Finding Ambiguity Groups in Low Testability Analog Circuits

Janusz A. Starzyk, *Senior Member, IEEE*, Jing Pang, *Student Member, IEEE*, Stefano Manetti, *Member, IEEE*, Maria Cristina Piccirilli, *Member, IEEE*, and Giulio Fedi, *Member, IEEE*

***Abstract*—This paper discusses a numerically efficient approach to identify complex ambiguity groups for the purpose of analog fault diagnosis in low-testability circuits. The approach presented uses a numerically efficient QR factorization technique applied to the testability matrix. Various ambiguity groups are identified. This helps to find unique solution of fault diagnosis equations or identifies which groups of components can be uniquely determined. This work extends results reported earlier in literature, where QR factorization was used in low-testability circuits, significantly increasing efficiency to determine ambiguity groups. Matlab program that implements this method was integrated with a symbolic analysis program that generates test equations. The method is illustrated on two low-testability electronic circuits. Finally, method efficiency is tested on larger electronic circuits with several hundred tested parameters.**

***Index Terms*—Ambiguity groups, analog system testing, low testability, QR factorization.**

NOMENCLATURE

a	Element of ambiguity group.
$\boldsymbol{a}$	Ambiguity group.
$\boldsymbol{A}$	Set of columns in ambiguity group.
$\boldsymbol{B}$	Testability matrix.
$\boldsymbol{b_1}$	Elements of a basis.
$\boldsymbol{b_2}$	Elements of a cobasis.
$\boldsymbol{C_1}$	Linear combination matrix.
$\boldsymbol{D}$	Binary equivalent matrix.
$\boldsymbol{E}$	Selection matrix.
$\boldsymbol{I}$	Identity matrix.
$\boldsymbol{J}, \boldsymbol{K}$	Sets of indexes.
m	Number of test measurements.
o	Order of ambiguity group.
$O()$	Order of complexity.
p	Number of tested parameters.
$\boldsymbol{P}$	Vector of tested parameters.
$\boldsymbol{Q}$	Orthogonal matrix.
$\boldsymbol{R}$	Upper triangular matrix.
r	Rank of the testability matrix.
$\boldsymbol{t}$	Set of testable components.
$\boldsymbol{V}$	Vector of test measurements.
$\{\}$	Set of elements.
$[]$	Matrix of elements.

Manuscript received October 22, 1998; revised November 30, 1999. This paper was recommended by Associate Editor A. H. M. Van Roermund.

J. A. Starzyk and J. Pang are with the School of Electrical Engineering and Computer Science, Ohio University, Athens, OH 45701 USA.

S. Manetti, M. C. Piccirilli, and G. Fedi are with the Department of Electronic Engineering, University of Florence, Via S. Maria 3-50139 Firenze, Italy.

Publisher Item Identifier S 1057-7122(00)06331-5.

I. INTRODUCTION

FAULT diagnosis and fault location are important parts of analog circuit testing and in the past they have received the attention of many researchers [1]–[8]. These subjects continued to interest researchers (see, for example, [9]–[15]), especially the efforts to automate fault diagnosis procedures [16]. In analog or mixed digital-analog systems, fault diagnosis and fault location are very complex tasks due to the lack of simple fault models and to the presence of component tolerances and circuit nonlinearities. For these reasons the automation level of fault diagnosis and fault location procedures in the analog field has not yet achieved the development level of the digital field, in which well-consolidated techniques for automated test and fault diagnosis are commonly used.

Analog fault diagnosis procedures are usually classified into two categories: the simulation-after-test approach (SAT) and the simulation-before-test approach (SBT). The SAT is currently the prevalent technique but needs more computational time than the SBT technique based on generated off line fault dictionaries. Usually, SBT is suitable for single catastrophic fault location because of the very large dictionary size in multiple soft fault situations.

Independent of the testing approach, testability quantifies the degree of problem solvability and is related to the network element value solvability introduced by Berkowitz [17]. Later, a very useful testability measure was introduced by Saeks *et al.* [18]–[20]. Although other definitions exist [8], [21], the Saeks definition is still very popular as a well-defined quantitative testability measure. Given the circuit topology, selected test points and unknown components, testability can establish *a priori* about the unique solvability of the problem. In the worst case it is necessary to consider further test points or decrease the number of potentially faulty components that can be uniquely diagnosed.

In the case of low-testability, the canonical ambiguity group becomes extremely useful. Roughly speaking, an ambiguity group is a set of components that, if considered as potentially faulty, do not give a unique solution in the fault location. A canonical ambiguity group is simply an ambiguity group containing no other ambiguity groups and is related to the solvability of the fault diagnosis problem with a bounded number of faults (k-fault hypothesis). However, if these important concepts are not taken into account properly, the quality of the obtained test results is severely limited [22].

Testability measure algorithms have been developed first by a numerical approach [23], [24]. Because of the inevitable roundoff errors, they were limited to networks of moderate size.

However, this was solved by the symbolic approach [25]–[27] through an efficient manipulation of algebraic expressions [28], [29]. Furthermore, an efficient algorithm to find ambiguity groups has been proposed by Stenbakken, Souders, and Stewart [30]. Recently, a new method determining all canonical ambiguity groups of a circuit has been developed by the authors [31], [32] by finding all possible ambiguity groups and all sets of circuit parameter values consistent with the test equations. However, the proposed algorithm was combinatorial in nature and was useful only for small analog circuits.

In this paper the ambiguities of the test equations of linear analog systems are dealt with using efficient numerical procedures based on the QR factorization of the testability matrix. The idea of the QR factorization for the testability measure has already been used in [30] and [33] but both methods require combinational searches for ambiguity groups and use QR factorization only to identify dependence of a selected combination of parameters. In the second section of this paper some concepts relevant to the testability matrix and ambiguity groups are recalled [32] and new lemmas and theorems related to the determination of a canonical form of the testability matrix and to its QR factorization are proposed. In the third and fourth sections the identification and complexity reduction problem of ambiguity groups is faced by introducing new theoretical results and explicative examples.

II. Canonical Form of Testability Matrix

Let us assume that test equations were formulated and circuit parameters $\boldsymbol{P}$ are related to test measurements $\boldsymbol{V}$ through the testability matrix $\boldsymbol{B}$ as follows:

$$\boldsymbol{BP} = \boldsymbol{V} \tag{1}$$

where mxp testability matrix $\boldsymbol{B}$ was generated from the test equations and is either related to the Jacobian of the test equations or is equal to a matrix used in linear verification techniques [6]. $\boldsymbol{P}$ and $\boldsymbol{V}$ are either incremental changes of parameters and measurements from the nominal values (related to the Jacobian) or the parameter and the measurement values in the verification approach. Notice that each parameter in the test equations is related to a corresponding column of the testability matrix. We will then identify columns of the testability matrix with their corresponding parameters while discussing ambiguity groups and their solutions.

In a large design many parameters influence the circuit response, which would require huge test equations to solve. This, in turn, would require a large number of the test points and the measurements to formulate the test equations. Such requirements are unacceptable in an industrial testing where both the testing time and the number of test points must be minimized for economical reasons. Fortunately, the number of independent parameter faults in a modern design is limited and parameter changes track each other in the uniform VLSI fabrication process. So, it is reasonable to assume that the number of faulty parameters which have to be identified using the test equations is small.

For a numerical stability and a reduction of the roundoff errors the testability matrix $\boldsymbol{B}$ must have a larger number of rows than columns. So, the number of the measurements m is greater than the number of tested parameters p. The rank of $\boldsymbol{B}$ defines circuit testability T—a maximum number of the identifiable circuit parameters obtained from test equations. In order to properly handle the ambiguity groups, we relate their presence to the solvability and uniqueness of the test equations solution. If $\boldsymbol{B}$ does not have the full column rank, then it can be partitioned into two submatrices $\boldsymbol{B} = [\boldsymbol{B_1}\, \boldsymbol{B_2}]$ which are linearly dependent

$$\boldsymbol{B_2} = \boldsymbol{B_1 C_1} \tag{2}$$

and where mxr matrix $\boldsymbol{B_1}$ has the full column rank equal to the rank of the matrix $\boldsymbol{B}$ and the columns of $r \times (p-r)$ matrix $\boldsymbol{C_1}$, called a linear combination matrix, represent an expansion of the corresponding columns of $\boldsymbol{B_2}$ in the basis vectors obtained from the columns of $\boldsymbol{B_1}$. Using this partition we can write $\boldsymbol{B}$ as

$$\boldsymbol{B} = \boldsymbol{B_1}[\boldsymbol{I C_1}]. \tag{3}$$

Selection of independent columns $\boldsymbol{B_1}$ is not unique and is an important issue in solving the test equations in the presence of ambiguities.

Mathematically, an ambiguity group can be defined as a set of circuit parameters which correspond to linearly dependent columns of the testability matrix $\boldsymbol{B}$. In addition, a canonical ambiguity group is defined as a minimal set of parameters which correspond to linearly dependent columns of $\boldsymbol{B}$. This means that if a single parameter is removed from the canonical ambiguity group, then the remaining set corresponds to independent columns of $\boldsymbol{B}$ and can be uniquely testable. All canonical ambiguity groups have the rank deficiency equal to one, which means that the rank of the corresponding set of columns is equal to the number of parameters in the canonical ambiguity group minus one.

Lemma 1: If $\boldsymbol{A} \subset \boldsymbol{B}$ is a subset of columns of $\boldsymbol{B}$ which corresponds to the canonical ambiguity group $\boldsymbol{a}$, then the following equation is satisfied:

$$\boldsymbol{AC} = \boldsymbol{0} \tag{4}$$

where $\boldsymbol{C}$ is a vector with all nonzero coefficients.

Proof: The requirement for $\boldsymbol{C}$ to have all its coefficients different than zero is easily justified. First, $\boldsymbol{C}$ cannot be equal to zero on the basis of the linear dependence of columns $\boldsymbol{A}$. Suppose that a coefficient $\boldsymbol{c_i} \in \boldsymbol{C}$ is equal to zero. Then, a proper subset of $\boldsymbol{A}$ obtained from $\boldsymbol{A}$ by removing ith column will be dependent, which violates the definition of the canonical ambiguity group. □

The order of a canonical ambiguity group was defined in [32] as equal to the number of components included in the ambiguity group. A combination of canonical ambiguity groups with at least one common element was defined in [32] as global ambiguity group, but in this paper is named the ambiguity cluster for the sake of brevity. Finally, all circuit components which correspond to columns of testability matrix that are not included in any ambiguity group are called surely testable components. The canonical ambiguity group has the following property.

Lemma 2: If the order o of the canonical ambiguity group $\boldsymbol{a}$ is less or equal to the rank r of the testability matrix $\boldsymbol{B}$, then there exists such a partition $\boldsymbol{B} = \boldsymbol{B_1}[\boldsymbol{IC_1}]$ in which a column of $\boldsymbol{C_1}$ that corresponds to an element of this ambiguity group has $r - o + 1$ elements equal to zero.

Lemma 2, which results from the simple transformation of (4) is very useful in deriving a method to break up complex ambiguities (ambiguity clusters) into a number of canonical ambiguities of which it is composed. Finding a proper partition of the testability matrix $\boldsymbol{B}$ is not a trivial task, as it may require combinatorial searches.

Let us consider an ambiguity cluster $\boldsymbol{a}$ composed of two canonical ambiguity groups $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$ which correspond to set of columns $\boldsymbol{A_1} \subset \boldsymbol{B}$ and $\boldsymbol{A_2} \subset \boldsymbol{B}$ with a common element $a_3 = \boldsymbol{a_1} \cap \boldsymbol{a_2}$. The set of columns $\boldsymbol{A}$ that corresponds to $\boldsymbol{a}$ is equal to $\boldsymbol{A_1} \cup \boldsymbol{A_2}$ and the rank deficiency of $\boldsymbol{A}$ is equal to 2. Using Lemma 2 it is easy to demonstrate that there exists such a partition $\boldsymbol{B} = \boldsymbol{B_1}[\boldsymbol{IC_1}]$ in which two columns of $\boldsymbol{C_1}$ that correspond to two elements of this ambiguity cluster different than the common element a_3 have the number of zero elements respectively equal to $r - o_1 + 1$ and $r - o_2 + 1$, where o_1 and o_2 are the corresponding orders of the two ambiguity groups. In addition, these two columns have only one common nonzero element.

Lemma 2, states only the existence of a partition. In order to efficiently find such a partition for any ambiguity group or its combination, we will look for a partition (3) with the matrix $\boldsymbol{C_1}$ in a minimum form, where a matrix $\boldsymbol{C_1}$ is in a minimum form if it has the maximum number of coefficients equal to zero. The corresponding partition (3) is called a canonical form of the testability matrix.

In [31] ambiguity groups were analyzed by checking all possible combinations of tested components with the computational cost on the order of $O(2^p p^3)$. This exponential dependence of the search time on the number of tested parameters renders ambiguity analysis impractical for all but very small designs. Instead, we will discuss a numerically robust solution algorithm based on the QR factorization. Test equations use more measurements than the number of unknown components in order to be able to find a unique solution as well as to compensate for measurement errors and the noise of the measurement equipment [34]. The QR algorithm finds a numerically stable solution of overdetermined system of linear equations that minimizes the least square error. Its numerical complexity is on the order of $O(p^3)$.

As a result of the QR factorization of $m \times p$ testability matrix $\boldsymbol{B}$ we can formulate the following equation:

$$\boldsymbol{BE} = \boldsymbol{QR} \tag{5}$$

where

- $\boldsymbol{E}$ is pxp column selection matrix;
- $\boldsymbol{Q}$ is mxm orthogonal matrix;
- $\boldsymbol{R}$ is mxp upper triangular matrix.

Matrix $\boldsymbol{E}$ has only a single nonzero element equal to one in each column. Matrix product $\boldsymbol{BE}$ represents a permutation of the original columns of the testability matrix $\boldsymbol{B}$. Matrix $\boldsymbol{R}$ has its rank equal to the rank of the testability matrix $\boldsymbol{B}$. Since $\boldsymbol{R}$ is an upper triangular matrix and $p < m$, therefore, all rows of $\boldsymbol{R}$ from $m+1$ to p are zero and, as a result, we need only to generate the first m columns of the orthogonal matrix $\boldsymbol{Q}$. Therefore, in our analysis we will assume that $\boldsymbol{R}$ was reduced to pxp matrix by removing all its zero rows. Furthermore, in the presence of ambiguity groups in the testability matrix $\boldsymbol{B}$, its rank and the rank of $\boldsymbol{R}$ are less than p. Therefore

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R_1} & \boldsymbol{R_2} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{6}$$

where $\boldsymbol{R}_1$ is rxr upper triangular and has its rank equal to the rank of the testability matrix $\boldsymbol{B}$.

The following theorem provides a basis for a numerically efficient approach to finding the ambiguity groups, the ambiguity clusters, and surely testable components.

Theorem 1: A linear combination matrix $\boldsymbol{C_1}$ can be numerically obtained from the QR factorization of the testability matrix $\boldsymbol{B}$ using

$$\boldsymbol{C_1} = \boldsymbol{R_1^{-1}} \boldsymbol{R_2}. \tag{7}$$

Proof: As a result of QR factorization we have $\boldsymbol{BE} = \boldsymbol{QR} = \boldsymbol{Q_r}[\boldsymbol{R_1}\boldsymbol{R_2}]$, where $\boldsymbol{Q_r}$ consists of the first r columns of the orthogonal matrix $\boldsymbol{Q}$. On the other hand, $\boldsymbol{BE} = [\boldsymbol{B_1}\boldsymbol{B_2}]$ is a partition of $\boldsymbol{B}$ that defines $\boldsymbol{C_1}$, with the matrix $\boldsymbol{B_1}$ that corresponds to $\boldsymbol{R_1}$ representing the independent set of columns ($\boldsymbol{R}_1$ has full column rank equal to r). From the definition of $\boldsymbol{C_1}$ we have $\boldsymbol{B_2} = \boldsymbol{B_1}\boldsymbol{C_1}$, so this corresponds to $\boldsymbol{Q_r}\boldsymbol{R_2} = \boldsymbol{Q_r}\boldsymbol{R_1}\boldsymbol{C_1}$. Since $\boldsymbol{Q_r}$ is composed of full columns of the orthogonal matrix we have $\boldsymbol{Q_r^T}\boldsymbol{Q_r} = \boldsymbol{I}$ and multiplying both sides of the previous equation by $\boldsymbol{Q_r^T}$ we get $\boldsymbol{R_2} = \boldsymbol{R_1}\boldsymbol{C_1}$, which in turns yields (7). □

Typically, the QR algorithm, like the one used in Matlab QR routine, performs selection of independent columns by choosing a dominating vector in the orthogonal projection space at each step of the algorithm. We can alter this selection by first normalizing all columns of the testability matrix and then premultiplying columns that we want to be selected first by a constant greater than one (for instance ten). This will not affect column dependencies and will yield a correct result for ambiguity group identification.

III. Identification of Ambiguity Groups

From the above discussion, identification of ambiguity groups becomes easy provided that the linear combination matrix $\boldsymbol{C_1}$ is in its minimum form. The following lemma provides a sufficient condition for the matrix $\boldsymbol{C_1}$ to be in the minimum form.

Lemma 3a: If any two columns of the linear combination matrix $\boldsymbol{C_1}$ have simultaneously nonzero elements in at most one common row, then $\boldsymbol{C_1}$ is in its minimum form.

Different partitions define different linear combination matrices $\boldsymbol{C_1}$. Let us define the basis of a partition as the set of components that correspond to columns of matrix $\boldsymbol{B_1}$ and the cobasis as a set of components that correspond to columns of matrix $\boldsymbol{b_2}$. A minimum form $\boldsymbol{C_1}$ is not unique, as it is enough to switch a component of the basis (that corresponds to a row of $\boldsymbol{C_1}$ with a single nonzero component) with the corresponding component of the cobasis (that corresponds to a column which includes this nonzero component) to obtain another minimum form of $\boldsymbol{C_1}$.

As was discussed in [19], the system testability measure defined as the rank of the testability matrix is independent on parameter values, which means that the rank of the testability matrix is equal to a given testability measure almost everywhere in the parameter space. We will extend this result to ranks of all submatrices of the testability matrix that are used to determine the existence of ambiguity groups. Under this assumption we may study properties of the linear combination matrix $\boldsymbol{C_1}$ considering its equivalent binary matrix $\boldsymbol{D}$ that has the same size as $\boldsymbol{C_1}$. An element of the matrix $\boldsymbol{D}$ is equal to one if the corresponding element of $\boldsymbol{C_1}$ is nonzero, all other elements are set to zero. As in matrix $\boldsymbol{C_1}$, rows of $\boldsymbol{D}$ correspond to the elements of the basis and columns correspond to the elements of the cobasis on a given partition. This equivalent representation simplifies the analysis of $\boldsymbol{C_1}$ as the set theory can be used to study its structural properties.

Using the equivalent binary matrix $\boldsymbol{D}$, Lemma 3a can be written in the equivalent form as follows.

Lemma 3b: If the intersection any two columns of the equivalent binary matrix $\boldsymbol{D}$ have at most one nonzero element, than $\boldsymbol{C_1}$ is in its minimum form.

Our aim in solving the ambiguity problem is to first identify ambiguities and, subsequently, to describe them in the simplest possible way that corresponds to a minimum form of the linear combination matrix $\boldsymbol{C_1}$. Useful results closely related to Lemma 3 describe the existence of surely testable components, canonical ambiguity groups, and ambiguity clusters.

Lemma 4a: A circuit component is surely testable if and only if the corresponding row of $\boldsymbol{C_1}$ is zero.

In order to define the canonical ambiguity groups and the ambiguity clusters, let us identify a set of elements of the cobasis $\boldsymbol{a_2} = \{a_{21}a_{22}\cdots a_{2k}\}$ that corresponds to a union of columns of $\boldsymbol{D}$ such that for each column that corresponds to $a_{2j} \in \boldsymbol{a_2}$ there exists another column that corresponds to $a_{2i} \in \boldsymbol{a_2}, a_{2j} \neq \boldsymbol{a_{2i}}$ such that the two columns have a nonempty intersection. This set of columns can be easily obtained from the matrix $\boldsymbol{D}$ using less than $O((p-r)^3)$ operations. Let the set of the elements of the basis $\boldsymbol{a_1} = \{a_{11}a_{12}\cdots a_{1k}\}$ correspond to nonzero rows in the set of columns described by $\boldsymbol{a_2}$.

Lemma 4b: A set of components described by the union $\boldsymbol{a} = \boldsymbol{a_1} \cup \boldsymbol{a_2}$ constitutes an ambiguity group of the testability matrix $\boldsymbol{B}$.

Ambiguity groups identified by Lemma 4b are either ambiguity clusters or canonical ambiguity groups. Canonical ambiguity groups can be identified by using the following lemma.

Lemma 4c: The ambiguity group represented by the set $\boldsymbol{a} = \boldsymbol{a_1} \cup \boldsymbol{a_2}$ is canonical if and only if cardinality of $\boldsymbol{a_2}$ is equal to one.

Obviously, if the ambiguity group identified in Lemma 4 is not canonical it is an ambiguity cluster. Lemma 4 does not require that the linear combination matrix $\boldsymbol{C_1}$ is in the minimum form. It partitions all the columns of $\boldsymbol{D}$ into row disjoint sets and this partition identifies all the ambiguity groups, ambiguity clusters, as well as all surely testable components. The numerical cost of this partition is very modest compared to the combinatorial search. It cost on the order of $O(p^3)$ operations to obtain the QR factorization (it is enough to run the QR factorization on pxp submatrix of $\boldsymbol{B}$ to determine ambiguities) of the testability matrix $\boldsymbol{B}$ and on the order of $O((p-r)^3)$ to obtain all canonical ambiguity groups and ambiguity clusters.

Example 1: As an example of the ambiguity identification let us consider the following testability matrix as shown in (8) at the bottom of the next page. After running the QR factorization process the following equivalent binary matrix $\boldsymbol{D}$ was obtained:

$$D = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{9}$$

and the following columns were selected for the basis $\boldsymbol{b_1} = \{12\ 13\ 14\ 15\ 11\ 7\ 5\ 10\ 2\ 9\ 4\}$ and the cobasis $\boldsymbol{b_2} = \{3\ 8\ 1\ 6\}$. Using Lemma 4a we can identify the surely testable components as those elements of the basis that correspond to zero rows of $\boldsymbol{D}$, so they are $\boldsymbol{t} = \{11\ 7\ 5\ 10\}$. There are three different sets $\boldsymbol{a_2}$ described in Lemma 4b. The first one corresponds to the first and third column of $\boldsymbol{D}$ so it contains elements three and one, and the other two have a single element only (corresponding to columns two and four). Elements three and one combined with elements of the basis for which the first and the third column have nonzero components $\{12\ 13\ 2\ 4\}$ form an ambiguity cluster $\boldsymbol{a^1} = \{3\ 1\ 12\ 13\ 2\ 4\}$. The remaining two are canonical ambiguity groups (according to Lemma 4c) and are as follows: $\boldsymbol{a^2} = \{8\ 15\ 9\}$ and $\boldsymbol{a^3} = \{6\ 14\}$. □

As mentioned before, Lemma 3 is only a sufficient condition for a minimum form of $\boldsymbol{C_1}$. If two or more columns of $\boldsymbol{D}$ have an intersection with more than one common element we have to analyze these common elements to check for possible simplification that will yield a minimum form. Since at the beginning of the ambiguity analysis we have no idea about the existence and the complexity of the ambiguity groups, the QR algorithm is run and the results are used to reduce the complexity of the ambiguity identification problem.

As a result of a single QR factorization, we can simplify our task by removing all surely testable components and all canonical ambiguity groups from further consideration. Only the ambiguity clusters have to be further analyzed in order to find a minimum form of the linear combination matrix $\boldsymbol{C_1}$. Since all ambiguity clusters identified by Lemma 4 are disjoint sets of the circuit components we can analyze them separately, which significantly reduces our effort to find the minimum form.

IV. Untangling the Complexity of Ambiguity Clusters

As a result of a single QR run we were able to identify all canonical ambiguity groups and all surely testable components in the testability matrix from Example 1. The remaining task is to analyze the ambiguity clusters.

Let us assume that $\boldsymbol{a} = \boldsymbol{a_1} \cup \boldsymbol{a_2}$ is an ambiguity cluster and select $\boldsymbol{A} \subset \boldsymbol{B}$ which corresponds to $\boldsymbol{a}$. We define a minimum form partition of an ambiguity cluster as a minimum form partition of $\boldsymbol{A}$. If the QR factorization is repeated on $\boldsymbol{A}$ and $\boldsymbol{a_1}$ columns are selected for the basis $\boldsymbol{b_1}$ with columns $\boldsymbol{a_2}$ selected as the cobasis $\boldsymbol{b_2}$, then the resulting matrix $\boldsymbol{D_a} \subset \boldsymbol{D}$ is obtained on the intersection of rows that correspond to $\boldsymbol{a_1}$ and columns that correspond to $\boldsymbol{a_2}$. Any reduction in the number of nonzero elements of $\boldsymbol{D_a}$ will result from swapping an element of the basis with an element of the cobasis.

If the matrix $\boldsymbol{A}$ does not include structural zeros (an element of the testability matrix is a structural zero if it is equal to zero for all values of circuit parameters) we could also reduce the number of rows of $\boldsymbol{A}$. This reduced submatrix will have an identical matrix $\boldsymbol{D_a}$ as the original submatrix $\boldsymbol{A}$ almost everywhere in the parameter space, therefore it can be used to identify a minimum form partition of the ambiguity cluster with less computational expense. For numerical stability we could chose the number of rows of $\boldsymbol{A}$ slightly larger than the number of its columns to identify a minimum form partition of the ambiguity cluster, without a significant increase in the computational effort.

Example 2: Using results of the QR factorization in Example 1, we will analyze the ambiguity cluster $\boldsymbol{a^1} = \boldsymbol{a_1} \cup \boldsymbol{a_2} = \{12\ 13\ 2\ 4\} \cup \{3\ 1\} = \{12\ 13\ 2\ 4\ 3\ 1\}$. To do so, let us select a 6×6 submatrix $\boldsymbol{A}$ of the testability matrix $\boldsymbol{B}$ in Example 1 which includes the set of columns $\{3\ 1\ 12\ 13\ 2\ 4\}$.

$$A = \begin{bmatrix} 9 & 3 & -16 & 0 & 1 & 9 \\ 0 & 0 & 40 & 8 & 8 & 2 \\ 3 & 9 & -53 & 48 & 2 & 9 \\ 4 & 4 & 17 & 0 & 9 & 2 \\ 1 & 6 & -17 & 49 & 5 & 9 \\ 6 & 4 & -28 & -14 & 0 & 1 \end{bmatrix}. \tag{10}$$

Its equivalent binary matrix $\boldsymbol{D_a}$ can be obtained from the submatrix $\boldsymbol{D}$ on the intersection of rows $\{1\ 2\ 9\ 11\}$ that correspond to the elements of the basis included in $\boldsymbol{a_1}$ and columns $\{1\ 3\}$ that correspond to the elements of the cobasis included in $\boldsymbol{a_2}$

$$D_a = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}. \tag{11}$$

In this case the only possible simplification may result from swapping elements of the cobasis with elements of the basis that correspond to rows one and three of $\boldsymbol{D_a}$. Therefore we run the QR factorization on $\boldsymbol{A}$ by trying to select as the basis these columns of $\boldsymbol{A}$ which correspond to the original circuit components $\{3\ 13\ 1\ 4\}$. However, in this case, columns which correspond to $\{3\ 13\ 1\ 4\}$ are dependent and cannot be selected to be a basis of the ambiguity group $\boldsymbol{A}$ partition. As the result of the QR factorization, columns that correspond to the circuit components $\{12\ 13\ 1\ 3\}$ were selected as a new basis of a partition of $\boldsymbol{A}$, and the components $\{4\ 2\}$ are in the cobasis. The reduced equivalent binary matrix $\boldsymbol{D_a}$ is as follows:

$$D_a = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}. \tag{12}$$

According to Lemma 3b, the linear combination matrix $\boldsymbol{C_1}$ which corresponds to $\boldsymbol{D_a}$ is in its minimum form, therefore, the obtained partition of columns constitutes the minimum form partition of the ambiguity cluster $\boldsymbol{A}$.

Finally, combining this result with the result of Example 1 we can obtain a minimum form partition of the original testability matrix $\boldsymbol{B}$. This minimum form partition has the basis columns which define matrix $\boldsymbol{B_1}$ in (3) that correspond to the set $\boldsymbol{b_1} = \{12\ 13\ 14\ 15\ 11\ 7\ 5\ 10\ 9\ 1\ 3\}$ and the cobasis equal

$$B = \begin{bmatrix} 3 & 1 & 9 & 9 & 5 & 5 & 6 & 2 & 8 & 4 & 1 & -16 & 0 & -15 & -18 \\ 0 & 8 & 0 & 2 & 4 & 3 & 5 & 8 & 4 & 4 & 6 & 40 & 8 & -9 & 12 \\ 9 & 2 & 3 & 9 & 1 & 3 & 3 & 2 & 5 & 6 & 5 & -53 & 48 & -9 & -9 \\ 4 & 9 & 4 & 2 & 1 & 3 & 5 & 2 & 5 & 6 & 1 & 17 & 0 & -9 & -9 \\ 6 & 5 & 1 & 9 & 7 & 2 & 4 & 6 & 4 & 4 & 8 & -17 & 49 & -6 & 6 \\ 4 & 0 & 6 & 1 & 5 & 6 & 6 & 2 & 7 & 9 & 9 & -28 & -14 & -18 & -15 \\ 5 & 4 & 2 & 3 & 5 & 4 & 7 & 2 & 0 & 2 & 6 & -15 & 17 & -12 & 6 \\ 9 & 9 & 2 & 5 & 3 & 9 & 3 & 3 & 2 & 1 & 7 & -18 & 37 & -27 & 3 \\ 5 & 6 & 1 & 0 & 0 & 4 & 1 & 0 & 1 & 5 & 1 & -5 & 10 & -12 & -3 \\ 5 & 1 & 5 & 4 & 4 & 8 & 8 & 3 & 8 & 4 & 0 & -30 & 6 & -24 & -15 \\ 3 & 8 & 6 & 3 & 7 & 9 & 1 & 0 & 5 & 9 & 4 & 19 & -9 & -27 & -15 \\ 6 & 4 & 7 & 2 & 9 & 8 & 6 & 8 & 8 & 9 & 3 & -22 & -9 & -24 & 0 \\ 5 & 0 & 7 & 5 & 4 & 4 & 9 & 8 & 9 & 4 & 7 & -35 & 0 & -12 & -3 \\ 4 & 1 & 7 & 8 & 8 & 0 & 6 & 1 & 5 & 8 & 8 & -23 & 9 & 0 & -12 \\ 0 & 0 & 2 & 9 & 7 & 2 & 1 & 6 & 6 & 6 & 4 & 0 & 26 & -6 & 0 \\ 8 & 4 & 3 & 9 & 1 & 7 & 6 & 4 & 6 & 5 & 2 & -36 & 45 & -21 & -6 \end{bmatrix} \tag{8}$$

to $\boldsymbol{b_2} = \{8\ 2\ 4\ 6\}$. Using this minimum form partition we obtain the following linear combination matrix $\boldsymbol{C_1}$ of the original testability matrix $\boldsymbol{B}$:

$$C_1 = \begin{bmatrix} 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & -\frac{1}{3} \\ \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1.4 & -0.75 & 0 \\ 0 & 0 & 1.25 & 0 \end{bmatrix}. \tag{13}$$

The linear combination matrix $\boldsymbol{C_1}$ is critical for the optimum solution of test diagnosis equations in the network with ambiguities.

In general, ambiguity clusters can be much more complex than those discussed so far. For instance, if we use the first 11 columns of $\boldsymbol{B}$ depicted in (8) as a basis $\boldsymbol{b_1} = \{4\ 2\ 10\ 11\ 7\ 6\ 8\ 5\ 1\ 3\ 9\}$ and create a new testability matrix by appending the basis columns $\boldsymbol{B_1}$ with new columns $\boldsymbol{B_2} = \boldsymbol{B_1 C_1}$ obtained from the linear combination matrix $\boldsymbol{C_1}$ equal to as shown in (14) at the bottom of this page where the selected cobasis $\boldsymbol{b_2} = \{18\ 19\ 26\ 16\ 21\ 17\ 12\ 13\ 20\ 15\ 22\ 23\ 24\ 25\ 14\ 27\}$ define components which correspond to columns of $\boldsymbol{C_1}$, then, as a result of the QR factorization of this new testability matrix, we may obtain the following nonminimum form of $\boldsymbol{C_1}$ (the following form shows only two digits of the result multiplied by 100 for the simplicity of the presentation) as shown in (15) at the bottom of this page. The QR factorization selected the following columns as a basis $\boldsymbol{b_1} = \{18\ 19\ 26\ 24\ 23\ 12\ 17\ 27\ 16\ 14\ 13\}$ and columns of $\boldsymbol{C_1}$ are arranged according to the selected cobasis $\boldsymbol{b_2} = \{21\ 22\ 15\ 25\ 20\ 7\ 1\ 2\ 9\ 6\ 11\ 5\ 4\ 10\ 3\ 8\}$. Due to the roundoff to two digits, some of zeros in the new $\boldsymbol{C_1}$ are not structural zeros. The binary equivalent matrix computed with the machine precision is as shown in (16) at the bottom of the next page. Our task is to find a minimum form partition of the ambiguity cluster represented by matrix $\boldsymbol{D}$. By comparing obtained $\boldsymbol{D}$ and $\boldsymbol{C_1}$ we see that some zeros in $\boldsymbol{C_1}$ are a result of rounding off to the nearest integer. The true count of zero coefficients, based on the matrix $\boldsymbol{D}$ shows that this form has 51 zeros while the original matrix $\boldsymbol{C_1}$ had 133 zeros. Definitely $\boldsymbol{D}$ is not in a minimum form. However, finding this minimum form requires a systematic approach to untangle the complexity of the ambiguity cluster. This approach is discussed next.

$$C_1 = \begin{bmatrix} 0 & 13 & 0 & 17 & 0 & 13 & 0 & 4 & 3 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 21 & 0 & 0 & 5 & 21 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 13 & -9 & 7 & 0 & 0 & -9 & 11 & 0 & 0 & 0 & 0 & 7 & 9 & 1 & 0 & 0 \\ 0 & 0 & 0 & -13 & 0 & 0 & 0 & 0 & 0 & -3 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3 & 0 & 0 & -5 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 0 & -7 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \\ 21 & 0 & 8 & 0 & 0 & 0 & 12 & -5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -9 & -5 & 0 & 0 & 0 & 0 & -5 & 0 & 0 & 0 & -3 & 0 & 5 & 0 & 0 & 0 \end{bmatrix} \tag{14}$$

$$100C_1 = \begin{bmatrix} -11 & 0 & 46 & -2 & -33 & -14 & -13 & -18 & -47 & 0 & 26 & -6 & 1 & -16 & -7 & -47 \\ -58 & 60 & -9 & 1 & 11 & 0 & 4 & 6 & 5 & 0 & -3 & 2 & 0 & 0 & 2 & 25 \\ 3 & 0 & -16 & 11 & 14 & 0 & -1 & -2 & 9 & 0 & -5 & 0 & 6 & 0 & 4 & 9 \\ 0 & 0 & 33 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 11 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -33 & 17 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -17 & 0 & 0 \\ -7 & 0 & 38 & -6 & -7 & 1 & -18 & -6 & -21 & 0 & 12 & -9 & 2 & 1 & -10 & -21 \\ 60 & -60 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -20 \\ 2 & 0 & -12 & -16 & 11 & 0 & -1 & -1 & 7 & 0 & -4 & -25 & 4 & 0 & 3 & 7 \\ 78 & 0 & -41 & 2 & 48 & 26 & 19 & 26 & 60 & 0 & -33 & 9 & -2 & 30 & 10 & 60 \\ -13 & 0 & 35 & -33 & -63 & 2 & 5 & 8 & -39 & -33 & 22 & 3 & -26 & -15 & -18 & -39 \\ -19 & 0 & 96 & -3 & -23 & 2 & -9 & -13 & -55 & 0 & 30 & -4 & 1 & 3 & -25 & -55 \end{bmatrix} \tag{15}$$

An exhaustive search for possible simplifications of the ambiguity cluster represented by the equivalent binary matrix $\boldsymbol{D}$ would require us to generate and check $\binom{b_1+b_2}{b_2}$ combinations of columns, where b_2 is the size of the cobasis (number of columns in $\boldsymbol{D}$) and b_1 is the size of the basis (number of rows in $\boldsymbol{D}$). Since the motivation for this work was to avoid the combinatorial searches, we need to simplify this task. Finding the minimum form partition corresponds to a proper selection of the basis, therefore, we will concentrate on swapping an element of the basis with an element of the cobasis in the ambiguity cluster. Swapping elements of the basis and the cobasis can be performed independently in different ambiguity clusters, since different clusters have mutually disjoint sets of components.

Let us consider a linear combination matrix $\boldsymbol{C_1}$ in a minimum form. The jth column of the corresponding cobasis $\boldsymbol{B_{2j}} \in \boldsymbol{B_2}$ is related to columns of the basis $\boldsymbol{B_1}$ through the jth column of $\boldsymbol{C_1}$ as follows:

$$B_{2j} = B_1 C_{1j} \tag{17}$$

or using the elements $c_{ij} \in \boldsymbol{C_{1j}}$ of the jth column of $\boldsymbol{C_1}$

$$B_{2j} = \sum_{i=1}^{r} c_{ij} B_{1i} \tag{18}$$

where $\boldsymbol{B_{1i}} \in \boldsymbol{B_1}$ is the ith column of the basis. Let us consider a nonzero coefficient c_{jk} of the kth column of $\boldsymbol{C_1}$ in a minimum form. If we swap the jth element of the basis with kth element of the cobasis, then

$$B_{1j} = \frac{1}{c_{jk}} B_{2k} - \sum_{i \neq j} \frac{c_{ik}}{c_{jk}} B_{1i}. \tag{19}$$

In addition, all other columns of the cobasis will be equal to

$$B_{2m} = \sum_{i \neq j} \left(c_{im} - \frac{c_{jm} c_{ik}}{c_{jk}} \right) B_{1i} + \frac{c_{jm}}{c_{jk}} B_{2k} \tag{20}$$

After swapping, column $\boldsymbol{B_{2k}}$ becomes an element of the basis and kth column of $\boldsymbol{C_1}$ changes to

$$C_{1k} = -\frac{1}{c_{jk}} \begin{bmatrix} c_{1k} \\ c_{2k} \\ \cdot \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ c_{rk} \end{bmatrix} \tag{21}$$

such that all zero locations in the kth column of $\boldsymbol{C_1}$ will remain zero as they were in the original $\boldsymbol{C_1}$. However, as can be deduced from (20), the zero locations c_{im} in columns m that had nonzero element in jth row (c_{jm}) will become nonzero if only the c_{ik} was not zero before transformation. Therefore, the linear combination matrix will no longer have its minimal form.

Lemma 5: A linear combination matrix $\boldsymbol{C_1}$ can be minimized only if it has a 2×2 singular submatrix of nonzero coefficients.

Proof: Let us consider a submatrix of $\boldsymbol{C_1}$ obtained on the intersection of rows i, j and columns k, m that resulted from transformation of a previously zero location c_{im} using (19) and (20). As a result of swapping this submatrix will be expressed by elements of matrix $\boldsymbol{C_1}$ before swapping as follows:

$$\begin{bmatrix} \overline{c_{ik}} & \overline{c_{im}} \\ \overline{c_{jk}} & \overline{c_{jm}} \end{bmatrix} = \begin{bmatrix} -\dfrac{c_{ik}}{c_{jk}} & -\dfrac{c_{jm} c_{ik}}{c_{jk}} \\ \dfrac{1}{c_{jk}} & \dfrac{c_{jm}}{c_{jk}} \end{bmatrix}. \tag{22}$$

It is easy to check that this submatrix is singular. □

Although the existence of a singular 2×2 matrix of all nonzero coefficients is necessary for progress toward a minimum form a number of 2×2 matrices would have to be checked out for possible simplification of the matrix form. In addition, simplification of the matrix form in one location may result in additional nonzero elements in other locations if it is not done properly.

Let us consider a nonzero coefficient c_{jk} of the linear combination matrix $\boldsymbol{C_1}$, and let $\boldsymbol{D_k}$ be a set of rows of the equivalent binary matrix that have 1 in the kth column.

$$D = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{16}$$

Lemma 6: A necessary condition for swapping of the jth element of the basis with the kth element of the cobasis not to increase the number of nonzero coefficients in a linear combination matrix is that all nonzero coefficients in the jth row of $\boldsymbol{D}$ are also nonzero in the corresponding columns of all rows of $\boldsymbol{D_k}$.

An element c_{jk} of the linear combination matrix $\boldsymbol{C_1}$ that satisfies condition of Lemma 6 is called a candidate for reduction toward a minimum form. In order to swap a number of basis and cobasis elements we define a prime candidate for reduction toward a minimum form as the largest minor of $\boldsymbol{C_1}$ whose elements are candidates for reduction toward a minimum form. Let $\boldsymbol{J}$ and $\boldsymbol{K}$ represent the set of indexes of a prime candidate rows and columns, respectively.

Lemma 7: If a submatrix of $\boldsymbol{C_1}$ obtained on intersection of rows $\boldsymbol{J}$ and columns $\boldsymbol{K}$ is a prime candidate then all rows $\boldsymbol{J}$ of the equivalent binary matrix are equal.

Let $\boldsymbol{JK}$ represent a set of rows of $\boldsymbol{D}$ that have a nonzero element in one of the columns $\boldsymbol{K}$.

Lemma 8: If a submatrix of $\boldsymbol{C_1}$ obtained on intersection of rows $\boldsymbol{J}$ and columns $\boldsymbol{K}$ is a prime candidate then all nonzero coefficients in the rows $\boldsymbol{J}$ of $\boldsymbol{D}$ are also nonzero in the corresponding columns of all rows $\boldsymbol{JK}$.

Lemma 8 extends the results of Lemma 6 to sets of rows and columns and improves efficiency of an algorithm that identifies prime candidates for reduction. The following lemma reduces identification of a prime candidate to identification of its diagonal:

Lemma 9: If two elements c_{jk} and c_{lm}, where $j \neq l$ and $k \neq m$ are the candidates for reduction and the corresponding rows of $\boldsymbol{D}$ are equal, then all elements of the submatrix obtained on the intersection of rows j and l and columns k and m are candidates for reduction.

Using Lemma 9, the largest set of candidates on the intersection of different rows and columns of $\boldsymbol{D}$ with all rows equal identify the prime candidate. The first step toward untangling a complex ambiguity cluster is to find all prime candidates for reduction toward a minimum form based on Lemmas 7–9. Then these prime candidates are used in swapping elements of the basis and the cobasis and the results of swapping are evaluated. The following procedure describes this approach.

Procedure 1—Finding a Minimum Form Partition of an Ambiguity Cluster:

1) Define the row equivalence classes as the subsets of identical rows of $\boldsymbol{D}$.
2) Starting from the largest row equivalence class, determine a largest subset of rows $\boldsymbol{J}$ in this equivalence class and the corresponding subset of columns $\boldsymbol{K}$ for which Lemma 9 is satisfied. If there is no nonzero coefficient c_{jk} for which Lemma 6 is satisfied—stop.
3) Repeat Point 2) for each row equivalence class that is larger than the size of the largest subset found in 2.
4) Use the largest subset found in Point 2) as a prime candidate and swap the basis elements that correspond to its rows with the co-basis elements that correspond to its columns.
5) If as a result of swapping, the number of zero coefficients in $\boldsymbol{C_1}$ did not increase, then repeat Points 2)–4) for the next prime candidate.
6) If there was no increase in the number of zero elements in $\boldsymbol{C_1}$ after executing Points 2)–5), then reduce the size of prime candidates by one and repeat Points 2)–5).
7) If the number of zero elements in $\boldsymbol{C_1}$ increased after executing Points 2)–6), then go back to Point 1, otherwise stop.

In order to illustrate the procedure to find a minimum form partition of an ambiguity cluster let us consider the following example.

Example 3: Let us apply Procedue 1 to the matrix $\boldsymbol{D}$ described in (16).

1) We have the following row equivalence classes: $\{1\ 3\ 6\ 8\ 9\ 11\}$, $\{2\}$, $\{4\}$, $\{5\}$, $\{7\}$, $\{10\}$.
2) To illustrate Lemma 6, let us consider the element c_{11}, for which $D_1 = \{1, 2, 3, 6, 7, 8, 9, 10, 11\}$. Now we check all the elements of the first row of D that are nonzero and verify if all the elements of the corresponding column of D_1 are nonzero. Since the coefficients $d_{13} = 1$ and $d_{73} = 0$, the necessary condition for swapping is not satisfied and, based on Lemma 6, the first element of the basis and the cobasis cannot be swapped. However, Lemma 6 is satisfied for c_{15}, so this element is a candidate for reduction. To find a prime candidate we first analyze other rows in the largest equivalence class looking for candidates for reduction. We see that Lemma 6 is also satisfied for the following elements: c_{36}, c_{67}, c_{88}, c_{99}, $c_{11\ 12}$, therefore, based on Lemma 9, a prime candidate for reduction is a submatrix on the intersection of rows $\boldsymbol{J} = \{1\ 3\ 6\ 8\ 9\ 11\}$ and columns $\boldsymbol{K} = \{5\ 6\ 7\ 8\ 9\ 12\}$.
3) Since no row equivalence class is larger than the size of $\boldsymbol{J}$ no other subsets of rows are considered.
4) We swap the basis elements $\boldsymbol{J}$ with the cobasis elements $\boldsymbol{K}$. As a result the elements $\{18\ 26\ 12\ 27\ 16\ 13\}$ of the old basis $\boldsymbol{b_1} = \{18\ 19\ 26\ 24\ 23\ 12\ 17\ 27\ 16\ 14\ 13\}$, will be replaced by the elements $\{20\ 7\ 1\ 2\ 9\ 5\}$, such that the proposed order of element selection for the new basis is $[19\ 24\ 23\ 17\ 14\ 20\ 7\ 1\ 2\ 9\ 5\}]$. Notice that not all of the proposed elements may be selected as a result of QR factorization, since they may be mutually dependent. This dependence will be automatically detected by the QR algorithm and dependent columns will be moved to the cobasis. When the QR is run again with the proposed basis it returns a new equivalent binary matrix $\boldsymbol{D}$ as shown in (23) at the bottom of the next page.

The selected basis is $\{17\ 23\ 5\ 1\ 9\ 14\ 2\ 24\ 7\ 20\ 3\}$ and the cobasis is $\{22\ 19\ 18\ 12\ 27\ 4\ 6\ 13\ 10\ 21\ 15\ 11\ 8\ 25\ 26\ 16\}$. The number of zeros in $\boldsymbol{D}$ increased significantly to 121 after this step and the matrix is now closer to its minimum form.

5.6.7. Since as a result of swapping, the number of zero coefficients in $\boldsymbol{C_1}$ increased, then we proceed to Step 1).

Steps 1)–7) are repeated for the new matrix $\boldsymbol{D}$. Since there is no identical rows, each row is its own equivalence class. The largest row equivalence class have size one, so it is enough to find a single coefficient c_{jk} for which Lemma 6 is satisfied. We can check that Lemma 6 is satisfied for the coefficient $c_{1\ 13}$ as row number one of $\boldsymbol{D}$ is included in all other rows that contain 13th element of the cobasis (rows 7, 9, and 10) therefore, the first element of the basis and the 13th element of the cobasis will be swapped.

In a similar fashion we will swap the ninth element of the cobasis with the second element of the basis as well as 12th element of the cobasis with the eighth element of the basis and, finally, the sixth element of the cobasis with the tenth element of the basis. As a result, new basis components tried for QR factorization are $[8\ 10\ 5\ 1\ 9\ 14\ 2\ 11\ 7\ 4\ 3]$. When the QR algorithm is run again with the selected elements of the proposed basis it returns a new equivalent binary matrix $\boldsymbol{D}$ is in its minimum form as shown in (24) at the bottom of this page. The selected basis elements are $\{1\ 8\ 2\ 3\ 11\ 4\ 10\ 14\ 5\ 7\ 9\}$ and the cobasis is $\{23\ 26\ 19\ 27\ 22\ 17\ 18\ 6\ 20\ 21\ 16\ 12\ 24\ 25\ 13\ 15\}$. Since there is no nonzero coefficient c_{jk} for which Lemma 6 is satisfied, the procedure stops.

Based on the presented discussion the following procedure identifies all surely testable components, ambiguity clusters, and canonical ambiguity groups, as well as finds a canonical form of the testability matrix.

Procedure 2—Canonical Form of the Testability Matrix:

1) Formulate test equations (1) and identify the testability matrix $\boldsymbol{B}$.
2) Run the QR factorization on $\boldsymbol{B}$ to obtain (5).
3) Use the column selection matrix $\boldsymbol{E}$ to find initial elements of the basis and the cobasis.
4) Represent $\boldsymbol{R}$ in the form (6).
5) Find the linear combination matrix $\boldsymbol{C_1}$ using (7).
6) Find the equivalent binary matrix $\boldsymbol{D}$ for $\boldsymbol{C_1}$.
7) Use Lemma 4 to identify all surely testable components, canonical ambiguity groups, and ambiguity clusters.
8) Find a minimum form partition of each ambiguity cluster as described in Procedure 1).
9) Combine all basis components from all ambiguity clusters with all surely testable components to form the final basis and the cobasis.
10) Use the final basis to obtain the canonical form of the testability matrix $\boldsymbol{B}$.

V. Analog Circuit Examples

Procedures 1) and 2) presented in previous section were programmed in Matlab and combined with symbolic analysis program SAPWIN [35] and SYFAD [16] to obtain and analyze testability equations of analog circuits.

Example 4: To illustrate results of these programs let us consider the following BJT transistor circuit shown in Fig. 1.

The BJT model used for the symbolic analysis is the simplified one that considers only the input conductance Gi and the current gain h_{fe}. If we select Test 1 in Fig. 1 as test point we obtain the following test point equation, written in a symbolic form, obtained with the software tool SAPWIN, as shown in (25) at the bottom of the next page. We consider as the second test point Test 2 in Fig. 2.

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{23}$$

$$D = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{24}$$

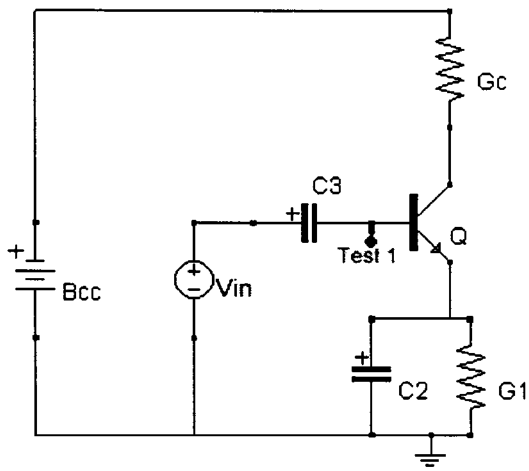


Fig. 1. An example of a low-testability circuit

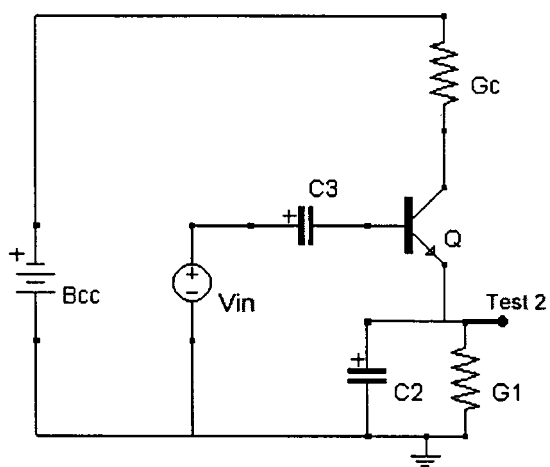


Fig. 2. Second test point selection

The symbolic test equation as it has been given by the software SAPWIN is as shown in (26) at the bottom of the page. Then the testability matrix $\boldsymbol{A}$, written in a symbolic form was obtained by the software SYFAD, as shown at the bottom of the next page. If we assign the following integer values to the parameters: $G1 = 1$; $Gc = 2$; $C2 = 4$; $C3 = 3$; $h_{fe} = 2$; $Gi = 1$ then we obtain the following numerical testability matrix:

$$a = \begin{bmatrix} \frac{1}{4} & 0 & -\frac{1}{4} & 0 & \frac{1}{4} & \frac{3}{4} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{3} & -\frac{3}{16} & -\frac{1}{4} & \frac{1}{6} & \frac{7}{12} \\ \frac{1}{4} & \frac{1}{3} & -\frac{1}{4} & -\frac{1}{3} & \frac{1}{4} & \frac{13}{12} \\ 0 & 0 & -\frac{3}{16} & 0 & \frac{1}{4} & \frac{3}{4} \\ \frac{1}{4} & \frac{1}{3} & -\frac{3}{16} & -\frac{1}{4} & \frac{1}{6} & \frac{7}{12} \\ \frac{1}{4} & \frac{1}{3} & -\frac{1}{4} & -\frac{1}{3} & \frac{1}{4} & \frac{13}{12} \end{bmatrix}.$$

The following results were obtained by the Matlab program which implements Procedures 1) and 2):

$$C_1 = \begin{bmatrix} 0.000 & -0.654 \\ -2.966 & 0.814 \\ 0.000 & -0.814 \\ -3.873 & 0.000 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

The order of columns selected by QR factorization is (2, 5, 4, 3, 1, and 6). The basis elements selected are $\{2\ 5\ 4\ 3\}$ and the cobasis is $\{1\ 6\}$. Hence, the following canonical ambiguity groups have been determined $a_1 = \{5\ 3\ 1\}$, and $a_2 = \{2\ 5\ 4\ 6\}$ corresponding to the following circuit parameters: $\{hfe, C2, G1\}$ and $\{Gc, hfe, C3, Gi\}$. These two groups form the ambiguity cluster with equivalent binary matrix D.

As expected, this result is in full agreement with the result of symbolic analysis, however, it is obtained at a fraction of the time needed for symbolic analysis. Time savings are important, particularly when the size of the test equation is large as the combinatorial search is replaced by a program of polynomial complexity.

Example 5: A somehow larger example, the attenuator circuit shown in Fig. 3 [30], has system test equation $\boldsymbol{BP} = \boldsymbol{V}$ with matrix $\boldsymbol{B}$ obtained by sensitivity analysis. Sensitivities with respect to 19 circuit parameters were calculated at 41 frequency points spread evenly from 10 Hz to 1 MHz. The only test point selected was the circuit output voltage. The complete sensitivity matrix in this example contains 19 columns and 41 rows and was analyzed using the Matlab based program called ambiguous test equations solver (ATES). To consider week dependencies between parameters we selected rank of the testability matrix using only the singular values that were greater than 10^{-8}. In addition, elements of the linear

$$\text{Test } 1(s) = \frac{(C3Gihfe + C3G1 + C3Gi)s + (C3C2)s^2}{(GiG1 + Gih_{fe}Gc + G1Gc + GiGc) + (C2Gc + GiC2 + C3Gi + C3Gi)s + (C3C2)s^2} \tag{25}$$

$$\text{Test } 2(s) = \frac{(C3Gihfe + C3G1)s}{(GiG1 + GihfeGc + G1Gc + GiGc) + (C2Gc + GiC2 + C3Gi + C3Gi)s + (C3C2)s^2} \tag{26}$$

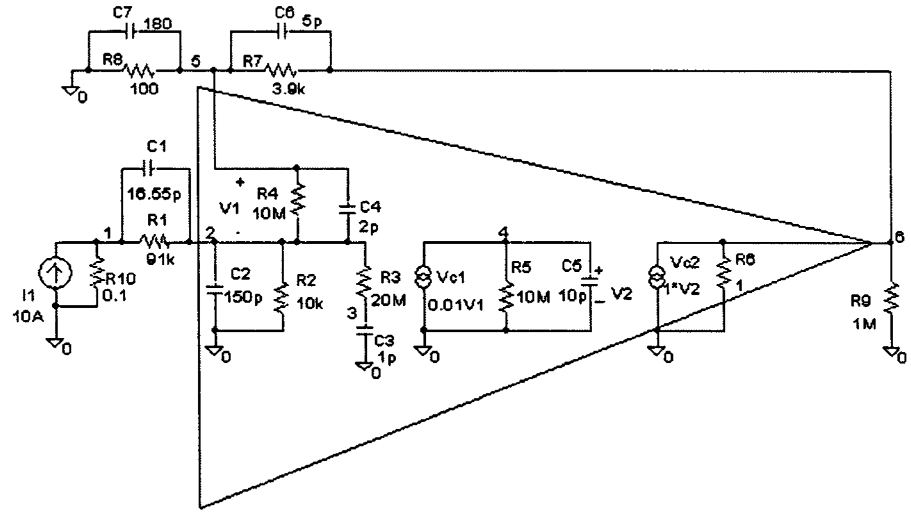


Fig. 3. Attenuator circuit example.

combination matrix that were less than 10^{-2} were set to zero. With this two threshold values, we obtained results similar to those reported in [30]. Basis elements selected were $\{R10, R1, VC2, R7, R2, C3, R3, C7\}$ and the cobasis were $\{C2, R8, C1, C6, VC1, C5, R6, C4, R5, R4, R9\}$. Parameter $R9$ was ambiguous by itself due to its low sensitivity values (zero column in the linear combination matrix). Parameters $R10$, $R1$, $R7$, $C3$, $R2$, $R3$ were surely testable. All ambiguity groups in this circuit were determined in 0.19 s.

Due to the complex dependencies between columns of testability matrix this circuit is difficult to test as observed in [30] and the observed dependencies between parameters vary significantly with changes in the selected threshold levels. This numerical study of ambiguity group dependencies on the measurement error and machine precision is an interesting topic, however, it is beyond the scope of this paper.

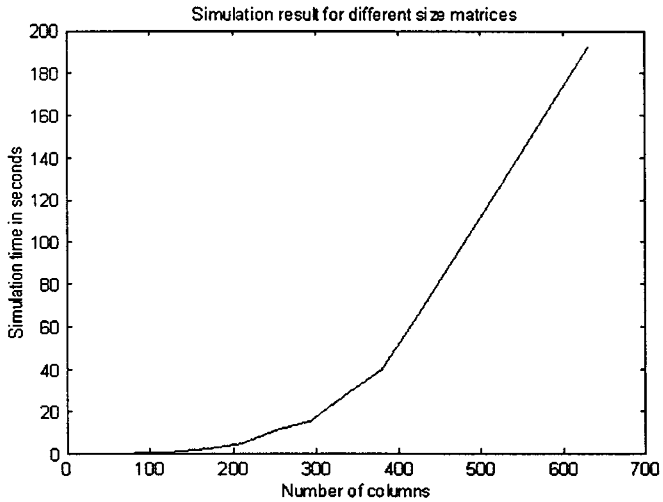
To demonstrate the numerical efficiency of the method, a number of test matrices of various dimensions were analyzed and ambiguity groups determined using the described methodology. All of these matrices were too big to complete calculations based on symbolic analysis, so no direct comparison is possible. The computer simulation time required to find all ambiguity groups in the analyzed matrices are displayed on Fig. 4. The simulation time is shown as a function of the number of matrix columns.

Simulation was performed using program ATES on 300-MHz Pentium PC computer with 128-MB RAM and Windows 95. In all cases the number of rows were greater than the number of columns. As can be seen from the Fig. 4 the simulation time grows approximately as a cube of the number of circuit parameters (number of columns of the sensitivity matrices). Based on the observed results we can perform ambiguity group analysis for medium size analog circuit with several hundred discrete parameters.

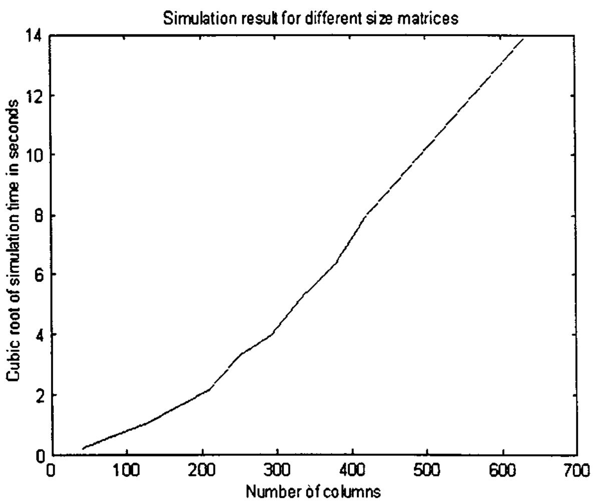
VI. Conclusions

An efficient numerical approach for testing linear analog systems with ambiguities has been presented. The paper describes techniques to identify various ambiguity groups and to present them in a simplest possible way in order to diagnose low-testability systems. All canonical ambiguity groups, ambiguity clusters and surely testable components can be easily identified by using results of the QR factorization of the circuit testability matrix. Computational complexity of the algorithm that identifies ambiguity groups is on the order of $O(p^3)$. The method can be

$$A = \begin{bmatrix} \frac{1}{C2} & 0 & -\frac{(Gihfe+G1+Gi)}{C2^2} & 0 & \frac{Gi}{C2} & \frac{(hfe+1)}{C2} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{(Gi+Gc)}{(C3\cdot C2)} & \frac{(Gihfe+G1+Gi)}{(C3\cdot C2)} & -\frac{(GiG1+GihfeGc+G1Gc+GiGc)}{(C3\cdot C2^2)} & -\frac{(GiG1+GihfeGc+G1Gc+GiGc)}{(C3^2\cdot C2)} & Gi\,\frac{Gc}{(C3\cdot C2)} & \frac{(G1+hfeGc+Gc)}{(C3\cdot C2)} \\ \frac{1}{C2} & \frac{1}{C3} & -\frac{(Gihfe+G1+Gi)}{C2^2} & -\frac{(Gi+Gc)}{C3^2} & \frac{Gi}{C2} & \frac{(C2+C3+C3hfe)}{(C3\cdot C2)} \\ 0 & 0 & -Gi\,\frac{(hfe+1)}{C2^2} & 0 & \frac{Gi}{C2} & \frac{(hfe+1)}{C2} \\ \frac{(Gi+Gc)}{(C3\cdot C2)} & \frac{(Gihfe+G1+Gi)}{(C3\cdot C2)} & -\frac{(GiG1+GihfeGc+G1Gc+GiGc)}{(C3\cdot C2^2)} & -\frac{(GiG1+GihfeGc+G1Gc+GiGc)}{(C3^2\cdot C2)} & Gi\,\frac{Gc}{(C3\cdot C2)} & \frac{(G1+hfeGc+Gc)}{(C3\cdot C2)} \\ \frac{1}{C2} & \frac{1}{C3} & -\frac{(Gihfe+G1+Gi)}{C2^2} & -\frac{(Gi+Gc)}{C3^2} & \frac{Gi}{C2} & \frac{(C2+C3+C3hfe)}{(C3\cdot C2)} \end{bmatrix}$$



(a)



(b)

Fig. 4. Simulation time as a function of number of parameters. (a) Time in seconds. (b) Cubic root of the simulation time

used to find a unique solution of the test diagnosis equations if such unique solution exists or determine which components are uniquely determined and which have ambiguous solutions. Such information is of fundamental importance in the fault diagnosis procedures or in the circuit parameters extraction, because it gives rigorous upper limit to the theoretical solvability of the problem. The computational cost of the method presented here are very little comparing to the combinatorial one proposed in the previous works. This allows to deal with analog circuits having several hundred parameters in an efficient way. A Matlab–based algorithm was developed to implement the described procedure and was integrated with the symbolic analysis program SYFAD that determines the testability matrix and program SAPWIN that determines testability equations. This has allowed to obtain a fully automated method that, starting from the topology of the circuit under test, determines the ambiguity groups and the surely testable circuit parameters.

References

[1] W. Hochwald and J. D. Bastian, "A dc approach for analog fault dictionary determination," *IEEE Trans. Circuits Syst.*, vol. CAS-26, pp. 523–529, July 1979.

[2] K. C. Varghese, J. H. Williams, and D. R. Towill, "Simplified ATPG and analog fault location via a clustering and separability technique," *IEEE Trans. Circuits Syst.*, vol. CAS-26, pp. 496–505, July 1979.

[3] N. Navid and A. N. Willson, "A theory and an algorithm for analog circuit fault diagnosis," *IEEE Trans. Circuits Syst.*, vol. CAS-26, pp. 440–457, July 1979.

[4] R. M. Biernacki and J. W. Bandler, "Multiple-fault location of analog circuits," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 361–367, May 1981.

[5] C. C. Wu, K. Nakajima, C. L. Wey, and R. Saeks, "Analog fault diagnosis with failure bounds," *IEEE Trans. Circuits Syst.*, vol. CAS-29, pp. 277–284, May 1982.

[6] J. A. Starzyk and J. W. Bandler, "Multiport approach to multiple-fault location in analog circuits," *IEEE Trans. Circuits Syst.*, vol. CAS-30, pp. 762–765, Oct. 1983.

[7] A. E. Salama, J. A. Starzyk, and J. W. Bandler, "A unified decomposition approach for fault location in large analog circuits," *IEEE Trans. Circuits Syst.*, vol. CAS-31, pp. 609–622, July 1984.

[8] J. W. Bandler and A. E. Salama, "Fault diagnosis of analog circuits," *Proc. IEEE*, vol. 73, pp. 1279–1325, Aug. 1985.

[9] R. Liu, *Testing and Diagnosis of Analog Circuits and Systems*. New York: Van Nostrand Reinhold, 1991.

[10] J. L. Huertas, "Test and design for testability of analog and mixed-signal integrated circuits: Theoretical basis and pragmatical approaches," in *Proc. Circuit Theory Design, ECCTD'93*, H. Dedieu, Ed., Amsterdam, The Netherlands, 1993, pp. 77–156.

[11] H. Zhihua Wang, G. Gielen, and W. Sansen, "A novel method for the fault location of analog integrated circuits," in *Proc. IEEE Int. Symp. Circuits Systems, ISCAS'94*, London, U.K., May 1994, pp. 347–350.

[12] M. Slamani and B. Kaminska, "Multifrequency analysis of faults in analog circuits," *IEEE Design Test Comput.*, vol. 12, pp. 70–80, Summer 1995.

[13] ——, "Fault observability analysis of analog circuits in frequency domain," *IEEE Trans. Circuits Syst.*, vol. 43, pp. 134–139, Feb. 1996.

[14] H. T. Sheu and Y. H. Chang, "Robust fault diagnosis for large-scale analog circuits with measurement noises," *IEEE Trans. Circuits Syst. I*, vol. 44, pp. 198–209, Mar. 1997.

[15] R. Spina and S. Upadhyaya, "Linear circuit fault diagnosis using neuromorphic analyzers," *IEEE Trans. Circuits Syst. II*, vol. 44, pp. 188–196, Mar. 1997.

[16] G. Fedi, R. Giomi, A. Luchetta, S. Manetti, and M. C. Piccirilli, "On the application of symbolic techniques to the multiple fault location in low testability analog circuits," *IEEE Trans. Circuits Syst. II*, vol. 45, pp. 1383–1388, Oct. 1988.

[17] R. S. Berkowitz, "Conditions for network-element-value solvability," *IEEE Trans. Circuit Theory*, vol. CT-9, pp. 24–29, Mar. 1962.

[18] TXR. Saeks, "A measure of testability and its application to test point selection theory," *Proc. 20th Midwest Symp. Circuits Systems.*, Aug. 1977.

[19] N. Sen and R. Saeks, "Fault diagnosis for linear systems via multifrequency measurement," *IEEE Trans. Circuits Syst.*, vol. CAS-26, pp. 457–465, July 1979.

[20] H. M. S. Chen and R. Saeks, "A search algorithm for the solution of multifrequency fault diagnosis equations," *IEEE Trans. Circuits Syst.*, vol. CAS-26, pp. 589–594, July 1979.

[21] R. W. Priester and J. B. Clary, "New measures of testability and test complexity for linear analog failure analysis," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 1088–1092, Nov. 1981.

[22] G. Fedi, S. Manetti, and M. C. Piccirilli, "Comments on Linear circuit fault diagnosis using neuromorphic analyzers," *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 483–485, Apr. 1999.

[23] G. Iuculano, A. Liberatore, S. Manetti, and M. Marini, "Multifrequency measurement of testability with application to large linear analog systems," *IEEE Trans. Circuits Syst.*, vol. CAS-23, pp. 644–648, June 1986.

[24] M. Catelani, G. Iuculano, A. Liberatore, S. Manetti, and M. Marini, "Improvements to numerical testability evaluation," *IEEE Trans. Instrum. Meas.*, vol. IM-36, pp. 902–907, Dec. 1987.

[25] R. Carmassi, M. Catelani, G. Iuculano, A. Liberatore, S. Manetti, and M. Marini, "Analog network testability measurement: A symbolic formulation approach," *IEEE Trans. Instrum. Meas.*, vol. 40, pp. 930–935, Dec. 1991.

[26] A. Liberatore, S. Manetti, and M. C. Piccirilli, "A new efficient method for analog circuit testability measurement," in *Proc. IEEE Instrumentation Measurement Technology Conf.*, Hamamatsu, Japan, May 1994, pp. 193–196.

[27] M. Catelani, G. Fedi, A. Luchetta, S. Manetti, M. Marini, and M. C. Piccirilli, " A new symbolic approach for testability measurement of analog networks ," in *Proc. MELECON '96*, Bari, Italy, May 1996, pp. 517–520.

[28] S. Manetti, "A new approach to automatic symbolic analysis of electric circuits," *Proc. Inst. Elect. Eng.*, pt. G, vol. 138, pp. 22–28, 1991.
[29] A. Liberatore and S. Manetti, "Network sensivity analysis via symbolic formulation," in *Proc. IEEE Int. Symp. Circuits Systems, ISCAS'89*, Portland, OR, May 1989, pp. 705–708.
[30] G. N. Stenbakken, T. M. Souders, and G. W. Stewart, "Ambiguity groups and testability," *IEEE Trans. Instrum. Meas.*, vol. IM-38, pp. 941–947, Oct. 1989.
[31] G. Fedi, R. Giomi, A. Luchetta, S. Manetti, and M. C. Piccirilli, "Symbolic algorithm for ambiguity group determination in analog fault diagnosis," in *Proc. ECCTD'97*, Budapest, Hungary, Aug. 1997.
[32] G. Fedi, S. Manetti, . C. Piccirilli, and J. A. Starzyk, "Determination of an optimum set of testable components in the fault diagnosis of analog linear circuits," *IEEE Trans. Circuits Syst. I*, vol. 46, pp. 779–787, July 1999.
[33] G. N. Stenbakken and T. M. Souders, "Test point selection and testability measures via QR factorization of linear models," *IEEE Trans. Instrum. Meas.*, vol. IM-36, pp. 406–410, June 1987.
[34] V. Brygilevicz, J. Wojciechowski, and J. A. Starzyk, "Testing of analog dynamic systems based on integral sensisitivity," in *Proc. ECCTD*, Stresa, Italy, Aug. 1999.
[35] A. Liberatore, A. Luchetta, S. Manetti, and M. C. Piccirilli, "A new symbolic program package for the interactive design of analog circuits," in *Proc. 1995 Int. Symp. Circuits Systems*, Seattle, WA, May 1995, pp. 2209–2212.

Janusz A. Starzyk (SM'83) received the M.S. degree in applied mathematics and the Ph.D. degree in electrical engineering from Warsaw University of Technology, Warsaw, Poland, in 1971 and 1976 respectively.

From 1977 to 1981 he was an Assistant Professor at the Institute of Electronics Fundamentals, Warsaw University of Technology, Warsaw, Poland. From 1981 to 1983 he was a Postdoctorate Fellow and Research Engineer at McMaster University, Hamilton, Ont. Canada. In 1983 he joined the Department of Electrical and Computer Engineering, Ohio University, Athens, where he is currently a Professor of EECS. He has cooperated with the National Institute of Standards and Technology in the area of testing and mixed signal fault diagnosis. He has been a consultant to AT&T Bell Laboratories, Sverdrup Technology, and Magnetek Corporation. In 1991 he was a Visiting Professor at the University of Florence, Florence, Italy. He was a Visiting Researcher at Redstone Arsenal, U.S. Army Test, Measurement, and Diagnostic Activity and at Wright Labs, Advanced Systems Research and Sensor ATR Technology Development. His current research is in the areas of neural networks, VLSI and VHDL design, computer-aided design of analog MOS circuits, and mixed signal testing.

Jing Pang (S'98) received the B.E. and M.E. degrees in electrical engineering from Xi'an Jiaotong University in 1993 and 1996, respectively.

She is currently a Ph.D. student in the Department of Electrical Engineering and Computer Science at Ohio University. Her current research focuses on computer-aided design of integrated circuits.

Stefano Manetti (M'96) received the degree in electronic engineering from the University of Florence, Florence, Italy, in 1977.

From 1977 to 1979, he was a Research Fellow at the Engineering Faculty, University of Florence. He was an Assistant Professor of applied electronics at the Academia Navale of Livomo, Livomo, Italy, from 1980 to 1983 and a Researcher in the Electronic Engineering Department, the University of Florence, from 1983 to 1987. From 1987 to 1994 he was an Associate Professor of network theory at the University of Florence. In 1994, he joined the University of Basilicata, Potenza, Italy, as a Full Professor of electrical sciences. Since November 1996, he has been a Full Professor of electrical sciences at the University of Florence. His research interests are in the areas of circuit theory, neural networks, and fault diagnosis of electronic circuits.

Dr. Manetti is a member of the ECS and AEI.

Maria Cristina Piccirilli (M'00) received the degree in electronic engineering from the University of Florence, Florence, Italy, in 1987.

From 1988 to 1990 she was a Research Fellow at the University of Pisa, Pisa, Italy. From March 1990 to October 1998 she was a Researcher at the Department of Electronic Engineering, the University of Florence. Since November 1998, she has been an Associate Professor of network theory in the Department of Electronics and Telecommunications, the University of Florence, where she works in the area of circuit theory, fault diagnosis of electronic circuits, and symbolic analysis.

Giulio Fedi (S'94–M'95) was bom in Milano, Italy, in 1970. He received the Laurea degree in electronic engineering (summa cum laude) from the University of Florence, Florence, Italy, in 1995 and the Ph.D. degree in electrical engineering from the University of Pisa, Pisa, Italy, in 1999.

He is now a Researcher at the Department of Electronics and Telecommunications, the University of Florence. His research activities are in the areas of analog circuit fault diagnosis using symbolic techniques and neural network applications to circuit modeling.

Dr. Fedi is a member of the AEI.