vec{w}_i^T \vec{v}_i^k|$ . As an intuition, one can imagine a zone surrounding the decision boundary. The boundary moves only if an erroneously classified pattern falls within this zone. The temperature is annealed from an initial value  $T_0$  to zero, causing a gradual reduction of the extent of the sensitive zone. In the limit of  $T \rightarrow 0$ , the zone disappears altogether, and the perceptron is stable, i.e., its training has been completed. Best results are obtained when  $\alpha$  is reduced at the same time as  $T$  is (see [32] for the rationale supporting this approach).

The thermal learning rule outperforms other known perceptron-based learning algorithms [33], provided that the temperature is chosen appropriately. In particular,  $T$  should be of the same order of magnitude as the range of values of  $\vec{w}_i^T \vec{v}_i^k$ . We followed the suggestion in [34].  $T$  decreases from  $T_0$  (initially 1) to zero during 500 cycles through the training set. Since  $\vec{w}_i^T \vec{v}_i^k$  might vary considerably for different device

instances, we calculate the average value of  $|\vec{w}_i^T \vec{v}_i^k|$  over the set of device instances  $\langle |\vec{w}_i^T \vec{v}_i^k| \rangle_k$  over each cycle. At the end of each cycle,  $T_o$  is set to  $T_o = (2T_o + 2\langle |\vec{w}_i^T \vec{v}_i^k| \rangle_k)/3$ . The temperature  $T$  is then set to  $\gamma T_o$ , where  $\gamma$  (initially 1) decreases linearly with each cycle to reach zero after 500 cycles.  $\alpha$  is set to  $0.1\gamma$ .

#### D. Training the Network

The network-training procedure corresponds to an iterative reduction of  $E_{S_t}$ . However, as training progresses and new layers are added, there comes a point where the network starts to overfit the training data. This can be observed by examining the classification error on an independent set of devices, which at first keeps decreasing and then starts increasing. The ability of the network to correctly classify previously unseen device instances, other than those included in  $S_t$ , is called generalization. In order to find the effective complexity of the network, such that it achieves the best possible generalization, we follow an early stopping inductive principle. More specifically, during training, the generalization at each layer is monitored on a second independent set of device instances (holdout set) denoted by  $S_h$ , and after training is complete, the network is pruned down to the layer that scores the best generalization. At this layer, an unbiased estimate of the generalization is computed on a third independent set of devices (test set) denoted by  $S_{te}$ . Since, in our case, the decision boundary is used as a guard band, the generalization is measured on the device instances belonging to the class that is being guarded. For the nominal guard band, the generalization error is estimated as

$$\begin{aligned} \hat{P}_{S_{te}}^n &= \frac{1}{|S_{te}|} \sum_{\substack{k \in S_{te} \\ k \in C^n}} h^{EC}(\vec{x}^k) \\ &= \frac{1}{|S_{te}|} \sum_{\substack{k \in S_{te} \\ k \in C^n}} \left( \frac{1 - y_{\text{eff}}(\vec{x}^k)}{2} \right) \end{aligned} \quad (15)$$

where  $y_{\text{eff}}$  denotes the output of the layer that has scored the best generalization on  $S_h$  during training. The set of devices in  $S_{te}$ , whose measurement pattern falls in the nominal region, can be expressed as

$$S_{te}^n = \{k \in S_{te} : y_{\text{eff}}(\vec{x}^k) = 1\}. \quad (16)$$

By analogy, the generalization error for the faulty guard band is given by

$$\begin{aligned} \hat{P}_{S_{te}}^f &= \frac{1}{|S_{te}|} \sum_{\substack{k \in S_{te} \\ k \in C^f}} h^{EC}(\vec{x}^k) \\ &= \frac{1}{|S_{te}|} \sum_{\substack{k \in S_{te} \\ k \in C^f}} \left( \frac{1 + y_{\text{eff}}(\vec{x}^k)}{2} \right) \end{aligned} \quad (17)$$

and the set of devices in  $S_{te}$ , whose measurement pattern falls in the faulty region, is

$$S_{te}^f = \{k \in S_{te} : y_{\text{eff}}(\vec{x}^k) = -1\}. \quad (18)$$

## VI. NEURAL SYSTEM

The neural system in Fig. 2 comprises a committee of two ontogenic neural networks that allocate the nominal and faulty guard bands. It requires  $O(L^2 + L \cdot d)$  computations to examine the relative position of a measurement pattern with respect to the guard bands, where  $L$  is the number of perceptrons in the ontogenic neural networks, and  $d$  is the dimensionality of the input measurement pattern. The process time of the neural system is a very small fraction of the total test time  $T_1$  of the first tier. Estimates of  $N_r$  and  $\epsilon_r$  are computed on  $S_{te}$ . In particular,  $N_r$  is equal to the percentage of devices in  $S_{te}$  whose measurement pattern falls in the guard-banded zone

$$N_r = \frac{|S_{te}^f \cap S_{te}^n|}{|S_{te}|} \quad (19)$$

and  $\epsilon_r$  is equal to the sum of the generalization errors of the two guard bands

$$\epsilon_r = \hat{P}_{S_{te}}^f + \hat{P}_{S_{te}}^n. \quad (20)$$

Evidently, given a measurement space, the aforementioned tradeoff between  $\epsilon_r$  and  $N_r$  can be explored by using distances  $\lambda_f D_f$  and  $\lambda_n D_n$  to clear the overlapping areas and by varying  $\lambda_f$  and  $\lambda_n$  around one. As we increase  $\lambda_f$  or  $\lambda_n$ ,  $N_r$  increases, and  $\epsilon_r$  decreases. In particular, for every measurement space, there exist  $\lambda_f$  and  $\lambda_n$  such that  $\epsilon_r = 0$ . More tradeoff points can be collected by repeating this procedure for various measurement spaces. Note that the neural system is unbiased, i.e.,  $\epsilon_r > 0$  corresponds to both test escapes and yield loss. If test-escape elimination is of higher importance, a larger  $\lambda_f$  can be used such that the faulty guard band is pushed deeper into the nominal region. In this case, the tradeoff is obtained by only varying  $\lambda_n$ .

## VII. MEASUREMENT-SPACE EXTRACTION

The effectiveness of the two-tier test scheme is measured by three parameters, namely, the test cost of the first tier ( $C_1 \cdot T_1$ ), the test error of the first tier ( $\epsilon_r$ ), and the percentage of devices that go through the second tier ( $N_r$ ). These three parameters, in turn, depend on the choice of the measurement pattern.

In order to guarantee a low test time  $T_1$ , the measurement pattern should be low dimensional and should be extracted by switching the device to a minimum number of test configurations, preferably only one. In order to guarantee a low test cost per second  $C_1$ , the measurement pattern should be extracted using a low-cost assortment of test equipment. To satisfy this objective for multigigahertz RF devices, it is necessary to avoid the use of expensive RF testers and to interface the methodology to the existing mixed-signal test equipment [35]. For example, in [10] and [19], the authors apply the concept of modulation and demodulation to translate a baseband test stimulus to the RF spectrum and to convert the response back to a baseband signature. In [12], it is proposed to undersample the RF response using a noise reference and, then, to obtain the Fourier harmonics in the spectrum. In [13] and [16], the authors embed sensors (e.g., peak and rms detectors,



differential-topology sensors, and recursive sensors) into the RF signal paths to extract dc or low-frequency signals. The dc and low-frequency signals can also be extracted by adding design-for-testability structures on-chip (e.g., loopback paths, offset cancellation digital-to-analog converters implemented at each low-frequency block output, and additional internal probes) [20]. A test configuration for time-division multiple-access RF power amplifiers is proposed in [17], where the transient-current response to a slow ascending ramp signal is captured. A test configuration for RF transceivers is proposed in [18], where the transmitted signal is looped back to the receiver. The measurement space is extracted from the spectrum at the output of the receive mixer. In [36], it is proposed to observe the quiescent current signature when the power supply is ramped in discrete steps.

Achieving low  $N_r$  requires that the measurement space provides adequate discrimination between the nominal and faulty distributions such that the ambivalent areas constitute only a small fraction of it. In contrast,  $\epsilon_r$  depends only on the parameters  $\lambda_f$  and  $\lambda_n$  and the dimensionality of the measurement space. In particular, we only need to circumvent the curse of dimensionality by keeping the ratio of  $|S_t|$  to the dimensionality of the measurement space high.

Given a test configuration, we initially extract  $d_I$  measurements, where  $d_I$  is large in order to increase the probability of extracting useful measurements. Then, we search in the space of candidate measurements to select a subspace of dimensionality  $d < d_I$  that best meets our objective on the tradeoff curve  $\epsilon_r - N_r$ . In particular, we can pursue the minimization of  $N_r$  under the constraint  $\epsilon_r \leq \delta$ ,  $\delta \in [0, \epsilon'_r]$ , where  $\epsilon_r = \epsilon'_r$  for  $N_r = 0$ . Note that since the training set has a finite size,  $\epsilon_r$  and  $N_r$  can take discrete values  $k/|S_t|$ , where  $k \in \mathbb{N}$ ,  $k \leq |S_t|$ . The points on the optimal tradeoff curve are largely known as Pareto-optimal solutions.

Recent comparative studies [37] show that GAs are the most suitable for large-scale measurement selection problems [38]. GAs start with a base population of chromosomes and generate successive populations through an intrinsically parallel search process that mimics the mechanics of natural selection and genetics [39]. In the search for subsets of measurements, a subset is encoded in a chromosome as a  $d$ -element bit string, with the  $i$ th bit denoting the presence or the absence of the  $i$ th measurement. GAs evolve by the juxtaposition of schemata (bit templates), resulting in rapid optimization of the target fitness function. Instead of running a GA many times by using a fitness function that emphasizes one particular Pareto-optimal solution for each time, we use a multiobjective GA, called the NSGA-II [40], in order to find multiple Pareto-optimal solutions in one single simulation run. The NSGA-II uses binary tournament selection, crossover, and mutation operators to generate offspring populations. It also includes elitism and a parameterless diversity-preservation mechanism to ensure a good spread of the Pareto-optimal solutions.

## VIII. EXPERIMENTAL RESULTS

The proposed method is evaluated on a fifth-order elliptic switched-capacitor filter, which is shown in Fig. 6, using syn-

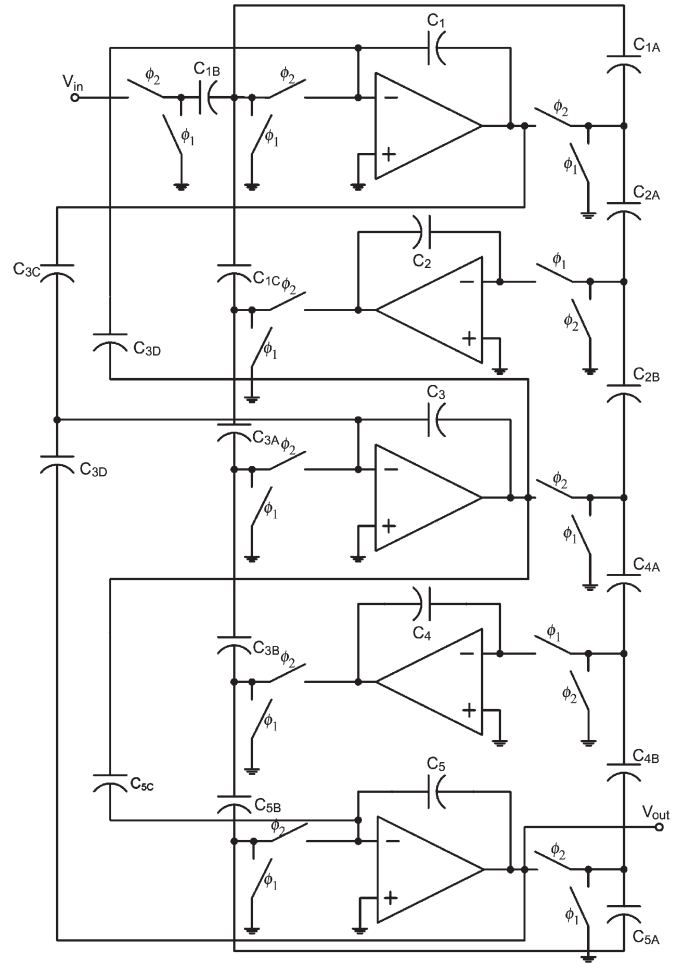


Fig. 6. Ladder realization of the fifth-order elliptic switched-capacitor filter [41].

thetic data from simulation analysis and an off-the-shelf RF device using real data. The studied RF device is a monolithic integrated UHF receiver front end that contains a low-noise amplifier (LNA) and a balanced mixer. The course of each experiment is as follows.

- 1) We start with a representative set of  $N$  device instances. Each instance  $k$ ,  $k = 1, \dots, N$ , undergoes full specification testing in order to associate it with an accurate nominal or faulty label  $t^k$ .
- 2) We select a test configuration, a test stimulus, and an initial set of  $d_I$  measurements.
- 3) We obtain the  $d_I$  measurements on all instances of  $N$ .
- 4) The  $d_I$  measurements are normalized in order to avert skewing of the distance between two measurement patterns in the computation of  $D_f$  and  $D_n$ . Moreover, in practice, normalization speeds up the training phase of the neural system. The normalized measurements are gathered in  $d_I$ -dimensional vectors  $\bar{x}^k$ ,  $k = 1, \dots, N$ .
- 5) The  $N$  device instances are divided into training, holdout, and test sets.
- 6) The NSGA-II algorithm is run to identify the Pareto-optimal subspaces of the  $d_I$ -dimensional measurement space. The parent population of measurement subspaces

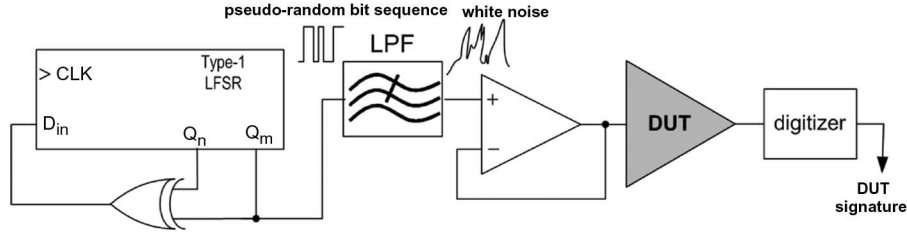


Fig. 7. Test configuration for the switched-capacitor filter of Fig. 6.

at each generation is 200, and the algorithm terminates after 100 generations. The crossover and mutation probabilities are set to 0.9 and  $1/d_I$ , respectively. Each measurement subspace is evaluated by training the neural system in order to estimate  $\epsilon_r$  and  $N_r$ . The parameters  $\lambda_f$  and  $\lambda_n$  are set to one.

- 7) After the NSGA-II converges, for every Pareto-optimal measurement subspace, we retrain the neural system using different values of  $\lambda_f$  and  $\lambda_n$ . We examine all combinations as  $\lambda_f$  and  $\lambda_n$  vary from 0.25 to 1.5 at a 0.25 step. Thus, for every Pareto-optimal measurement subspace, we obtain an additional set of 36 points ( $\epsilon_r$  and  $N_r$ ), which may improve the Pareto-optimal front.
- 8) We plot the tradeoff curve  $\epsilon_r - N_r$  that connects the points of the Pareto-optimal front.

The costs  $C_h$  and  $C^*(N_r)$  in (5) depend on a large number of factors and may vary widely across the industry. In order to provide estimates of the average test cost per device, we adopt a simplified model

$$C = C_1 \cdot T_1 + N_r \cdot C_s \cdot (T_s^e + T_h) \quad (21)$$

which can be deduced from the general model of (5) by setting  $C_h = C^*(N_r) = 0$ . Note that the aforementioned simplified model is not necessarily optimistic since  $C^*(N_r) \leq 0$ , and thus,  $C_h + C^*(N_r)$  could attain a negative value for low  $N_r$ . On the contrary, the simplified model is pessimistic if the devices do not need to be transferred to another tester to undergo specification testing, in which case  $C_h = 0$ .

Next, we describe each of the experiments in detail.

#### A. Switched-Capacitor Filter

We generated  $N = 2000$  instances of the switched-capacitor filter by Monte Carlo analysis, letting various design parameters follow a normal distribution, centered at their nominal values with a 3% standard deviation. The design parameters considered include the switched-capacitor values and the geometry, oxide thickness, threshold voltage, body-effect coefficient, and junction capacitances of the transistors in the op-amps. Catastrophic shorts and opens in the MOS switches are excluded since they generate outlier points in the faulty distribution and, thus, do not affect the positioning of the guard bands.  $N/2$  instances are assigned to the training set, whereas  $N/4$  instances are assigned to each of the holdout and test sets. The performance parameters considered include the ripples in

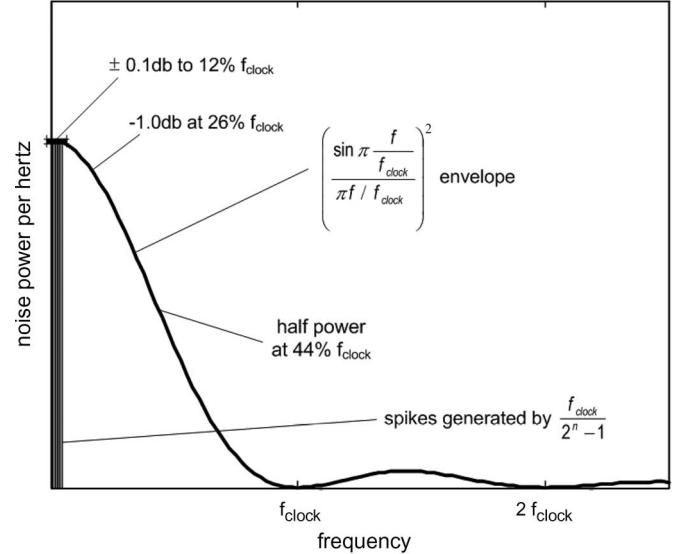


Fig. 8. Power spectrum of the unfiltered LFSR bit sequence output [42].

the passband and stopband, gain errors, group delay, phase response, and total harmonic distortion.

As a test stimulus, we use white noise limited up to a frequency multiple of the bandwidth of the switched-capacitor filter [24]. Intuitively, this is a promising stimulus since it contains infinite tones that can generate persistently exciting response waveforms. The band-limited white noise can be digitally synthesized by passing the pseudorandom bit sequence output of a linear feedback shift register (LFSR) through a low-pass filter (LPF) [42]. The filtered bit pattern is applied to the switched-capacitor filter through a driving buffer. The  $d_I$  measurements are obtained by digitizing its response at  $d_I$  equidistant points. The complete test configuration is shown in Fig. 7.

The parameters of the test configuration, namely, the clock frequency of the LFSR  $f_{clk}$ , the length of the LFSR  $m$ , and the cutoff frequency of the LPF, can be defined by examining the power spectrum of the unfiltered LFSR output, which is shown in Fig. 8. It can be seen that the envelope of the spectrum is proportional to the square of  $(\sin x)/x$ . The spectrum is flat within  $\pm 0.1$  dB up to 12% of  $f_{clk}$  and drops rapidly beyond its  $-3$ -dB point of  $0.44f_{clk}$ . Thus, low-pass filtering with a high-frequency cutoff of 5%–10% of  $f_{clk}$  will convert the LFSR output to a band-limited white-noise voltage. Since a sharp cutoff characteristic is not required, simple  $RC$  filtering suffices. According to the aforementioned discussion,  $f_{clk}$  must be chosen such that  $0.1f_{clk} \geq \nu \cdot BW$ , where  $BW$  is the

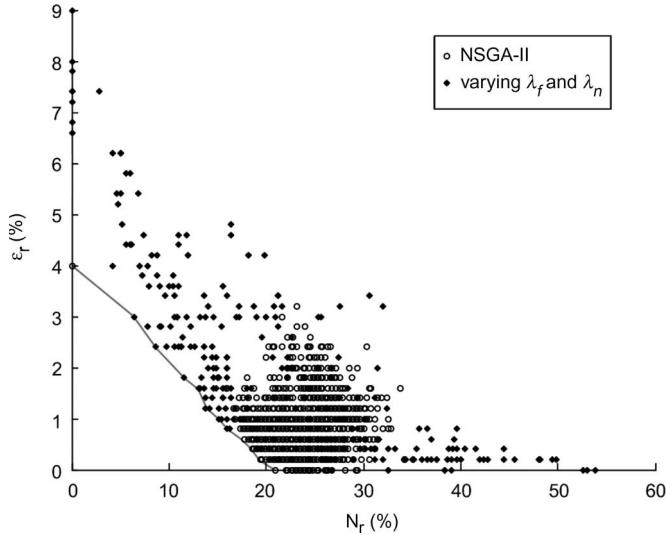


Fig. 9. Tradeoff curve between the test error and the percentage of retested devices for the switched-capacitor filter.

bandwidth of the switched-capacitor filter, and  $\nu$  is a positive integer. The passband of the switched-capacitor filter is in the range of 0–1 kHz; thus, we choose to clock the LFSR at  $f_{\text{clk}} = 100$  kHz. Now, let  $t_r$  be the time resolution between consecutive measurements, and let  $t_o$  be the settling time. In order to avoid repeating measurements,  $m$  must be chosen such that the period of the LFSR  $T_{\text{LFSR}} = (2^m - 1) \cdot T_{\text{clk}}$  (assuming that the LFSR generates a maximal-length pseudorandom sequence) satisfies  $T_{\text{LFSR}} \geq t_o + t_r \cdot d_I$ . We extracted  $d_I = 30$  measurements with a conservative resolution  $t_r = 0.3$  ms. The settling time is approximately  $t_o = 0.5$  ms. Thus, it is sufficient to use an  $m = 10$ -bit LFSR, which will have a period of  $T_{\text{LFSR}} = 10.23$  ms. The characteristic polynomial of the 10-bit maximal-length LFSR that we used is  $z^{10} + z^7 + 1$ .

The results are shown in Fig. 9, which shows a scatter plot of feasible tradeoff points ( $\epsilon_T$  and  $N_r$ ). The elapsed CPU time to generate this scatter plot is about 129 h on a Pentium-IV 2-GHz PC. The circle points correspond to the measurement subspaces that were visited during the course of the NSGA-II. The diamond-filled points are produced by reallocating the guard bands in the identified Pareto-optimal measurement subspaces using different values for the parameters  $\lambda_f$  and  $\lambda_n$ . The Pareto-optimal measurement subspaces have dimensionalities that range between  $d = 10$  and  $d = 13$ . The continuous line runs along the Pareto-optimal front. It can be seen that the test error is 4% when all devices go only through the first tier, it decreases to 2.4% when  $N_r = 8.6\%$ , it drops further to 1% when  $N_r = 15.2\%$ , and it reaches zero when  $N_r = 21\%$ . At this point, for the devices that do not fall within the guard-banded zone, the decision of the first tier is equally accurate to the specification testing. With regard to the average test cost per device, assuming  $C_1 = C_s$  (i.e., the tests in both tiers are performed on the same tester) and a handling time of 0.25 s in the first tier, and given the representative electrical testing times of  $T_1^e = 10$  ms and  $T_s^e = 0.5$  s, an estimate can be calculated from (21) with  $T_h = 0$  as  $C = ((10 + 250) + 0.21 \cdot 500 / (500 + 250))C_s \cdot T_s = 0.49C_s \cdot T_s$ .

## B. UHF Receiver Front End

The data for the UHF receiver front end were provided by our industrial collaborators (i.e., the first three steps in the beginning of this section, where we describe the course of each experiment, were not performed by us). The experiment, which was conducted to generate these data, was originally designed to craft an optimal alternate test stimulus with regard to the ATE constraints, the test time, and the performance-parameter prediction accuracy using MARS as the underlying learning method [19]. Thus, it should be noted that the initial set of  $d_I$  measurements is not specifically extracted for use in conjunction with the neural system described in this paper. In fact, the test error of the first tier without introducing guard bands (e.g., using a single test hypersurface) is 2.36%. In this section, we show that test-error moderation is feasible at the expense of a test-cost increase compared with the low cost reported in [19]. In the following paragraph, for the purpose of completeness, we provide a summary of the experiment. A more detailed description can be found in [19].

The data were obtained on a set of  $N = 541$  devices that were selected among 25 different lots. Every lot contained 25 devices, apart from one which contained only 16. The data sheet consists of 30 specifications at 850/1800 MHz. For the sake of simplicity, only the 850-MHz band was considered, which reduced the set of performance parameters to 13. This includes the gain, the input third-order intercept point, and the noise figure for the LNA, the mixer and their cascade connection, the input standing wave ratio for the LNA and the mixer, and the output standing wave ratio and reverse isolation for the LNA. A total of seven configurations are required to explicitly measure these performance parameters. All 541 devices passed the specification testing successfully. The selected single test configuration, which is shown in Fig. 10 [10], was implemented on a load board and interfaced with a commercial mixed-signal tester. The tester supplied a baseband signal  $x_t(t)$ , which consisted of seven tones around 138 MHz with 1-MHz step, ranging from  $-12$  to  $-19.5$  dBm in amplitude. The center frequency of this baseband signal was upconverted to  $f_1 = 850$  MHz with an external mixer. The modulated signal was then used as a test input stimulus for the LNA. The response of the LNA was downconverted to  $f_1 - f_2 = 50$  MHz with the mixer in the device and passed through the LPF. The output of the LPF is given by

$$x_s(t) = A \cdot x_t(t) \cdot \cos(2\pi(f_1 - f_2) \cdot t + \phi) \quad (22)$$

where  $A$  is the gain of the LNA, and  $\phi$  is the phase difference between the two mixers. The effect of  $\phi$  is removed by taking the fast Fourier transform (FFT) of  $x_s(t)$  and by considering the magnitude of the resulting FFT spectrum as the new signature. The FFT is computed on  $2^{13}$  samples of  $x_s(t)$ , which were obtained at a rate of  $\Delta = 1.3427$  ns.

Fig. 11 shows the FFT transform of  $x_s(t)$  for a randomly selected device. For every harmonic, we calculated the average amplitude across all 541 devices. Then, we set a noise level, and we only considered the tones whose average value is above this level. This resulted in a set of  $d_I = 28$  tones. For the purpose of evaluating our method, we consistently shrunk

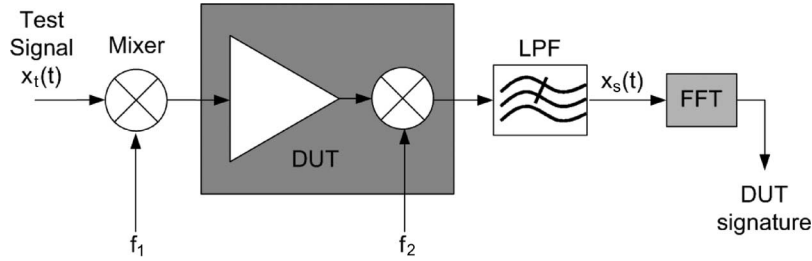


Fig. 10. Test configuration for the UHF receiver front end.

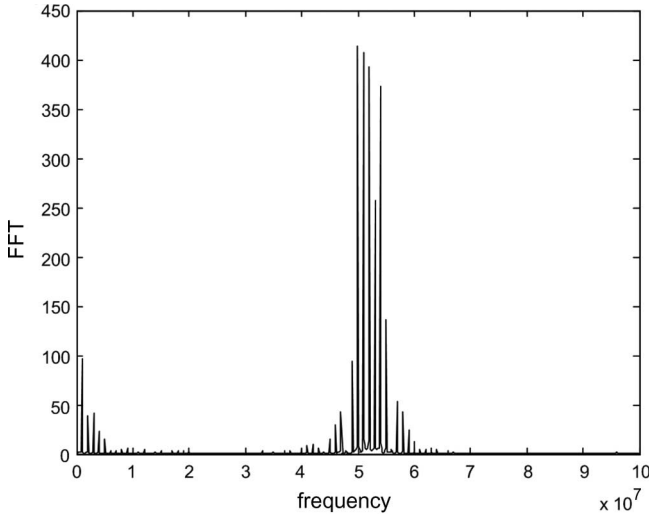


Fig. 11. FFT of the LPF output for a randomly selected device.

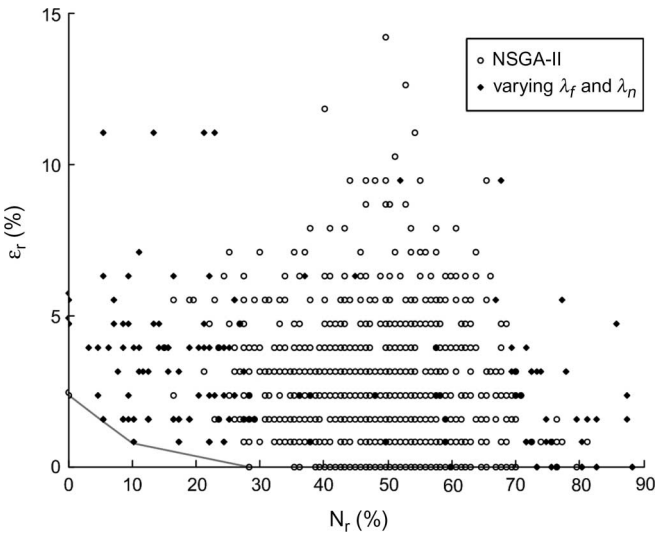


Fig. 12. Tradeoff curve between the test error and the percentage of retested devices for the UHF receiver front end.

the specification limits in order to render some devices faulty. This resulted in 96 devices being labeled as faulty. Out of the 541 devices, we assign 250 to the training set (including 80% of the faulty devices), 164 to the holdout set, and 127 to the test set.

The scatter plot in Fig. 12 shows feasible tradeoff points ( $\epsilon_r$  and  $N_r$ ). The elapsed CPU time to generate this scatter plot is about 19 h on a Pentium-IV 2-GHz PC. Similar to the previ-

ous example, the circle points correspond to the measurement subspaces that were visited during the course of the NSGA-II, whereas the diamond-filled points are produced by reallocating the guard bands in the identified Pareto-optimal measurement subspaces using different values for the parameters  $\lambda_f$  and  $\lambda_n$ . The Pareto-optimal measurement subspaces have dimensionalities that range between  $d = 4$  and  $d = 7$ . The continuous line runs along the Pareto-optimal front. As can be observed, the test error is 2.36% when all devices go only through the first tier, it decreases to 1.57% when  $N_r = 5.51\%$ , it drops further to 0.79% when  $N_r = 10.24\%$ , and finally, it reaches zero when  $N_r = 28.35\%$ . At this point, for the devices that do not fall within the guard-banded zone, the decision of the first tier is equally accurate to the specification testing. The first tier achieves a 36% reduction in test time compared with the specification testing, whereas the mixed-signal tester and the local oscillators to drive the mixers cost approximately 48% less than a commercial RF tester with an adequate functionality to perform the required specification tests [19]. Therefore, without considering the tester depreciation costs, operation costs, etc., it is estimated that  $C_1 \cdot T_1 = 0.52 \cdot 0.64C_s \cdot T_s = 0.3328C_s \cdot T_s$ . An estimate of the average test cost per device can be calculated from (21) as  $C = (0.3328 + 0.2835)C_s \cdot T_s = 0.62C_s \cdot T_s$ .

IX. CONCLUSION

The use of guard bands in machine-learning-based testing of analog/RF devices enables the exploration of the tradeoff between the test accuracy and the test cost. As demonstrated in this paper, efficient allocation of guard bands in carefully selected measurement subspaces allows the majority of devices to be tested through low-cost yet equivalently accurate test criteria to standard specification testing. Additionally, it pinpoints the small fraction of devices that are suspect to misclassification and should be retested through the specification testing in order to ensure the accuracy of the test decision across the entire device population. Results obtained on a switch-capacitor filter and a UHF receiver front end show that the proposed test method maintains the accuracy of specification testing while reducing its cost by 51% and 38%, respectively.

ACKNOWLEDGMENT

The authors would like to thank J. Torres and T. Swettlen of Intel Corporation for providing the UHF receiver front-end data.

## REFERENCES

- [1] D. Gizopoulos, Ed., *Advances in Electronic Testing*, ser. Frontiers in Electronic Testing. New York: Springer-Verlag, 2006.
- [2] C. Y. Pan and K. T. Cheng, "Pseudorandom testing for mixed-signal circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 16, no. 10, pp. 1173–1185, Oct. 1997.
- [3] 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.
- [4] W. M. Lindermeir, H. E. Graeb, and K. J. Antreich, "Analog testing by characteristic observation inference," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 18, no. 9, pp. 1353–1368, Sep. 1999.
- [5] P. N. Variyam and A. Chatterjee, "Specification-driven test generation for analog circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 10, pp. 1189–1201, Oct. 2000.
- [6] V. Stopjakova, P. Malosek, D. Micusik, M. Matej, and M. Margala, "Classification of defective analog integrated circuits using artificial neural networks," *J. Electron. Test.: Theory Appl.*, vol. 20, no. 1, pp. 25–37, Feb. 2004.
- [7] V. Stopjakova, P. Malosek, M. Matej, V. Nagy, and M. Margala, "Defect detection in analog and mixed circuits by neural networks using wavelet analysis," *IEEE Trans. Rel.*, vol. 54, no. 3, pp. 441–448, Sep. 2005.
- [8] H.-G. D. Stratigopoulos and Y. Makris, "Nonlinear decision boundaries for testing analog circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 11, pp. 1760–1773, Nov. 2005.
- [9] P. N. Variyam, S. Cherubal, and A. Chatterjee, "Prediction of analog performance parameters using fast transient testing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 21, no. 3, pp. 349–361, Mar. 2002.
- [10] R. Voorakaranam, S. Cherubal, and A. Chatterjee, "A signature test framework for rapid production testing of RF circuits," in *Proc. Des., Autom. Test Eur.*, 2002, pp. 186–191.
- [11] R. Voorakaranam, R. Newby, S. Cherubal, B. Cometta, T. Kuehl, D. Majernik, and A. Chatterjee, "Production deployment of a fast transient testing methodology for analog circuits: Case study and results," in *Proc. IEEE Int. Test Conf.*, 2003, pp. 1174–1181.
- [12] S. S. Akbay and A. Chatterjee, "Feature extraction based built-in alternate test of RF components using a noise reference," in *Proc. IEEE VLSI Test Symp.*, 2004, pp. 273–278.
- [13] S. Bhattacharya and A. Chatterjee, "Use of embedded sensors for built-in-test of RF circuits," in *Proc. IEEE Int. Test Conf.*, 2004, pp. 801–809.
- [14] A. Raghunathan, J. H. Chun, J. A. Abraham, and A. Chatterjee, "Quasi-oscillation based test for improved prediction of analog performance parameters," in *Proc. IEEE Int. Test Conf.*, 2004, pp. 252–261.
- [15] S. S. Akbay, A. Halder, A. Chatterjee, and D. Keezer, "Low-cost test of embedded RF/analog/mixed-signal circuits in SOPs," *IEEE Trans. Adv. Packag.*, vol. 27, no. 2, pp. 352–363, May 2004.
- [16] S. S. Akbay and A. Chatterjee, "Built-in test of RF components using mapped feature extraction sensors," in *Proc. IEEE VLSI Test Symp.*, 2005, pp. 243–248.
- [17] G. Srinivasan, S. Bhattacharya, S. Cherubal, and A. Chatterjee, "Fast specification test of TDMA power amplifiers using transient current measurements," *Proc. Inst. Electr. Eng.—Comput. Digit. Tech.*, vol. 152, no. 5, pp. 632–642, Sep. 2005.
- [18] G. Srinivasan, A. Chatterjee, and F. Taenzler, "Alternate loop-back diagnostic tests for wafer-level diagnosis of modern wireless transceivers using spectral signatures," in *Proc. IEEE VLSI Test Symp.*, 2006, pp. 222–227.
- [19] S. S. Akbay, J. L. Torres, J. M. Rumer, A. Chatterjee, and J. Amtsfeld, "Alternate test of RF front ends with IP constraints: Frequency domain test generation and validation," in *Proc. IEEE Int. Test Conf.*, 2006, pp. 4.4.1–4.4.10.
- [20] S. Ellouz, P. Gamand, C. Kelma, B. Vandewiele, and B. Allard, "Combining internal probing with artificial neural networks for optimal RFIC testing," in *Proc. IEEE Int. Test Conf.*, 2006, pp. 4.3.1–4.3.9.
- [21] P. Collins, S. Yu, K. R. Eckersall, B. W. Jervis, I. M. Bell, and G. E. Taylor, "Application of Kohonen and supervised forced organization maps to fault diagnosis in CMOS opamps," *Electron. Lett.*, vol. 30, no. 22, pp. 1846–1847, Oct. 1994.
- [22] S. Yu, B. W. Jervis, K. R. Eckersall, I. M. Bell, A. G. Hall, and G. E. Taylor, "Neural network approach to fault diagnosis in CMOS opamps with gate oxide short faults," *Electron. Lett.*, vol. 30, no. 9, pp. 695–696, Apr. 1994.
- [23] S. S. Somayajula, E. Sanchez-Sinencio, and J. P. 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.
- [24] R. Spina and S. Upadhyaya, "Linear circuit fault diagnosis using neuro-morphic analyzers," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 44, no. 3, pp. 188–196, Mar. 1997.
- [25] Y. Maidon, B. W. Jervis, N. Dutton, and S. Lesage, "Diagnosis of multifaults in analogue circuits using multilayer perceptrons," *Proc. Inst. Electr. Eng.—Circuits Devices Syst.*, vol. 144, no. 3, pp. 149–154, Jun. 1997.
- [26] 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 Design Integr. Circuits Syst.*, vol. 19, no. 1, pp. 142–151, Jan. 2000.
- [27] J. H. Friedman, "Multivariate adaptive regression splines," *Ann. Stat.*, vol. 19, no. 1, pp. 1–67, 1991.
- [28] C. M. Bishop, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford Univ. Press, 1995.
- [29] S. Biswas, P. Li, R. D. S. Blanton, and L. Pileggi, "Specification test compaction for analog circuits and MEMS," in *Proc. Des., Autom. Test Eur.*, 2005, pp. 164–169.
- [30] V. Honavar and L. Uhr, "Generative learning structures and processes for generalized connectionist networks," *Inf. Sci.*, vol. 70, no. 1/2, pp. 75–108, 1993.
- [31] R. Parekh, J. Yang, and V. Honavar, "Constructive neural-network learning algorithms for pattern classification," *IEEE Trans. Neural Netw.*, vol. 11, no. 2, pp. 436–451, Mar. 2000.
- [32] M. Frean, "A 'thermal' perceptron learning rule," *Neural Comput.*, vol. 4, no. 6, pp. 946–957, Nov. 1992.
- [33] S. I. Gallant, "Perceptron-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 1, no. 2, pp. 179–191, Jun. 1990.
- [34] N. Burgess, "A constructive algorithm that converges for real-valued input patterns," *Int. J. Neural Syst.*, vol. 5, no. 1, pp. 59–66, 1994.
- [35] D. Brown, J. Ferrario, R. Wolf, J. Li, and J. Bhagat, "RF testing on a mixed-signal tester," in *Proc. IEEE Int. Test Conf.*, 2004, pp. 793–800.
- [36] J. P. de Gyvez, G. Gronthoud, and R. Amine, "VDD ramp testing for RF circuits," in *Proc. IEEE Int. Test Conf.*, 2003, pp. 651–658.
- [37] M. Kudo and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers," *Pattern Recognit.*, vol. 33, no. 1, pp. 25–41, 2000.
- [38] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognit. Lett.*, vol. 10, no. 5, pp. 335–347, Nov. 1989.
- [39] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [40] K. Deb, A. Pratap, A. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [41] R. Gregorian and G. C. Temes, *Analog MOS Integrated Circuits for Signal Processing*. Hoboken, NJ: Wiley, 1986.
- [42] P. Horowitz and W. Hill, *The Art of Electronics*. Cambridge, U.K.: Cambridge Univ. Press, 1989.



**Haralampos-G. Stratigopoulos** (S'02) received the Diploma degree in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 2001, and the M.S. and Ph.D. degrees in electrical engineering from Yale University, New Haven, CT, in 2003 and 2006.

He is currently a Researcher with the Centre National de la Recherche Scientifique (CNRS), Grenoble, France. His research interests are in the areas of mixed-signal/RF design and test, machine learning, and neuromorphic very large scale integration circuits.



**Yiorgos Makris** (S'99–A'01–M'03) received the Diploma degree 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 at San Diego, La Jolla, in 1997 and 2001, respectively.

Since 2001, he has been a member of the faculty of Yale University, New Haven, CT, where he is currently an Associate Professor with the Department of Electrical Engineering and the Department of Computer Science, leading the Testable and Reliable Architectures Research Group. His research interests include test and reliability of analog, digital, and asynchronous circuits and systems.