JAISCR, 2019, Vol. 9, No. 1, pp. 5 – 19
10.2478/jaiscr-2018-0021

A MLMVN WITH ARBITRARY COMPLEX-VALUED INPUTS AND A HYBRID TESTABILITY APPROACH FOR THE EXTRACTION OF LUMPED MODELS USING FRA

Igor Aizenberg[1], Antonio Luchetta[2], Stefano Manetti[2] and Maria Cristina Piccirilli[2]

[1]*Manhattan College, Riverdale, New York, USA*
[2]*Dept. of Information Engineering, University of Florence, Firenze, Italy*

E-mail: igor.aizenberg@manhattan.edu, {antonio.luchetta, stefano.manetti, mariacristina.piccirilli}@unifi.it

Submitted: 28th May 2017; Accepted: 19th October 2017

Abstract

A procedure for the identification of lumped models of distributed parameter electromagnetic systems is presented in this paper. A Frequency Response Analysis (FRA) of the device to be modeled is performed, executing repeated measurements or intensive simulations. The method can be used to extract the values of the components. The fundamental brick of this architecture is a multi-valued neuron (MVN), used in a multilayer neural network (MLMVN); the neuron is modified in order to use arbitrary complex-valued inputs, which represent the frequency response of the device. It is shown that this modification requires just a slight change in the MLMVN learning algorithm. The method is tested over three completely different examples to clearly explain its generality.

Keywords: Analog circuits, Complex-valued neural networks, Lumped models, Testability

1 Introduction

A fundamental characteristic of the complex-valued neural networks (CVNNs) is the proper treatment of phase and the information contained in it. These networks have been well described in [1] and have successfully been used for solving various real-world problems, such as landmine detection [2], forecasting of wind profiles [3], and medical image analysis [4]. In this paper, a new application of CVNNs is introduced, namely the lumped model identification of distributed parameter electromagnetic systems, and simultaneously the CVNN paradigm is further developed.

In many different operative situations, the identification of the circuital lumped model of a generic system (that could be distributed or physically heterogeneous) still represents an important challenge that can help the designer in many activities. This task can be difficult to approach due to: 1) the large number of parameters, 2) the nonlinear nature of the system to solve (nonlinear issues can appear in linear systems either), 3) problems in the choice of the parameters to be extracted. At the same time, a lumped model can be useful for several applications, as, for instance, the study of the transient response, the evaluation of the electromagnetic compatibility, the estimate of the harmonic content, the detection and localization of faults, the complete description of a complex structure. In the last years, a number of soft computing techniques have been applied for solving this problem, by means of artificial neural networks (ANNs) [5], genetic algorithms (GAs) [6], and particle swarm optimizers

(PSOs) [7, 8]. Most of these techniques, however, do not advantageously exploit the complex nature of the frequency response data. Moreover, they often do not take into account testability, ambiguity groups, and the sensitivity of the model to be identified.

In this paper, which significantly extends [9], where some promising preliminary results were obtained, a new approach for solving this problem is proposed, introducing a new neural tool based on a further generalization of the multi-valued neuron (MVN) [10] and multilayer neural network with multi-valued neurons (MLMVN) [11]. MVN is a neuron with complex-valued weights and inputs/output located on the unit circle. Its activation function depends only on the phase of the weighted sum and does not care about its magnitude. In this work, a new modification of a multi-valued neuron with arbitrary complex-valued inputs (MVN-ACVI) will be introduced. This kind of neuron, and consequently the network employing it, seems to be particularly useful in that kind of problem where input data are directly represented by the frequency response of the device or system under exam. This approach has a great advantage when solving system identification problems like those considered in this paper. It is important to point out that no conversion or normalization of the input data is needed and just a pretty small network with a low number of network parameters (and of neurons, accordingly) is usually enough to achieve an excellent performance.

The main idea is to use a set of measurements or simulations made on the device/system, taken at different values of electrical parameters and at different frequencies, to train a multilayer network with MVN-ACVI (MLMVN-ACVI) and then to estimate the electrical parameters of the lumped model, or in other words to “invert” the lumped model. A preliminary evaluation of the testability of the lumped model is performed for determining the solvability degree with respect to the circuit parameters, following the classic definition given in [12]. When testability evaluation is not sufficient to locate the parameters, also a sensitivity analysis can be performed. If the proposed lumped model is not suitable or not complete, the neural learning process does not converge. Hence, this kind of approach provides a useful information for the design process in the modeling and simulation phase.

The paper is organized as follows: the theoretical foundation is shown in Section 2, the network for the parameter extraction, together with testability analysis, is presented in Section 3, the simulation results over some application cases are described in Section 4.

2 Outlines on MVN and MLMVN

The discrete MVN was introduced in [10] as the further development of an element of multiple-valued threshold logic considered in [13]. In this paper, the continuous MVN has been used, which was introduced in [14]. It performs a mapping between n inputs and a single output. This mapping is described by a multi-valued function of n variables $f(x_1,\ldots,x_n)$:$O^n \quad O$, where O is a set of points $e^{i\varphi}$ located on the unit circle (i is the imaginary unit). The continuous MVN activation function is

$$P(z) = e^{iArgz} = z/|z|, \tag{1}$$

where z=w_0+w_1x_1+... w_nx_n is the weighted sum, and Arg(z) is the phase of z. Thus, for the continuous MVN, the output is the projection of the weighted sum on the unit circle, as it is determined by the activation function (1) (see Figure 1).

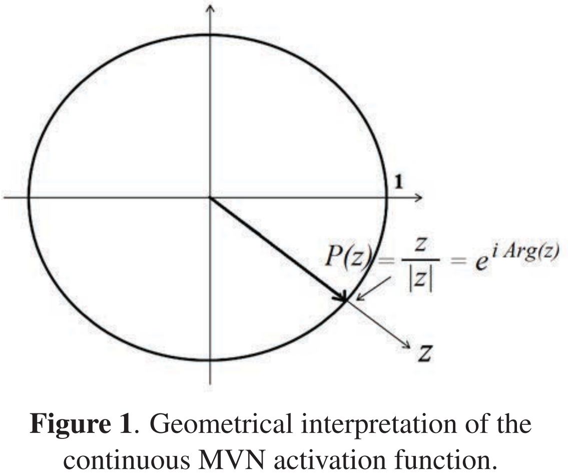


Figure 1. Geometrical interpretation of the continuous MVN activation function.

MVN and its learning algorithm are comprehensively reviewed in [15]. The MVN learning algorithm is based on the error-correction rule. According to this rule the weights are adjusted in one of the following two ways [15]

$$W_{r+1} = W_r + \frac{C_r}{n+1}\delta\overline{X},$$
$$W_{r+1} = W_r + \frac{C_r}{(n+1)|z_r|}\delta\overline{X}, \qquad (2)$$

where δ is the error, $\overline{X}$ is the neuron input vector with the conjugated components (here and further we assume that the unitary input $x_0 = 1$ is a "pseudo input" corresponding to the bias), r is the index of the learning step, n is the number of neuron inputs, W_r and W_{r+1} are the weighting vectors before and after correction, respectively, z_r is the weighted sum on the current (r^{th}) learning step, and C_r is the learning rate.

MLMVN, which was introduced in [11], is a multilayer neural network with a standard feedforward topology where neurons are integrated into layers, and the output of each neuron from the current layer is connected to the corresponding inputs of neurons from the following layer. The use of MVN as a basic neuron in MLMVN determines important distinctions and advantages of MLMVN compared to a classical multilayer perceptron (MLP). These advantages are the simplicity of learning and a significantly better generalization capability. The MLMVN learning algorithm is derivative-free. It is based on the same error-correction learning rule as the one of a single MVN. This MLMVN learning algorithm works as follows [14-16]. Let MLMVN contain m layers of neurons (m-1 hidden layers and the output layer whose index is m). First, the network error shall be calculated and backpropagated. This error backpropagation process is determined as follows

$$\delta^*_{km} = D_{km} - Y_{km},$$
$$\delta_{km} = \frac{1}{s_m} delta^*_{km},$$
$$\delta_{kj} = \frac{1}{s_j}\sum_{i=1}^{N_{j+1}} delta_{ij+1}(w_k^{ij+1})^{-1}; j = 1,...,m-1, \qquad (3)$$

where: D_{km} and Y_{km} are, respectively, the desired and actual outputs of the k^{th} neuron from the m^{th} (output) layer; δ^*_{km} is a network error calculated at the k^{th} neuron of the m^{th} (output) layer; δ_{km} is a local error of the k^{th} neuron from the m^{th} (output) layer; δ_{km} is a local error of the k^{th} neuron from the j^{th} layer; $s_j = N_{j-1}+1, j = 2,...,m; s_1 = n+1$ (where N_j is the number of neurons in the j^{th} layer and n is the number of network inputs); w_k^{ij} is the k^{th} weight of the i^{th} neuron from the j^{th} layer.

After the error backpropagation is done, the weights shall be adjusted layer by layer as follows

$$w_i^{\%km} = w_i^{km} + \frac{C_{km}}{(N_{m-1}+1)}\delta_{km}\overline{Y}^{\%}_{i,m-1} i = 1,...,N_{m-1}, \qquad (4)$$

$$w_i^{\%kj} = w_i^{kj} + \frac{C_{kj}}{(N_{j-1}+1)|z_{kj}|}\delta_{kj}\overline{Y}^{\%}_{i,j-1} i = 1,...,N_{m-1}, \qquad (5)$$

where $w_i^{\%km}$, $w_i^{\%kj}$ are corrected weights, N_j is the number of neurons in the j^{th} layer ($N_0 = n$, that is the number of network inputs), C_{kj} is the learning rate (it should be complex-valued in general, but it is empirically set to 1 in all known applications),$Y_{i,j-1}$ is the actual output of the i^{th} neuron of the (j-1)th layer (it is intended corrected when its superscript is ˜ and conjugated when its superscript is "bar"; $Y_{i,0} = Y^{\%}_{i,0}$, that is the i^{th} network input). Equation (4) determines the correction of the output neuron weights, while (5) determines the correction of the hidden neuron weights from the 1st hidden layer to the (m-1)th one.

The most important limitation of MLMVN is to be "slow" in the learning time with respect to the real value NN [16], but this drawback has been dramatically mitigated by a modification in the algorithm, introduced in [17], where it is also demonstrated the high efficiency of this tool for continuous function approximation. The version of the MLMVN presented in [17] has been used in the present work. MVN inputs and outputs (and MLMVN inputs and outputs accordingly) are located on the unit circle. This is very suitable for solving various classification and regression problems. In fact, if such a problem can be described by a function $f : S^n \to S$ where$S = [a,b] \subset$ R, it can easily be mapped (using a trivial normalization) to $f : \tilde{O}^n \to \tilde{O}$ where O is a set of points belonging to the unit circle and $\tilde{O} \subset O$. The circularity property of MVN/MLMVN inputs and outputs makes it also highly efficient for decoding signals in an

EEG-based brain-computer interface [18]. However, there are many applications where the parameters are complex-valued and not necessarily located on the unit circle. On such occasions, not only phase matters (as it is in MVN), but the magnitude is also very important.

Thus, it should be attractive to modify MVN in such a way that it will be able to operate over an arbitrary complex input (not necessarily reduced to unitary magnitude). This approach shall make it possible to preserve and properly handle the full information about phase and magnitude of frequency response samples.

Let us show that this modification of MVN requires just a slight change in its error-correction learning rule. Moreover, if this modified MVN with arbitrary complex-valued inputs (MVN-ACVI) will be used in the first hidden layer of MLMVN (to ensure acceptance of the arbitrary complex-valued inputs by the network), its learning algorithm will also require just a very slight modification. In this way, we may obtain MLMVN with the arbitrary complex-valued inputs (MLMVN-ACVI).

Let us use the idea developed in [18] where a complex-valued neuron with arbitrary complex-valued inputs and output was considered. However, in our case, while the neuron inputs are arbitrary complex numbers, its output is still located on the unit circle.

Let us consider first a single MVN-ACVI. Let D be a desired output of an MVN-ACVI and z its current weighted sum. Let the neuron error be determined by

$$\delta = D - z, \qquad (6)$$

like in [17]. Let us adjust the weights using the same error-correction learning rule, which was suggested in [19] for a complex-valued neuron with arbitrary inputs and outputs. We obtain the following learning rule (assuming x_0= 1)

$$w_i^{\%} = w_i + \frac{C\delta}{(n+1)} x^{-1} i = 0, 1, ..., n, \qquad (7)$$

where w_i is the current weight (to be adjusted), $\tilde{w}_i$ is the adjusted weight and C is a learning rate (complex-valued in general, but we still use $C = 1$ in all simulations as in the learning rule (2b)). It is interesting that if $x \in O$ (thus x is located on the unit circle), then $x^{-1} = \bar{x}$ and (7) coincides with the error correction learning rule (2a) for MVN. Actually, this means that (7) is a more general form of the error-correction learning rule and the first of the rules (2) is its particular form for MVN.

We may easily check that if the neuron weights are adjusted according to (7), then the adjusted weighted sum will be equal to the desired output. Let us find the adjusted weighted sum

$$\begin{aligned}
&z^{\%} = w_0^{\%} + w_1^{\%} x_1 + ... + w_n^{\%} x_1 = \\
&= \left(w_0 + \tfrac{\delta}{(n+1)} \right) + \left(w_1 + \tfrac{\delta}{(n+1)} x_1^{-1} \right) x_1 + ... \\
&+ \left(w_n + \tfrac{\delta}{(n+1)} x_n^{-1} \right) x_n = \\
&= w_0 + \tfrac{\delta}{(n+1)} + w_1 x_1 + \tfrac{\delta}{(n+1)} + ... + w_n x_n + \tfrac{\delta}{(n+1)} = \\
&= w_0 + w_1 x_1 + ... + w_n x_n + \delta = z + \delta = D
\end{aligned}$$

Evidently, (7) can easily be modified to get a rule, which generalizes the rule (2b)

$$w_i^{\%} = w_i + \frac{C\delta}{(n+1)\,|z|} x^{-1} \quad i = 0, 1, ..., n, \qquad (8)$$

where z is a current weighted sum. Evidently, the convergence theorem for the learning algorithm of MVN-ACVI based on either of the learning rules (7) or (8) can easily be proven in the same way as it was done for the MVN learning algorithm in [15].

Let us employ now MVN-ACVI in the first hidden layer of MLMVN, to ensure that a network accepts arbitrary complex-valued inputs. In other hidden layers (if any) and in the output layer we will still use MVN. Hence we obtain in this way MLMVN-ACVI. To train this network, we may use the learning algorithm with a batch acceleration step for the output neurons [17]. The only modification to this algorithm, which we have to make, is the adaptation of the learning rule (8) to the neurons from the first hidden layer. Hence we obtain the following algorithm.

1 The error of the network is calculated according to (6) for all the learning samples and then the output neuron weights are adjusted as described in [17]. This completes the batch acceleration step.

2 Then the learning process continues in the regular, serial manner, one sample after the other. The network error (errors in case of multiple output neurons) is calculated for a learning sample and this error (errors) is backpropagated to the hidden neurons according to (3).

3 The first hidden layer neuron weights are corrected according to the following rule, which follows from (8)

$$\tilde{w}_i^{k1} = w_i^{k1} + \frac{C_{k1}}{(n+1)\,|z_{k1}|}\delta_{k1} x_i^{-1} \quad ; i = 0, 1, ..., n.$$

4 For the rest of the hidden layers, the neuron weights are then corrected layer-by-layer according to (5).

The output neuron weights remain unchanged.

5 Repeat steps 2-4 for all the learning samples.

6 If the learning goal has not yet been reached (usually, the learning process continues until the mean squared error or the root mean squared error for a learning set drops below some tolerance value), then go back to step 1). If the learning goal is reached, then the learning process terminates.

It is important to mention that the suggested modification of MLMVN, which leads to MLMVN-ACVI, does not introduce any significant change in the proof of the convergence theorem. It can be proven in the same way as it was done for MLMVN in [15] (with some adjustments presented in [20]).

3 Complex-valued network for the parameter extraction

The main advantage of this kind of network is the ability to directly work over a set of frequency response values of the circuit, given in a complex form (magnitude and phase, or real and imaginary parts). As it will be shown below, employing a minimal amount of neurons and demonstrating a very fast convergence, this network can be a great tool to use in any kind of architecture that requires a complex-valued input evaluation.

3.1 Testability and ambiguity groups

Any mathematical model aimed at representing a real system relies on a set of parameters, specific for this particular system. An accurate parameter estimation is essential because it affects the model reliability, necessary for all activities associated with analysis, design, maintenance and fault diagnosis. However, it is not in general granted that all parameters of a system can be identified, hence a prerequisite to any parameter identification approach is the verification of its feasibility. Knowing a priori that a given parameter extraction method cannot successfully work allows saving time and resources, avoiding attempts to estimate undistinguishable parameters.

The solvability degree of the parameter identification problem can be determined by means of testability analysis [21-26], that, even if specific of parametric fault diagnosis of analog circuits, can also be applied to the parameter extraction problem. In order to understand the meaning of the testability concept, we refer to the parametric faults, i.e. the kind of faults in which the component parameters are out of their tolerance range. In this case, a measure of the solvability degree of the fault diagnosis problem may be naturally identified with the solvability degree of the fault diagnosis equations. These equations are obtained by comparing a number of input-output relationships with the set of corresponding measurements. In other words, after choosing some injection points and some test points, the theoretical and measured outputs are compared. The solvability measure quantifies the circuit attitude to be tested and is called testability. The parameter identification problem can take advantage of this methodology. In fact, the solution of the fault diagnosis equations is essentially a parameter extraction problem, in the sense that the faulty components are identified by the real values of their parameters. So, also in this case, the comparison of the input-output theoretical relationships with the corresponding sets of measurements provides the parameter identification equations. Therefore the fault diagnosis solvability degree, i.e. the testability, becomes a parameter identification solvability degree.

Much attention has been devoted to testability concept and several testability definitions have been given [12, 27-35]. However, the definition by Sen and Saeks [12], based on multifrequency measurements and linked to the rank of the Jacobian of the network function vector, is the most popular, due to its quantitative nature. It provides an upper bound, independent of the component value and frequency, to the performance of whatever algorithm employed for fault diagnosis. Of course, for what stated above, this definition is perfectly applicable to parameter identification. On the basis of this definition, the problem of parameter extraction can be formulated in the following way. Let us consider a lumped model with n_x inputs and n_y outputs. Let $\boldsymbol{p}$= $[p_1, \ldots, p_{np}]$ be the vector of the unknown circuit parameters and $n = n_x n_y$. The network functions

$$h_r(s,p) = \frac{\sum_{k=0}^{N} a_k^{(r)}(p)s^k}{\sum_{i=0}^{N} b_i(p)s^i}, r = 1,...,n, \qquad (9)$$

constitute the parameter extraction equations when equated to their measures for arbitrary values of the variable s. The unknowns of these nonlinear equations are the circuit parameters $\boldsymbol{p}$. The rank of the associated Jacobian matrix, evaluated for a given value $\boldsymbol{p}$* of the vector $\boldsymbol{p}$, contains the information about the solvability degree of the problem, i.e. the testability. Since the Jacobian is a matrix constituted by rational functions, the computation of its rank, and, hence, of testability, is not easy. In [25], it has been demonstrated that the testability value is the rank of the following numerical matrix Equation 10.

The testability T represents the maximum number of simultaneously identifiable parameters in correspondence to the given input-output sets and so it gives the solvability degree of the parameter identification problem at a circuit level. In other words, T allows to know a priori if a unique solution exists and, if it does not exist, T gives a quantitative measure of how far we are from it, i.e. how many parameters cannot be identified with the chosen set of measurements. Although this is a fundamental information, the knowledge of the solvability degree at a circuit level is generally incomplete, unless testability analysis is carried out also at a component level. For understating the importance of the component level analysis it can be observed that, in general, $T < n_p$. As stated above, this means that the parameter identification is not unique. The problem can be solved by assuming that, among the n_p parameters initially selected as unknown, n_p - T can be fixed a priori. The choice of these a priori known parameters can be based on heuristic considerations. In addition, if a network function is not very sensitive to a given parameter, this parameter can hardly be identified and it is worth assigning to it a fixed value. However, a more rigorous approach consists in the identification of testable groups (TGs) [23], i.e. the groups of T parameters that, if considered as simultaneously unknown, can be unambiguously identified. Now, since each parameter is univocally associated to a column of the matrix $C(\boldsymbol{p}_*)$, a group of parameters is testable if and only if the pertaining columns of $C(\boldsymbol{p}_*)$ constitutes a maximal linearly independent set. Furthermore it is equally important to individuate the canonical ambiguity groups (CAGs) [23], i.e. those minimal groups of parameters that, if assumed as simultaneously unknown, cannot be univocally identified. A group of parameters constitute a CAG if and only if the corresponding columns of $C(\boldsymbol{p}_*)$ constitute a minimal linearly dependent set. Testability analysis at a component level consists in the determination of the CAGs and the TGs. Its results can be then used for refining the parameter estimation strategy, in terms of I/O selection and choice of the set of parameters to be identified. In fact, since the solvability degree is linked with the selected network functions, the testability can be used to find the network functions of the model, i.e. the input and output sets, ensuring the maximum solvability. Furthermore, the knowledge of testability and ambiguity groups allows to determine the parameters that can be considered as unknowns. This is important for obtaining a unique solution [23].

Testability analysis at both circuit and component level can be carried out in a fully automated form using a computer program named TALIC (Testability Analysis for Linear Time-Invariant Circuits). It implements the algorithm proposed in [25] in such a way that, starting from the model under analysis, it can predict how many and which parameters can be unambiguously identified. TALIC uses symbolic techniques which help to easily determine the network function coefficients and their derivatives, necessary for the building of the matrix $C(\boldsymbol{p}*)$. The steps followed by the program can be summarized as follows:

STEP 1: Chosen n_x input points and n_y output points, evaluate the corresponding $n = n_x \times n_y$ network functions in fully symbolic form by means of SapWin [36].

STEP 2: From the SapWin output file, extract all the coefficients of the network functions.

$$
C(p^*) = \begin{bmatrix} \mathrm{B}(p^*) & 0 & \cdots & 0 & -\mathrm{A}^{(1)}(p^*) \\ 0 & \mathrm{B}(p^*) & \cdots & 0 & -\mathrm{A}^{(2)}(p^*) \\ \vdots & \vdots & \ldots & \vdots & \vdots \\ 0 & 0 & \cdots & \mathrm{B}(p^*) & -\mathrm{A}^{(n)}(p^*) \end{bmatrix} \bullet \begin{bmatrix} \frac{\partial}{\partial p_1} a^{(1)}(p^*) & \frac{\partial}{\partial p_2} a^{(1)}(p^*) & \cdots & \frac{\partial}{\partial p_{n_p}} a^{(1)}(p^*) \\ \frac{\partial}{\partial p_1} a^{(2)}(p^*) & \frac{\partial}{\partial p_2} a^{(2)}(p^*) & \cdots & \frac{\partial}{\partial p_{n_p}} a^{(2)}(p^*) \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial}{\partial p_1} a^{(n)}(p^*) & \frac{\partial}{\partial p_2} a^{(n)}(p^*) & \cdots & \frac{\partial}{\partial p_{n_p}} a^{(n)}(p^*) \\ \frac{\partial}{\partial p_1} b(p^*) & \frac{\partial}{\partial p_2} b(p^*) & \cdots & \frac{\partial}{\partial p_{n_p}} b(p^*) \end{bmatrix}
$$

$$
\mathrm{B}(p^*) = \begin{bmatrix} b_0(p^*) & 0 & 0 & \cdots & 0 \\ b_1(p^*) & b_0(p^*) & 0 & \cdots & 0 \\ b_2(p^*) & b_1(p^*) & b_0(p^*) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ b_N(p^*) & b_{N-1}(p^*) & b_{N-2}(p^*) & \cdots & b_0(p^*) \\ 0 & b_N(p^*) & b_{N-1}(p^*) & \cdots & b_1(p^*) \\ 0 & 0 & b_N(p^*) & \cdots & b_2(p^*) \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & 0 & \cdots & b_N(p^*) \end{bmatrix} \tag{10}
$$

$$
\mathrm{A}^{(r)}(p^*) = \begin{bmatrix} a_0^{(r)}(p^*) & 0 & 0 & \cdots & 0 \\ a_1^{(r)}(p^*) & a_0^{(r)}(p^*) & 0 & \cdots & 0 \\ a_2^{(r)}(p^*) & a_1^{(r)}(p^*) & a_0^{(r)}(p^*) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_N^{(r)}(p^*) & a_{N-1}^{(r)}(p^*) & a_{N-2}^{(r)}(p^*) & \cdots & a_0^{(r)}(p^*) \\ 0 & a_N^{(r)}(p^*) & a_{N-1}^{(r)}(p^*) & \cdots & a_1^{(r)}(p^*) \\ 0 & 0 & a_N^{(r)}(p^*) & \cdots & a_2^{(r)}(p^*) \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & 0 & \cdots & a_N^{(r)}(p^*) \end{bmatrix}
$$

STEP 3: Assign any value $\boldsymbol{p*}$ to $\boldsymbol{p}$.

STEP 4: Construct the testability matrix $C(\boldsymbol{p}_*)$.

STEP 5: Carry on a Singular Value Decomposition of $C(\boldsymbol{p}_*)$ as in [24].

STEP 6: Calculate:

a) Testability $T = rank\ [C(\boldsymbol{p}_*)]$

b) TGs

c) CAGs

The circuit is first drawn by using the SapWin schematic capture tool: in this phase, the input/output points are selected together with the parameters to be assumed as unknowns, whereas the remaining ones are fixed at an arbitrary value. Then the simulation is run, producing a file with the network functions. TALIC reads this file and carries out testability analysis according to the above outlined algorithm, thereby providing (for the selected sets of inputs, outputs and unknown parameters) the Testability T, the TGs, and the CAGs.

3.2 Parameter extraction procedure

When a lumped model has to be identified, matching the measurements carried out on a distributed structure, a MLMVN-ACVI is used to associate a measured output to a given equivalent circuit of the structure. Its block diagram is shown in Figure 2.

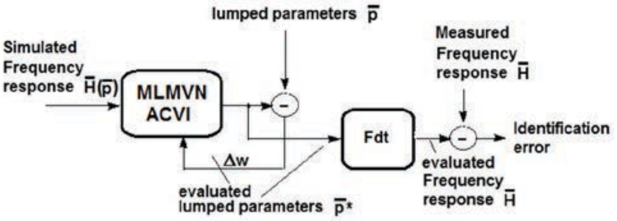

lumped parameters $\bar{p}$
Simulated Frequency response $\bar{H}(\bar{p})$
MLMVN ACVI
-
Measured Frequency response $\bar{H}$
Fdt
-
Identification error
Δw
evaluated lumped parameters $\bar{p}*$
evaluated Frequency response $\bar{H}$

Figure 2. General scheme of the identification system.

Once the architecture is chosen, the network function and the unknown parameters are fixed, then the unknown parameters become the MLMVN-ACVI outputs. In other words, the parameter values are determined without solving any system, and the network function is used for determining testability and ambiguity groups, necessary to identify the unknown parameters giving a unique solution.

The parameter extraction procedure can be summarized as follows:

1 *establish the lumped model equivalent circuit of the device;*

2 *calculate the testability and ambiguity groups and then evaluate which parameters should be assumed as unknowns;*

3 *if necessary, evaluate the sensitivity of the network response with respect to the electrical parameters, in order to tune the choice of the parameters left as variable;*

4 *generate an adequate number of samples to be used in the training phase;*

5 *train the MLMVN-ACVI part of the corresponding neural model (see Figure 2);*

6 *extract the parameters, as the output of the MLMVN-ACVI part of the whole system;*

7 *evaluate the quality of identification, by comparing the measured frequency response with the one calculated with extracted parameters.*

The lumped model equivalent circuit is established on the base of the collected data (measured or simulated) representing the relation $\boldsymbol{H}(\mathbf{p},\omega)$. The number of samples used in the training phase depends on the problem to be analyzed.

It is worth pointing out that the testability analysis, i.e. the testability evaluation and ambiguity group determination, is very efficiently performed by means of symbolic techniques [25, 36] without significantly increasing the computational cost of the whole parameter extraction system.

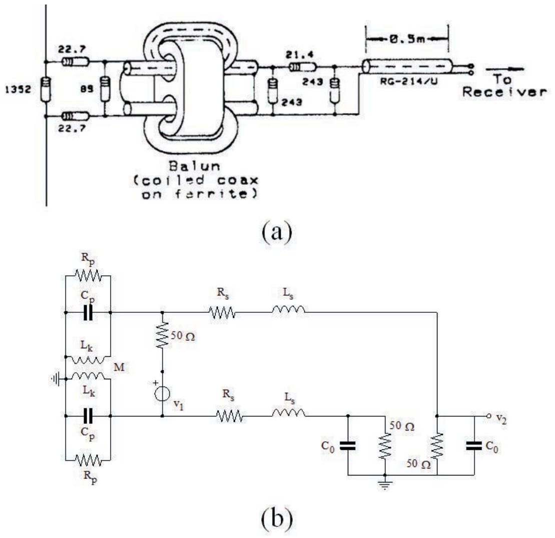
4 Simulation results

Three application examples are given in the following paragraphs.

4.1 Antenna Balun Circuit

As a first example, let us consider the parameter identification of a balun (**bal**anced to the

unbalanced circuit) in a Schwarzbeck half-wave dipole antenna shown in Fig. 3 [37]. An equivalent electrical lumped circuit of the balun is introduced by physical considerations, integrated by the imposition of the symmetry between the two conductors. Each parameter of the circuit model has a straightforward physical interpretation, L_k represents the self-inductance of each winding, C_p and R_p are the winding parasitic capacitance and resistance, respectively. The coupling factor of the two windings is k=M_k/L_k, so that M_k is the mutual inductance. R_s and L_s are the parasitic resistance and inductance of the connection, running from the coil to the isolated terminals, while C_0 represents the capacitance between each terminal and the ground.



(a)

(b)

Figure 3. Balun construction details (a) and equivalent circuit (b).

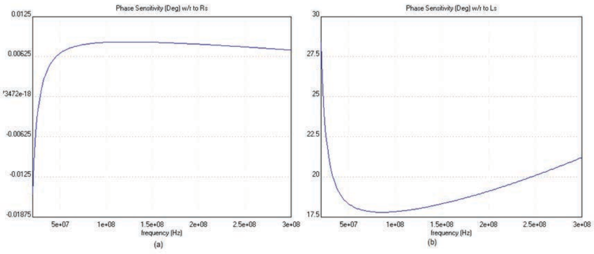


Figure 4. Phase-sensitivity versus frequency of the parameters Rs (a) and Ls (b) of the Balun circuit.

The reference values are used as initial values of the components and are obtained using an empirical approach over the physical model. The neural architecture is therefore used to extract and tune the component values, which give a frequency response as close as possible to the measured one. To utilize that, the following steps shall be made:

1 the circuit testability is evaluated (its value is maximum and equal to 7); all the elements can be potentially calculated;

2 a set of 2000 samples is generated using the circuit simulator SapWin [36], letting the component values randomly varying in a range of ±10% with respect to their reference values;

3 MLMVN-ACVI is trained over the set of generated samples, using a part of that (1400 samples) for the training and the remaining part of 600 samples for the validation.

4 the measurements are finally used for extracting the identified components of the model and to test the quality of the global approximation.

As for the step i), it should be mentioned that, since we use the testability maximum in our experiment, no ambiguity groups appear there. Because of that, all the components should be simultaneously varied in order to "cover" all the possible behaviors of the circuit response. This procedure has some drawbacks, the network becomes bulkier, slower and subjected to a higher identification error (recall that during the training, the error is evaluated just over the component values). In order to address this issue, a selection of the components through the sensitivity of the transfer function with respect to them is made. In the specific case, the curves of sensitivity (reported in Fig. 4) show the preponderant importance of the Rs and Ls parameters in the identification process. Based on this consideration, the resistive parameters keep their nominal values, while the reactive parameters are adjusted in the model via MLMVN-ACVI. The experimental results are reported in Table 1.

Table 1. Values of weights for Bonnor-Ebert gas sphere equation

L_k	M	R_p	R_s	L_s	C_p	c_o
(μH)	(nH)	$(k\Omega)$	$(m\Omega)$	(nH)	(pF)	(pF)
1.00	0.139	1.9	30.0	17.1	4.73	5.12

Figure 5 provides the comparison for the scattering parameter s_{31} (relevant to Magnitude and

Phase), showing the values calculated with the lumped parameter model and the values measured by instrumentation. The correspon-ding curves are almost perfectly overlapped, with a final mean error under 1%.

In order to give a deeper evaluation of the proposed solution, which we got using MLMVN-ACVI, we compared it with the ones gotten using classical MLP and standard MLMVN described in [14, 15], with no extended complex inputs and no QR-modified learning rule. The results are shown in Table 2. The comparison is provided in terms of the number of required learning epochs to reach the desired error, RMSE averaged over all the varied components, and RMSE calculated over the frequency response of the corresponding set of values in the given frequency range.

This kind of application of the proposed structure can be of particular interest in electromagnetic compatibility (EMC) measurements, given that an accurate identification of the circuit model of the balun permits to get insight into the limits of reliable operation of the network.

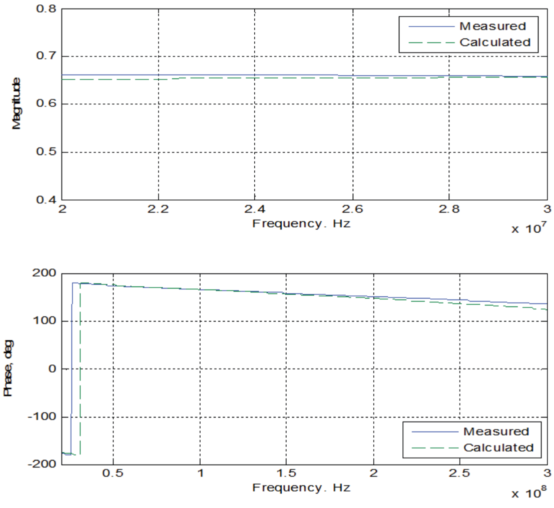


Figure 5. Frequency response amplitude and phase of the Balun circuit simulated by lumped parameter model (dashed curve) compared with the measured ones (solid curve).

Table 2. Comparison among the MLMVN-ACVI extraction procedure and others two networks for the balun circuit

.

	N^0 of epochs	RMSE over comp. value	RMSE over freq. resp.
MLMVN-ACVI	16	0.00251	0.0170
MLPNN	324	0.0570	0.1462
MLMVN	2038	0.00526	0.0545

Moreover, as it has been shown, MLMVN-ACVI introduced in this paper shows its superiority when compared to the traditional techniques. A great advantage of MLMVN-ACVI is its ability to directly elaborate the complex-valued frequency response. Finally, MLMVN-ACVI requires a drastically lower number of learning epochs to train it.

4.2 Lumped model of a microwave BP filter

Let us present the lumped parameter model of a two-post microwave filter, like that represented in Fig. 6, as a second example. The corresponding equivalent circuit is shown in Fig. 7.

The filter behaves like a bandpass in microwave range, with a very narrow band. In particular, the center frequency is around 13 GHz and the bandwidth is in the MHz range. As reported in [38], a very small variation in the geometric dimensions of the filter causes a consistent variation in the position of the center frequency. Therefore, to obtain a good resolution of the filter response, it is necessary to keep the frequency step small enough. A uniform sampling step is used in this case, obtaining 500 sampled data over the range from 12.5 to 13.5 GHz. These data are constituted by two values (magnitude and phase of the signal) and are taken as the complex input of any input MVN neuron.

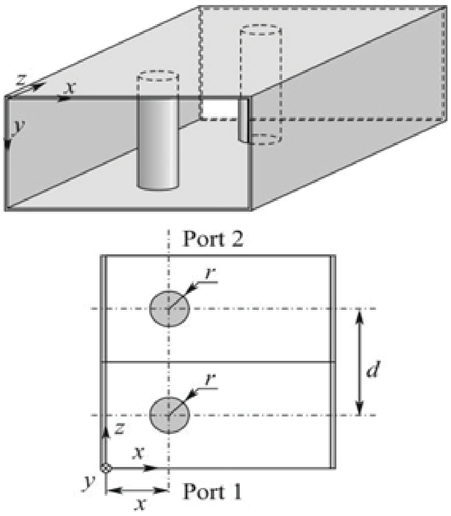


Figure 6. Geometry of the two-post microwave filter.

A hybrid finite-element–modal-expansion (FEM-ME) technique is used for the computation of the Generalized Scattering Matrix (GSM) of the device [39]. The geometry of the filter is completely represented by the three geometrical parameters indicated in Fig. 6: *x*, *r* and *d*. The equivalent lumped model is derived from the filter response, which is well represented by a 4^{th} order transfer function.

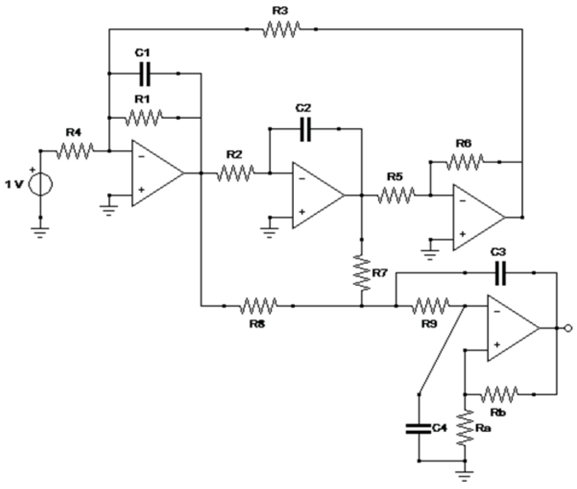


Figure 7. Equivalent lumped circuit of the microwave filter.

Table 3. Values of the circuit model parameters of mw filter as obtained from the MLMVN-ACVI extraction procedure

.

R1(Ω)	R2 (Ω)	R3 (Ω)	R$_4$(Ω)	R$_5$(Ω)
208.13	0.1217	0.1217	4.372	1.0
R$_6$(Ω)	R$_7$(Ω)	R$_8$(Ω)	R$_9$(Ω)	R$_a$(Ω)
1.007	0.2175	0.0773	0.2173	1.0
R$_b$(Ω)	C$_1$(pF)	C$_2$(pF)	C$_3$(pF)	C$_4$(pF)
4.537	0.9898	0.9908	0.9987	1.0

The testability analysis of the model is more complicated in this case. In fact, TALIC testability analyzer described before now gives a testability value T = 6, 272 ambiguity groups and 1306 testable groups.

Given that, only one among all the possible testable groups (each one including a combination of 6 elements) has been chosen: $T_G = \{C_1, C_2, C_3, R_1, R_6, R_9\}$. The dataset is generated using again the SapWin simulator [36], in order to obtain a dataset of 2000 learning examples (1500 are then used for training and 500 for validation). During the simulations the six components are randomly variated within a range ±20% with respect to the nominal value. The network is then trained following the procedure outlined in Section 3.2. In Fig. 8, a single output is reported in the frequency domain, relating to the values reported in Table 3. The best performing MLMVN-ACVI contains just 5 neurons in a single hidden layer and six output neurons (one per each adjusted parameter in the model). In Table 4, a comparison is reported, analogous to the previous example, with the other two paradigms.

Table 4. Comparison among the MLMVN-ACVI extraction procedure and others two networks for the mw filter circuit

.

	N^0 of epochs	RMSE over comp. value	Err% over freq. resp. (f$_0$, BW)
MLMVN-ACVI	36	0.0752	0.86, 3.75
MLPNN	514	0.1270	1.36, 8.20
MLMVN	3012	0.0926	0.98, 4.43

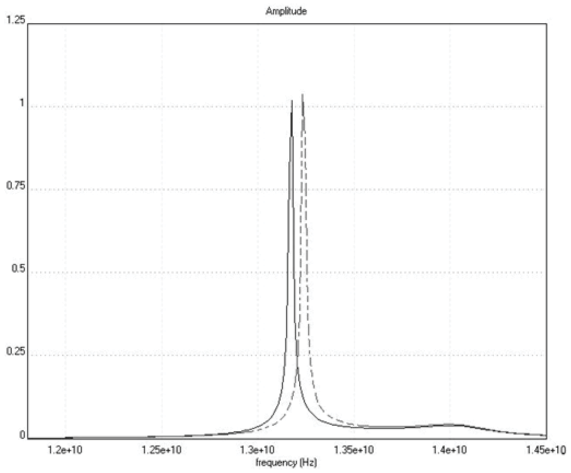


Figure 8. Frequency response amplitude of the microwave circuit simulated by lumped parameter model (dashed curve) compared with the one generated by simulator (solid curve).

In this case the curves are not perfectly overlapped because the band is very narrow, but the error over the central frequency and the one over the 3 dB bandwidth remain the best one with respect to other systems.

4.3 Lumped model of a transformer core

As a third example, the lumped parameter circuit of a laminated core of a three-phase transformer has been considered, which is modeled according to the principle of duality [40]. In this case, the neural identifier has been used to obtain the electrical/magnetic parameters μ_r (permeability), σ (conductivity) and the capacitance, relating to the equivalent circuit of Fig. 9. These parameters are related to the lumped components of the LRC equivalent circuit, by means of the geometrical characteristic of the windings and of the core of the transformer (cross-sectional area of the core, thicknesses of single lamination sheet, thickness of insulation layer of the sheet, height of the core, width of the core, length of the winding turn, diameter of the wire, thickness of insulation layer of the wire).

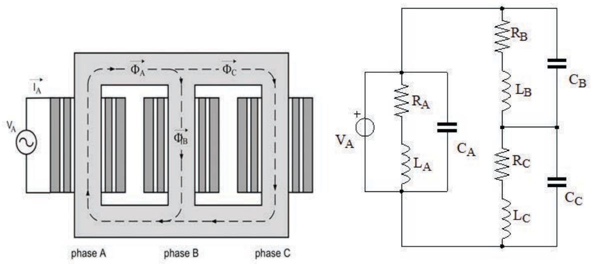


Figure 9. Three-phase transformer section and equivalent circuit of the transformer core.

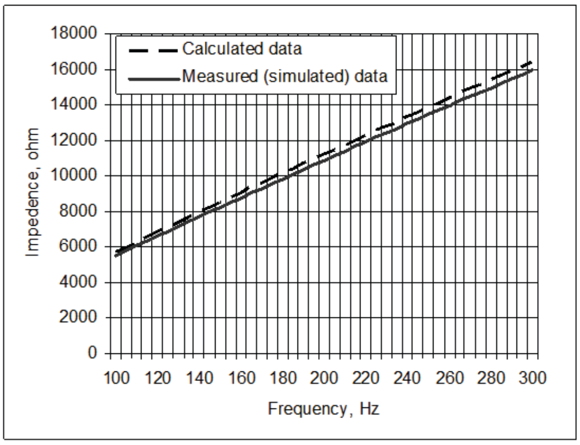


Figure 10. Comparison of the magnitude of the input impedance frequency responses in transformer core.

Also in this case the initial values of the components are obtained using an empirical approach. The neural architecture has been used to extract the values, and to give a frequency response as close as possible to the measured one. In similar way, the steps are:

1 the circuit testability is evaluated (its value is maximum and equal to 9); all the elements can be potentially calculated;

2 a set of 1200 samples is used, relating the electrical parameters L and R to the parameters μ_r and σ via analytical relations, reported, for instance, in [41], and letting the component values randomly varying in a range of ±10% with respect to their reference values; the set is generated in the frequency range 100-300 Hz, evaluating as output the input impedance of the equivalent model, that is $\boldsymbol{H} = \mathbf{Z}_A = \mathbf{V}_A/\mathbf{I}_A$;

3 MLMVN-ACVI is trained over the set of gener-

ated samples, using 1000 samples for the training and 200 samples for the validation.

The measurements are finally used for extracting the identified components of the model and for testing the quality of the global approximation (as reported in the Fig. 10).

The Table 5 summarizes the performance over parameter values and over frequency response in the same way of previous cases.

Table 5. Comparison among the MLMVN-ACVI extraction procedure and others two networks for the transformer core equivalent circuit.

	N^0 of epochs	RMSE over comp. value	RMSE over freq. resp.
MLMVN-ACVI	11	0.099	0.032
MLPNN	370	0.132	0.081
MLMVN	2830	0.097	0.031

5 Conclusion

The technique described in this paper is particularly suitable for the elaboration of data which come from a FRA approach of an electromagnetic system modeled by means of a lumped circuit. The combined use of different concepts developed by the authors (testability, symbolic analysis and representation, MVN-based neural networks with complex inputs) allows a rigorous and straightforward approach to an intricate issue. The results confirm the validity of the procedure.

References

[1] A. Hirose, Complex-Valued Neural Networks, 2^{nd} Edn., Springer, Berlin, Heidelberg, 2012.

[2] Y.Nakano, A.Hirose, Improvement of plastic landmine visualization performance by use of ring-CSOM and frequency-domain local correlation, IEICE Transactions on Electronics, vol. E92-C, no. 1, pp. 102-108, Jan. 2009.

[3] S. L. Goh, M. Chen, D. H. Popovic, K. Aihara, D. Obradovic and D. P. Mandic, Complex Valued Forecasting of Wind Profile, Renewable Energy, vol. 31, pp. 1733-1750, Sep. 2006.

[4] A. Handayani, A.B.Suksmono, T.L.R.Mengko, and A.Hirose, Blood Vessel Segmentation in Complex-Valued Magnetic Resonance Images with Snake Active Contour Model, International Journal of E-Health and Medical Communications, vol. 1, no. 1, pp. 41-52, Jan. 2010.

[5] G. Avitabile, B. Chellini, G. Fedi, A. Luchetta and S. Manetti, A neural architecture for the parameter extraction of high frequency devices, in Proc. of IEEE Int. Symposium on Circuits and Systems (ISCAS), Sidney, Australia, 2001, pp. 577-580.

[6] V. Rashtchi, E. Rahimpour, and E. M. Rezapour, Using a genetic algorithm for parameter identification of transformer R-L-C-M model, Electrical Engineering, 88, no.5, 417-422, June 2006.

[7] A. Shinterimov, W. J. Tang, W. H. Tang, and Q. H. Wu, Improved modelling of power transformer winding using bacterial swarming algorithm and frequency response analysis, Electric Power Systems Research, 80, no. 9, 1111-1120, Sep. 2010.

[8] W. H. Tang, S. He, Q. H. Wu and Z. J. Richardson, Winding deformation identification using a particle swarm optimiser with passive congregation of power transformes. Int. J. of Innovations in Energy Systems and Power, 1(11), 46-52, 2006.

[9] I. Aizenberg, A. Luchetta, S. Manetti and M.C. Piccirilli, " System Identification using FRA and a modified MLMVN with Arbitrary Complex-Valued Inputs", Proc. of the IEEE International Joint Conference on Neural Networks (IJCNN'16), Vancouver, July 2016, pp. 4404-4411.

[10] N.N. Aizenberg and I.N. Aizenberg, "CNN Based on Multi-Valued Neuron as a Model of Associative Memory for Gray-Scale Images", Proceedings of the Second IEEE International Workshop on Cellular Neural Networks and their Applications, Munich, October 14-16, 1992, pp.36-41.

[11] I. Aizenberg, and C. Moraga, Multilayer feedforward neural network based on multi-valued neurons (MLMVN) and a backpropagation learning algorithm, Soft Computing, 11, no. 2, 169-183, Jan. 2007.

[12] N. Sen and R. Saeks, Fault Diagnosis for Linear System via Multifrequency Measurement", IEEE trans. Circuits and Systems, vol. CAS-26, pp.457 - 465, 1979.

[13] N. N. Aizenberg, L. Ivaskiv, D. A. Pospelov, and G.F. Hudiakov, Multivalued Threshold Functions. II. Synthesis of Multivalued Threshold Elements, Cybernetics and Systems Analysis, vol. 9, no. 1, pp. 61-77, Jan 1973.

[14] I. Aizenberg,, C. Moraga, and D. Paliy, A Feedforward Neural Network based on Multi-Valued Neurons, In Computational Intelligence, Theory and Applications. Advances in Soft Computing, XIV, (B. Reusch - Ed.), Springer, Berlin, Heidelberg, New York, 2005, pp. 599-612.

[15] I. Aizenberg, I., Complex-Valued Neural Networks with Multi-Valued Neurons. Berlin: Springer-Verlag Publishers, 2011.

[16] I. Aizenberg, D. Paliy, J. Zurada, and J. Astola, Blur identification by multilayer neural network based on multivalued neurons, IEEE Transactions on Neural Networks, vol. 19, no. 5, 883-898, May 2008.

[17] I. Aizenberg, A. Luchetta and S. Manetti, S, A modified learning algorithm for the multilayer neural network with multi-valued neurons based on the complex QR decomposition, Soft Computing, vol. 16, no. 4, 563-575, Apr. 2012.

[18] N.V.Manyakov, I. Aizenberg, N. Chumerin, and M. Van Hulle, Phase-Coded Brain-Computer Interface Based on MLMVN, book chapter in Complex-Valued Neural Networks: Advances and Applications (A. Hirose – Ed.), Wiley, 2012, pp. 185-208.

[19] I. Aizenberg, Hebbian and Error-Correction Learning for Complex-Valued Neurons, Soft Computing, vol. 17, no. 2, pp. 265-273, Feb. 2013.

[20] I. Aizenberg, Adjustments to the proofs of the convergence theorems, available online at http://www.eagle.tamut.edu/faculty/igor/CVNNMVN_book_Convergence_Proofs_Adjustments.htm (2013).

[21] G. Fedi, A. Luchetta, S. Manetti, and M. C. Piccirilli, A new symbolic method for analog circuit testability evaluation, IEEE Transactions on Instrumentation and Measurement, vol. 47, no. 10, 554-565, Apr. 1998.

[22] A. Liberatore, S. Manetti, and M. C. Piccirilli, A new efficient method for analog circuit testability measurement. Proc. of IMTC'94, Hamamatsu, Japan, pp. 193-196, 1994.

[23] G. Fedi, S. Manetti, M. C. Piccirilli, and J. Starzyk, Determination of an optimum set of testable components in the fault diagnosis of analog linear circuits, IEEE Transactions on Circuits and Systems - Part I, 46, 779-787, Jul. 1999.

[24] S. Manetti, and M. C. Piccirilli, A singular-value decomposition approach for ambiguity group determination in analog circuits, IEEE Transactions on Circuits and Systems – Part I, vol. 50, no. 4, 477-487, Apr. 2003.

[25] G. Fontana, A. Luchetta, S. Manetti and M. C. Piccirilli An unconditionally sound algorithm for testability analysis in linear time-invariant electrical networks, Int. J. On Circuit Theory and Applications, vol. 44 no. 6, pp. 1308-1340, 2016.

[26] G. Fontana, A. Luchetta, S. Manetti, M. C. Piccirilli, A Fast Algorithm for Testability Analysis of Large Linear Time-Invariant Networks, IEEE Trans. on Circuits and Systems – Part I, DOI: 10.1109/TCSI.2016.2645079, 2017.

[27] R. Berkowitz, Conditions for network-element-value solvability, IRE Trans. Circ. Theory, vol. 9, pp. 24-29, 1962.

[28] W. J. Deika, A review of measures of testability for analog systems, Proc. Int. Autom. Test Conf. (AUTOTESTCON), 1977, pp. 279-284.

[29] R. W. Priester, J. B. Clary, New measures of testability and test complexity for linear analog failure, IEEE Trans. Circuits Syst., vol. 30, pp. 884-888, 1981.

[30] C. Lin, Z. F. Huang, R. Liu, Topological conditions for single-branch-fault, IEEE Trans. Autom. Control, vol. 28, pp. 689-694, 1983.

[31] G. N. Stenbakken, T. M. Souders, Test-point selection and testability measures via QR factorization of linear models, IEEE Trans. Instrum. Meas., vol. 36, pp. 406-410, 1987.

[32] G. N. Stenbakken, T. M. Souders and G. W. Stewart, Ambiguity groups and testability, IEEE Trans. Instrum. and Meas., vol. 38, pp.941-947, 1989.

[33] W. H. Huang and C. L. Wey, Diagnosability analysis of analogue circuits, Int. J. Circ. Theor. Appl., vol. 26, pp. 439–451, 1998.

[34] J. A. Starzyk and M. A. El-Gamal, Diagnosability of analog circuits-a graph theoretical approach, Proc. IEEE Int. Symp. Circuits and Systems, 1988, pp. 945-948.

[35] B. Cannas, A. Fanni and A. Montisci, Testability evaluation for analog linear circuits via transfer function analysis Proc. IEEE Int. Symp. Circuits and Systems, 2005, pp. 992-995.

[36] F. Grasso, A. Luchetta, S. Manetti, M. C. Piccirilli and A. Reatti, SapWin 4.0–a new simulation program for electrical engineering education using symbolic analysis, Computer Applications in Engineering Education, vol. 24 no. 1, pp. 44-57, 2016.

[37] W. S. Bennett, Properly Applied Antenna Factors, IEEE Trans. on Electromagnetic Compatibility, EMC-28, 1, pp. 2-6, Feb. 1986.

[38] G. Fedi, S. Manetti, G. Pelosi and S. Selleri, FEM-trained artificial neural networks for the analysis and design of cylindrical posts in rectangular waveguide, Electromagnetics, 22, 323–330, 2002.

[39] G. Pelosi, R. Coccioli, and S. Selleri, Quick finite elements method for electromagnetic waves, (pp. 89–113). London: Artech House, 1998.

[40] Shinterimov, W.H. Tang, and Q.H. Wu, Transformer Core Parameter Identification Using Frequency Response Analysis, IEEE Trans. on Magnetics, vol. 46, pp. 141-149, January 2010.

[41] D. Roger, E. Napieralska-Juszczak, and A. Henneton, High frequency extension of non-linear models of laminated cores, Int. J. Comput. Math. Electr. Electron. Eng., vol. 25, pp. 140-156, 2009.

Igor Aizenberg received the M.Sc. degree in Mathematics from Uzhhorod National University (Ukraine) in 1982, and the Ph.D. degree in Computer Science from the Dorodnicyn Computing Center of the Academy of Sciences of the USSR (Moscow, Russia) in 1986, respectively. In 1982-1990, he was a Research Scientist with the Institute for Information Transmission Problems of the Academy of Sciences of the USSR (Moscow, Russia). In 1990-1993, he was an Assistant Professor and then in 1993-1996 and 1998-1999, he was an Associate Professor with the Department of Cybernetics at Uzhhorod National University (Uzhhorod, Ukraine). In 1996-1998, he was a Research Scientist with the Department of Electrical Engineering at the Catholic University of Leuven (Leuven, KU Leuven, Belgium). In 1999-2002, he was a Chief Research Scientist and VP Research at the company "Neural Networks Technologies" (Israel). In 2002-2006, he was a Visiting Research Professor at the University of Dortmund (Dortmund, Germany) and Tampere University of Technology (Tampere, Finland). In 2006-2016, he was Professor with the Department of Computer Science and Director of the Computational Intelligence Lab, Texas A&M University-Texarkana (Texarkana, TX, USA). Since August 2016 he is Professor and Chair of the Department of Computer Science at Manhattan College (Riverdale, NY, USA). Dr. Igor Aizenberg has also recently served as an invited Visiting Professor at the Polytechnic of Porto (Porto, Portugal), National Sun Yat Sen University (Kaohsiung City, Taiwan), Masaryk University (Brno, Czech Republic), and National Polytechnic Institute of Mexico (Mexico City, Mexico). Research interests include classification, pattern recognition, intelligent image processing, and spectral techniques.

Antonio Luchetta (M'96) graduated in electronic engineering at the University of Florence, Italy, in 1993. From 1995 to 2005, he was an Assistant Professor in the Dept. of Environmental and Physics Engineering of the University of Basilicata, and then in the Department of Electronics and Telecommunications of the University of Florence, where he is at present an Associate Professor in the Department of Information Engineering. He teaches "Circuit Theory for Computer Engineering" and "Fundamentals of Electrical Engineering for Civil and Environmental Engineering". His research interests are in the areas of circuit theory, neural networks, symbolic analysis and simulation of analog circuits. Prof. Luchetta is a member of AEIT.

Stefano Manetti (M'96) received the Laurea degree in electronic engineering from the University of Florence in 1977. From 1977 to 1979, he was a Research Fellow with the Engineering Faculty, University of Florence. From 1980 to 1983, he was an Assistant Professor of Applied Electronics with the Accademia Navale of Livorno, Italy. He was a Researcher from 1983 to 1987 and an Associate Professor of network theory from May 1987 to 1994 with the Electronic Engineering Department, University of Florence. In 1994, he was a Full Professor of Electrical Sciences with the University of Basilicata, Potenza, Italy. Since November 1996, he has been a Full Professor of Electrical Sciences with the University of Florence. He has been the Dean of the Engineering Faculty of the University of Florence from 2009 to 2013. His research interests are in the areas of circuit theory, neural networks, fault diagnosis of electronic circuits, and symbolic analysis of analog circuits. Prof. Manetti is a member of AEIT.

Maria Cristina Piccirilli (M'00) received the Laurea degree in electronic engineering from the University of 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 in the Department of Electronic Engineering, University of Florence. Since November 1998, she has been an Associate Professor of Network Theory in the same Department, now Department of Information Engineering, where she works in the area of circuit theory, analog filters, fault diagnosis of electronic circuits, neural networks and symbolic analysis of analog circuits. Prof. Piccirilli is a member of AEIT.