Generating Decision Regions in Analog Measurement Spaces

Haralampos-G. D. Stratigopoulos
Electrical Engineering Dept.
Yale University
New Haven, CT 06520, USA
haralampos-g.stratigopoulos@yale.edu

Yiorgos Makris
Electrical Engineering Dept.
Yale University
New Haven, CT 06520, USA
yiorgos.makris@yale.edu

ABSTRACT

We develop a neural network that learns to separate the nominal from the faulty instances of a circuit in a measurement space. We demonstrate that the required separation boundaries are, in general, non-linear. Unlike previous solutions which draw hyperplanes, our network is capable of drawing the necessary non-linear hypersurfaces. The hypersurfaces translate to test criteria that are strongly correlated to functional tests. A feature selection algorithm interacts with the network to identify a discriminative low-dimensional measurement space.

Categories and Subject Descriptors

B.7.3 [**Integrated Circuits**]: Reliability and Testing

General Terms

Algorithms, Reliability

Keywords

Analog Circuits, Neural Networks, Implicit Functional Test

1. INTRODUCTION

Functional test of analog circuits is time-consuming since it requires multiple stimuli and test configurations. Early attempts to reduce the cost relied on inductive fault analysis, which sets test thresholds based on observations of fault effects. In the presence of un-modelled faults or strict parametric requirements, such thresholds can severely jeopardize the test decision. Inevitably, they affect either the yield, if they are too strict, or the fault coverage, if they are too lenient. Implicit functional testing [4, 5, 7] is a promising new direction that alleviates the aforementioned limitations. It aims to statistically learn test criteria that strongly correlate to functional tests, yet they are simple to assess. The learning routine is run off-line and only once for any particular device type or production run.

The authors in [7] propose a regression technique to approximate the functions that map the measurement space to the performance parameter space. In this paper, we develop a classification system that establishes a mapping of the form $f : \vec{x} \mapsto Y$, where $\vec{x} \in \mathcal{R}^d$ is a d-dimensional measurement space and Y is a boolean output that indicates a pass or fail decision. In essence, f translates into an adequate number of decision boundaries that separate the nominal from the faulty class in the measurement space. A new instance is tested by examining on which side of the boundaries its measurement pattern lies on. Given a measurement space, the effectiveness of this method depends on the flexibility of the drawn boundaries. Previously reported classifiers [4, 5], establishing the above mapping, allocate hyperplanes. As will be illustrated in detail in section 2, the boundaries are non-linear in practice, with curvatures that cannot be approximated by hyperplanes. Our classifier draws the necessary non-linear hypersurfaces, thus it reciprocates very well even in the presence of complex distributions. As an ancillary benefit of the non-linearity, the proposed classifier requires fewer measurements to solve the separation problem.

2. NON-LINEAR DECISION BOUNDARIES

Fig. 1 illustrates a few real distribution examples in a two-dimensional measurement space $x_1 - x_2$. The solid curves show the linear decision boundaries b_1 drawn by a network similar to the one described in [5], as well as the non-linear decision boundaries b_2 drawn by the higher-order network that we will be describing in the subsequent sections.

Figures 1(a) and 1(b) demonstrate the non-linear nature of the ideal decision boundaries. In addition, 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 non-convex and disjoint. The network is not cognizant of the disjoint nature of the faulty distribution and tries to fit a single boundary. 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. In contrast, our network draws two disjoint boundaries at the two sides of the nominal distribution. As will be illustrated in the next section, it essentially fits a second-order polynomial in the measurement space. Therefore, in the case of the two-dimensional 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

GLSVLSI'05, April 17–19, 2005, Chicago, Illinois, USA.
Copyright 2005 ACM 1-59593-057-4/05/0004 ...$5.00.

to boundaries of any order. While we considered the option of polynomials of higher degree, all the decision boundaries that we came across in our 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 over-fitting the data and does not necessarily *generalize* better.

The flexibility of non-linear boundaries facilitates the discrimination of the two populations in a *lower-dimensional* measurement space. It is true that, by adding more measurements, we can eventually reach the point where the two populations are linearly separable. 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 labelled during the testing phase. This undesirable phenomenon has been termed as *curse of dimensionality* [1].

3. CLASSIFIER STRUCTURE

The proposed system is an artificial neural network that determines the class of a circuit instance by processing its measurement pattern $\vec{x}$. The connectivity of the network is shown in Fig. 2. The network combines the intermediate decisions y_i, $i = 1, ..., M$, corresponding to the M circuit performance parameters, to provide a single pass or fail output decision. The network assigns $y_i = +1$ if the i-th performance parameter lies within the tolerance limits and $y_i = -1$ otherwise. The output neuron computes the logic AND of the intermediate decisions. It indicates $Y = +1$ if $\sum_{i=1}^{M} y_i - M = 0$ and $Y = -1$ otherwise.

The intermediate decisions are made by a hidden layer of perceptrons. A *perceptron* transforms the measurement pattern $\vec{x}$ through a fixed set of $k+1$ processing elements ϕ_j and indicates $y_i = +1$ if $\sum_{j=0}^{k} w_j \phi_j(\vec{x}) > 0$ and $y_i = -1$ otherwise, where $\tilde{w} = [w_0, w_1, ..., w_k]^T$ is the weight vector and ϕ_0 is permanently set to $+1$. The perceptron has a simple geometrical interpretation. It divides the measurement space by allocating a hypersurface composed of the set of solutions to the equation $\sum_{j=0}^{k} w_j \phi_j(\vec{x}) = 0$. In the two sides of the hypersurface, the perceptron activates a pass ($y_i = +1$) and an a fail ($y_i = -1$) decision. The learning process aims to adjust $\vec{w}$ such that the allocated boundary results in the lowest possible decision error rate among the patterns in a *training set.*

The shape of the hypersurface depends on the selection of the fixed processing units. If we set $\phi_j(\vec{x}) = x_j$ and $k = d$, the hypersurface is a conventional hyperplane. For a second-order decision boundary, the fixed processing units ϕ_j are chosen such that $\vec{w}^T\vec{\phi} = \sum_{i_1=0}^{d} \sum_{i_2=0}^{d} w^*_{i_1 i_2} x_{i_1} x_{i_2}$, where $x_0 = +1$. The summations can be constrained to allow for the permutation symmetry of the terms. Based on this remark, each perceptron comprises $\left(\begin{array}{c} d+2 \\ 2 \end{array} \right) = \frac{(d+1)(d+2)}{2}$ fixed processing units.

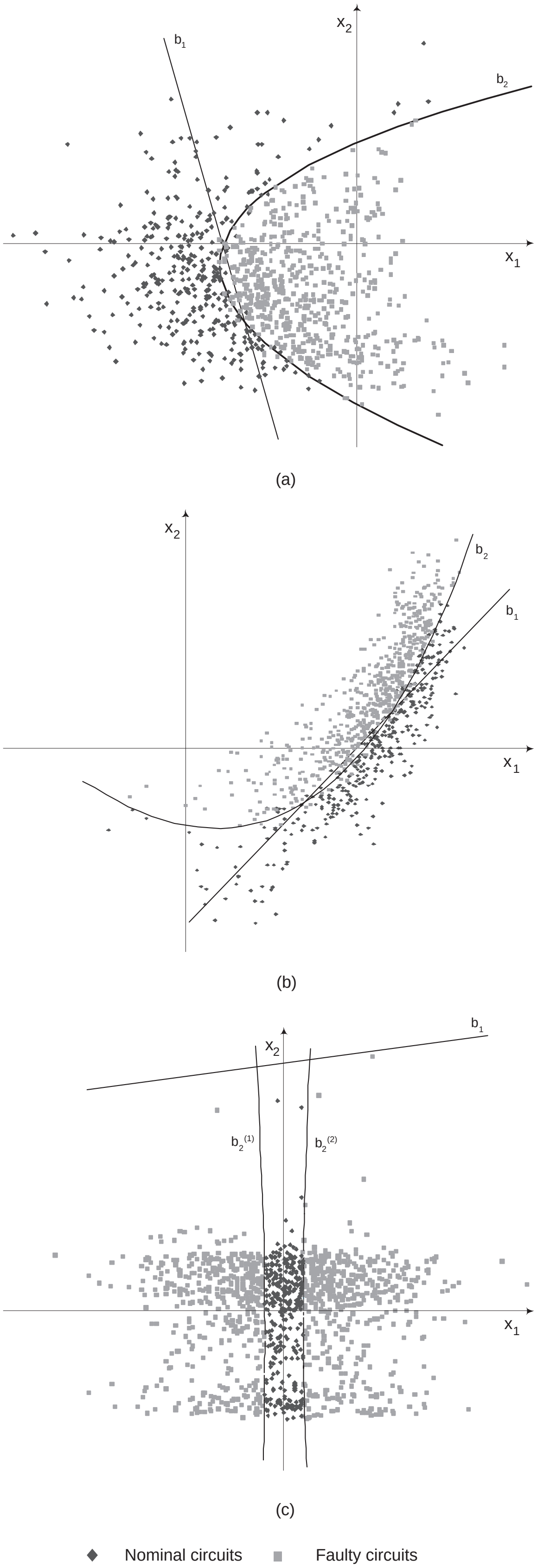

Figure 1: Distributions of nominal and faulty circuits. Instances are labelled with respect to a single-ended specification.

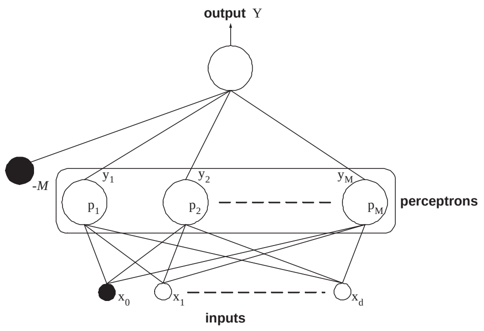

Figure 2: The structure of the proposed neural network. Each p unit corresponds to a perceptron.

4. TRAINING ALGORITHM

The training instances are labelled after they undergo full functional testing. Thus, the training set encodes the statistical impact of process drifts on the circuit performance.

Training is performed for each perceptron separately. We implemented a modification of the classic perceptron learning rule [1], called *pocket algorithm* [2]. The algorithm cycles through the patterns in the training set and modifies the boundary when a visited pattern is misclassified. The new boundary settles closer to the pattern reducing the error and is now likely to classify it correctly. Once a weight vector that has a longer run of consecutive correct classifications is found, it replaces the current weight vector and is kept "in the pocket". The algorithm evolves until a set of iterations is ran with no weight replacement.

As an example, Fig. 3 illustrates the movement of the decision boundary in a two-dimensional space as training progresses. Here, the classifier learns the separation boundary of the two populations shown in Fig. 1(a). In Fig. 3, boundary $b_2^{(i)}$ corresponds to the i-th pocket vector. Table 1 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.

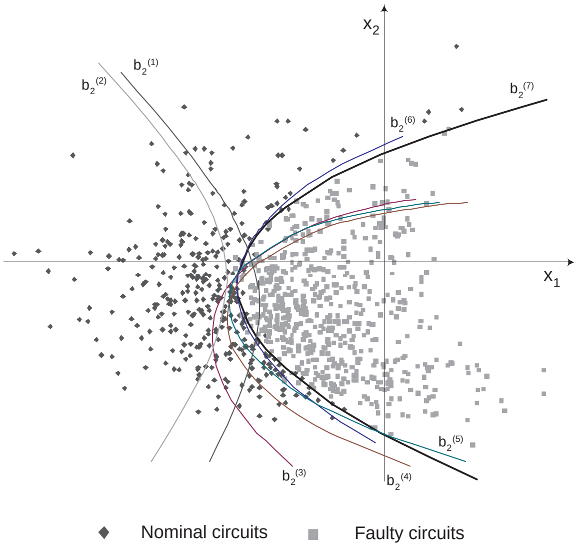

Figure 3: The movement of the decision boundary as training progresses.

	$b_2^{(1)}$	$b_2^{(2)}$	$b_2^{(3)}$	$b_2^{(4)}$	$b_2^{(5)}$	$b_2^{(6)}$	$b_2^{(7)}$
iteration	4	7	84	93	118	137	168
rate	83.4	85.3	86.6	87.3	89.7	94.1	96.4

Table 1: Iterations in which the pocket vector changes and the classification rate that the resulting boundary achieves.

5. SELECTION OF MEASUREMENTS

Increasing the dimensionality of the measurement space does not necessarily improve the ability of the network to generalize and may even cause an adverse effect due to the curse of dimensionality. The problem of selecting the most effective $d^{'}$ measurements from a given set of d measurements, $d^{'} < d$, is called *feature selection*. We implemented a method called *floating search* [6]. The algorithm uses two basic procedures, the *sequential forward selection* (SFS) and the *sequential backward selection* (SBS). Given a subset of measurements, SFS selects from the remaining measurements and includes the one that is the most significant with respect to this subset. Similarly, SBS selects from the current subset of measurements and excludes the one that is the least significant with respect to this subset. Each measurement subset, X_i, is evaluated by training the neural network and computing the classification rate, $J\left(X_i\right)$, achieved on the training set. A subset, X_i, is deemed better than another subset, X_j, if and only if $J\left(X_i\right) > J\left(X_j\right)$. The algorithm is a bottom up search procedure which 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. Upon termination, the algorithm reports the best identified subsets of $d^{'}$ measurements for $d^{'} \in \{1, ..., d-1\}$.

6. EXPERIMENTAL RESULTS

In this section, the advantages of the non-linear decision boundaries are demonstrated on an operational amplifier [3]. Its list of specifications is given in Table 2.

In a production environment, the training set comprises circuit instances across different lots. For the purpose of our experiment, we generate this set through a realistic Monte Carlo analysis. A second independent set is also generated to validate the generalization capacity of the classifier. We consider a diverse initial set of ten measurements that includes Fourier coefficients of the power supply current and samples of the circuit response to *dc*, *ac*, pulse stimuli and to a ramp voltage applied at the positive terminal. The feature selection algorithm identifies the best measurement spaces of cardinalities $d^{'} = 1, ..., 9$. The classifier is trained in each measurement space and its generalization performance is assessed on the independent validation set. For comparison purposes, we also train a linear neural network that has perceptrons with fixed processing units $\phi(\vec{x}) = x_j$, $j = 1, ..., d$.

Performance parameter	Specification limits
Low frequency gain	$A_0 \geq 82db$
Unity-gain frequency	$f_0 \geq 4.8MHz$
Slew-rate	$S_r \geq 1.7V/\mu s$
Common-mode rejection ratio	$CMRR \geq 113db$
Phase margin	$\phi M > 80^o$

Table 2: Specifications of the operational amplifier.

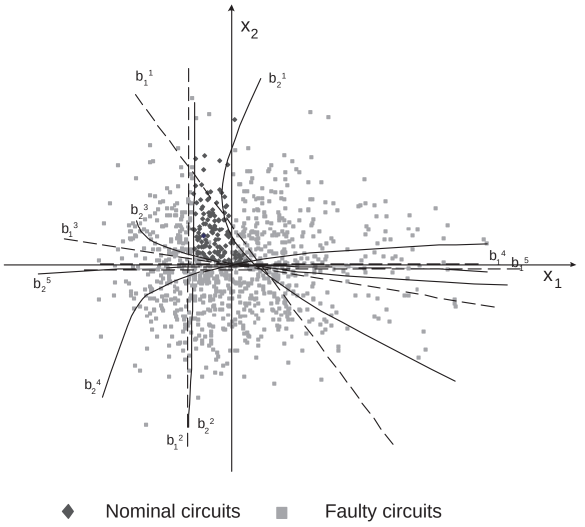

Figure 4: Distributions in the space of the best pair of measurements for the operational amplifier.

Fig. 4 displays the nominal and faulty distributions projected onto the space of the best pair of measurements. Each decision boundary b_j^i, where $i \in \{1, ..., 5\}$, $j = 1$ for the linear network and $j = 2$ for the proposed network, divides the measurement space into two regions $A_n^i(j)$ and $A_f^i(j)$. Circuits that fall into $A_n^i(j)$ are classified as nominal with respect to the i-th performance parameter, while circuits that fall into $A_f^i(j)$ are classified as faulty. As can be observed, the acceptance region defined by the five non-linear boundaries, $\bigcap_{i=1}^5 A_n^i(2)$, approximates the area of nominal circuits better than the acceptance region defined by the five linear boundaries, $\bigcap_{i=1}^5 A_n^i(1)$. This holds for any input cardinality, as can be seen from Fig. 5.

The maximum generalization performance for the proposed network is appreciably larger than that for the linear network. Thus, the resulting test criterion is more accurate. Furthermore, the maximum generalization is obtained in a low dimensional space, which points to a test criterion that is simple to assess.

In both networks, the classification rate for the training set increases monotonically with the number 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. 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.

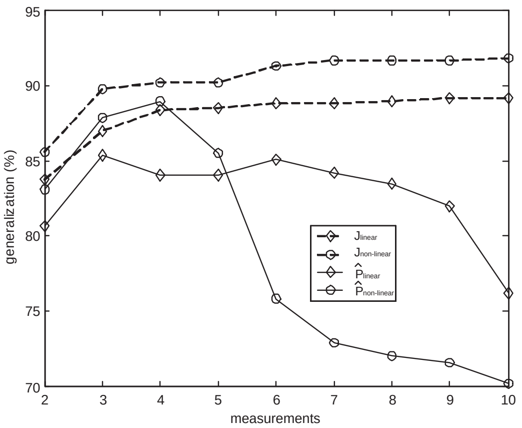

Figure 5: Performance on the training set J and generalization performance $\hat{P}$ of the linear and non-linear boundaries for the operational amplifier.

7. CONCLUSION

We discussed the design of a neural network that tests analog circuits by processing a few measurements. The test criteria are independent of fault models and, in essence, encode the functional tests. A measurement selection algorithm interacts with the neural network to identify useful measurements within a given set. Experimental results show that the proposed system provides a substantially better generalization performance than previously reported linear methods. Furthermore, we pointed out the occurrence of the curse of dimensionality, which suggests that an efficient combination of measurements is desired, rather than a large set.

8. REFERENCES

[1] C. M. Bishop. *Neural Networks for Pattern Pecognition.* Oxford University Press, 1995.

[2] S. I. Gallant. Perceptron-based learning algorithms. *IEEE Transactions on Neural Networks*, 1(2):179–191, 1990.

[3] B. Kaminska, K. Arabi, I. Bell, P. Goteti, J. L. Huertas, B. Kim, A. Rueda, and M. Soma. Analog and mixed-signal benchmark circuits-first release. In *IEEE International Test Conference*, pages 183–190, 1997.

[4] W. M. Lindermeir, H. E. Graeb, and K. J. Antreich. Analog testing by characteristic observation inference. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(9):1353–1368, 1999.

[5] C. Y. Pan and K. T. Cheng. Test generation for linear time-invariant analog circuits. *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, 46(5):554–564, 1999.

[6] P. Pudil, J. Novovicova, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15:1119–1125, 1994.

[7] P. N. Variyam, S. Cherubal, and A. Chatterjee. Prediction of analog performance parameters using fast transient testing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(3):349–361, 2002.