

# Systematic Methods for the Implementation of Nonlinear Wave-Digital Structures

Augusto Sarti, *Member, IEEE*, and Giovanni De Sanctis

**Abstract**—Wave-digital (WD) structures containing adaptors with memory (characterized by port reflection filters) and nonlinear elements are suitable for the modeling of a wide range of nonlinear circuits and physical structures. In this paper, we propose two methods for automating the construction of algorithms that efficiently implement such structures, starting from their symbolic description. The former is based on the solution of state-space equations, while the latter is based on direct structural inspection. The state-space approach starts from the blockwise construction of a tableau matrix for the direct implementation of a generic WD structure and, for this reason, is here referred to as the *wave tableau* (WT) method. It has very general applicability as it works for a generic WD structure. The second technique (binary connection tree) implements a WD structure through a direct inspection (scanning) of the treelike topological representation of the reference model. Although valid for a slightly less general range of cases, this approach turns out to be much more efficient and flexible than that of the WT method. Such methods are particularly interesting for an interactive and immediate prototyping of physical models for the synthesis of sounds as they bring nonlinear WD structures with dynamic adaptors to a level of practical usability for a wide range of users while enabling the modeling of a wide variety of time-varying nonlinear physical models in an automatic fashion. The proposed solutions have been extensively tested on applications for the automatic modeling of acoustic interactions of musical interest.

**Index Terms**—Physical modeling, sound synthesis, wave-digital filters (WDFs).

## I. INTRODUCTION

THE DESIGN of wave-digital (WD) filters [1] (WDFs) is an interesting example of optimal physical modeling of digital structures. With this approach, in fact, we start with an analog filter, and from there, we develop a digital system that preserves most of the desirable properties of the reference analog circuit, such as passivity, losslessness, stability, minimal parameter sensitivity, etc. In the past three decades, this approach has reached an advanced level of maturity, and a plethora of applications and extensions have appeared in literature. In particular, the aforementioned “physical optimality” has recently attracted the attention of researchers in the field of

sound synthesis through physical modeling [2]–[4]. This has triggered a great deal of activity in the direction of modeling WD structures that include nonlinearities [5]–[8] and whose scattering cells and adaptors are not instantaneous [9]–[11]. It is important to point out, in fact, that physical mechanisms for the generation of sounds can be typically seen as based on the nonlinear interaction between two otherwise linear systems, at least one of which being a resonating structure [2]. For example, a piano hammer interacts with a string through a contact condition (a Heaviside function) and is felt with limited compressibility [5], [12]; the bow of a violin interacts with a string with a stick-and-slide mechanism that can be described through a nonlinear and discontinuous relationship between force and velocity at the contact point (Helmholtz model)[12], [13], etc.

Modeling nonlinear interactions between physically modeled blocks can be seen as a nonlinear-circuit-modeling problem. One major advantage in working with a circuit is that blocks communicate through ports, each characterized by a pair of dual variables (an *across/through* pair of variables such as voltage/current). This fact, among many other advantages, allows us to monitor the flow of energy that is exchanged between blocks and to ensure the preservation of the stability conditions of the analog-reference circuit. This is achieved by guaranteeing that all block interconnections inherently satisfy the global laws of continuity, i.e., the Kirchhoff laws, and that the analog-to-digital mapping is performed through a bilinear transformation. WDF theory [1] offers a strategy for modeling local equations (block descriptions) as well as global continuity laws in an explicit fashion. This approach uses scattering variables (incident and reflected waves) instead of voltage/current pairs to implement the Kirchhoff laws in explicit form. The user can then apply such laws using an interconnection of properly defined adaptors which results in a computable and energy-transparent structure that implements the physical interconnection between prepackaged blocks.

The literature of sound synthesis through physical modeling is rich with solutions that are closely related to WDFs. A well-known example of this sort is given by the Digital Wave-Guide networks (DWGs) [3], which are particularly suitable for modeling acoustic resonating structures, and whose signals are compatible with those that are used in WDFs. As a matter of fact, DWGs can be seen as close relatives to multidimensional WDF[23] (as explained in detail in [24]).

Hybrid WDF/DWG [4] structures represent a good solution to the problem of sound synthesis by physical modeling as, besides referring to an acoustic instrument, they are based on a local (block-based) discretization of the physical elements

Manuscript received June 29, 2006; revised June 19, 2007, December 04, 2007, and April 04, 2008. First published July 22, 2008; current version published February 11, 2009. This paper was recommended by Associate Editor A. Kummert.

A. Sarti is with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, 20131 Milano, Italy (e-mail: sarti@elet.polimi.it).

G. De Sanctis is with the Sonic Arts Research Centre, Queen’s University Belfast, BT7 1NN Belfast, U.K. (e-mail: gdesanctis01@qub.ac.uk).

Digital Object Identifier 10.1109/TCSL.2008.2001801

that constitute the analog model. In other words, these solutions open the way to the development of a flexible synthesis technique based on the interconnection of predefined building blocks. Traditional WDF/DWG structures, however, are inherently linear; therefore, they cannot easily model nonlinear excitational interactions other than nonlinear frictions (nonlinear resistors [15]). Nonetheless, some clever *ad hoc* solutions exist for including other types of nonlinearities in WD structures [9], [10]. In the past few years, however, new solutions and methodologies have emerged, which generalize the WDF/DWG framework in order to encompass a wide class of nonlinearities with memory, without giving up flexibility and modularity in the synthesis strategy [4], [14], [16]. An embryo of these ideas was first presented in [16] and then more completely formalized in [14]. In this approach, nonlinear WD structures are characterized by WD adaptors that are able to perform not just changes of port resistances but also changes of port filters. The adaptors are thus inherently “dynamic” as they are based on reflection and transmission filters instead of simple coefficients. Dynamic adaptors allow us to accommodate a wider range of NonLinear Elements (NLEs) with memory (nonlinear reactances, algebraic nonlinearities, etc.) in a WD structure and model them as instantaneous NLEs. Adaptors with memory can count on many of the properties of their instantaneous counterpart (traditional WDF adaptors), specifically passivity, losslessness, and conservation of energy. There are some properties, however, that do not hold true, such as those concerning coefficient quantization [1]. Nonetheless, in the rest of this paper, we will adopt the term “adaptor” in both dynamic and instantaneous cases, for the sake of simplicity and in accordance to [14].

In this paper, we refer to the formulation of WD structures proposed in [14], which enables the modeling of a wide variety of nonlinear interactions between physically modeled systems. We propose two systematic and automatic techniques for the construction of sound-synthesis algorithms based on the physical interaction of blocks. Our modeling solutions start from a symbolic description of the physical model and automatically construct an efficient algorithm for the synthesis of sounds, which preserves the stability properties of the analog-reference model. The symbolic model description is provided in a similar way as a classical nonlinear-circuit simulation software, i.e., a list of blocks, their interconnection topology, their initial conditions, their parameters, and their input/output signals/ports. After parsing and validating this description, the system automatically builds a WD structure [14] that implements the reference physical model. This approach enables the interaction between WD blocks in compliance with the global laws of dynamics (Kirchhoff laws). This means that even those models that exhibit a critical oscillating behavior (e.g., a subharmonic chaotic oscillator) do not need any oversampling to be correctly modeled [14].

This paper is organized as follows. In Section II, we discuss the topological structure of a network of WD adaptors. We show, in particular, that the memory that is embedded in such adaptors can be extracted from the whole structure so that the interconnection network can be seen as a memoryless macrojunction that connects the WD blocks through special two-port WD scatterers with memory (e.g., mutators). We then present, in

Sections III and IV, our two solutions for automating the implementation process of such WD structures. A comparative evaluation of the two solutions is presented in Section V. Conclusions and future works are discussed in Section VI.

## II. STRUCTURAL ASSUMPTIONS

It is well known that the interconnection between one-port WD elements through parallel and serial adaptors gives rise to computable WDFs, because the corresponding signal-flow diagram does not contain any delay-free directed loops [1], [22]. This realizability condition implies that any connection between two ports does not lead to any delay-free directed loop and that no other delay-free directed loop can be created via some outer path [1]. This is indeed guaranteed if the network of adaptors has a *treelike* structure, which is true for most reference physical systems that are encountered in musical acoustics.

The hypothesis of treelike structure is not a general one. The Jaumann structure [22], for example, results in a topology that is not treelike due to a particular interconnection of transformers. This WD structure, however, is a noncomputable one. So far, we have not found any example of computable WD structures that do not exhibit a treelike topology. We can therefore reasonably assume that, even if one such structure existed, it would be a rather unusual one. However, we can expect that these configurations could be implemented with *ad hoc* solutions.

Such considerations hold true for interconnections of dynamic adaptors. These elements are general types of (parallel or series) adaptors whose reference resistances  $R_i$  are replaced by reference filters [or reference transfer functions (RTFs)]  $R_i(z)$ ; therefore, incident port waves are subject to noninstantaneous (filtered) scattering [14]. RTFs can thus be thought of as reference port impedances. A treelike network of (dynamic) adaptors is here referred to as a (dynamic) macroadaptor (MA), and a port that turns out to exhibit no instantaneous reflection is called an *adapted* port. As reflected waves at this port are either absent or delayed, this port can be freely connected to another port without causing computability problems. Realizability issues (a nonadapted port can only be connected to an adapted one) imply that the MA can only have up to one adapted port, just like the adaptors that it is made of. The WD structures that we consider in this paper are made of a number of bipoles connected together through such MA. As there is only one adapted port per MA, we can only accommodate one NLE by connecting it to that port.

The fact that a nonlinear WD circuit of this sort can only have one NLE is a typical limitation of WD structures, including WDFs [15], which can be overcome in an exact fashion by either solving an implicit equation or by lumping together portions of the circuit in order to incorporate multiple NLEs into a single and more complex NLE. Fortunately, that of multiple NLEs is not frequently encountered in applications of musical acoustics, where multiple nonlinearities (e.g., contact and compressibility conditions) are usually connected in a “delayed” fashion with each other (through distributed-parameter resonating structures). As a consequence, a complete physical model of a musical instrument usually consists of a “delayed” interconnection of several MAs, each connected to a different NLE.

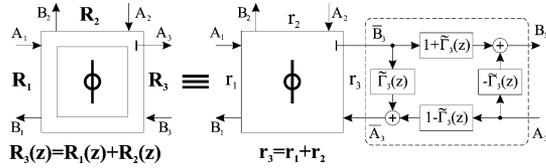


Fig. 1. Any three-port series adaptor can always be implemented as a standard-series WDF adaptor, whose adapted port is connected to a two-port scattering cell. The double-box symbol [14] is used here to distinguish between an adaptor with memory and a traditional (instantaneous) one, whereas the terms  $r_i$ ,  $i = 1, 2, 3$  are the instantaneous part of the corresponding  $R_i(z)$ .

In this paper, however, we will not consider the case of multiple MAs. In fact, if two MAs are connected together directly, then they form a larger MA. If they are connected together through a delaying multiport element, then the two WD structures can be independently analyzed and implemented.

We recently proved [18], [19] that a computable treelike interconnection of adaptors with memory is completely equivalent to a memoryless MA whose outer ports are connected to scattering cells with memory [or dynamic scattering cells (DSCs)] [11]. The proof of this equivalency is constructive and can be done in two steps: 1) Replace all three-port parallel/series adaptors with memory with memoryless adaptors of the same type, whose adapted port is connected to a properly defined DSC and 2) have all the inner DSC (those that end up in between two adaptors) slide outward until they reach the periphery, according to specific sliding rules.

In the next section, we will briefly illustrate such procedure (more details can be found in [18] and [19]).

#### A. Memory Extraction From Adaptors

Let us consider a three-port dynamic series adaptor with rational, causal, and stable RTFs  $R_k(z)$ ,  $k = 1, 2, 3$  being the port index. This adaptor can always be implemented as a standard series WDF adaptor, whose adapted port is connected to a DSC, as shown in Fig. 1, where  $\tilde{\Gamma}_3(z)$  is obtained from the reference RTF of the adapted port by removing the instantaneous I/O connection. More specifically, the reflection filters are

$$\Gamma_k(z) = \frac{2R_k(z)}{\sum R_i(z)} = \gamma_k + \tilde{\Gamma}_k(z) \quad (1)$$

from which we can extract a constant  $\gamma_k$ , while the rest can be written as

$$\tilde{\Gamma}_k(z) = z^{-1}\hat{\Gamma}_k(z)$$

where  $\hat{\Gamma}_k(z)$  is assumed as causal and stable. A similar result can be derived for the parallel adaptor. A proof of this statement is reported in [18] and [19]. It is worth noting that the scattering cells obtained by the memory extraction are very peculiar. In fact, they do not present any computability problem, as their scattering filter does not have any constant term.

#### B. Memory Extraction From a MA

It is now possible to solve the more general problem of extracting the dynamics from a MA. The aim is to find an equivalent structure that is made of an instantaneous MA and a number of DSCs connected to some (or all) of the ports.

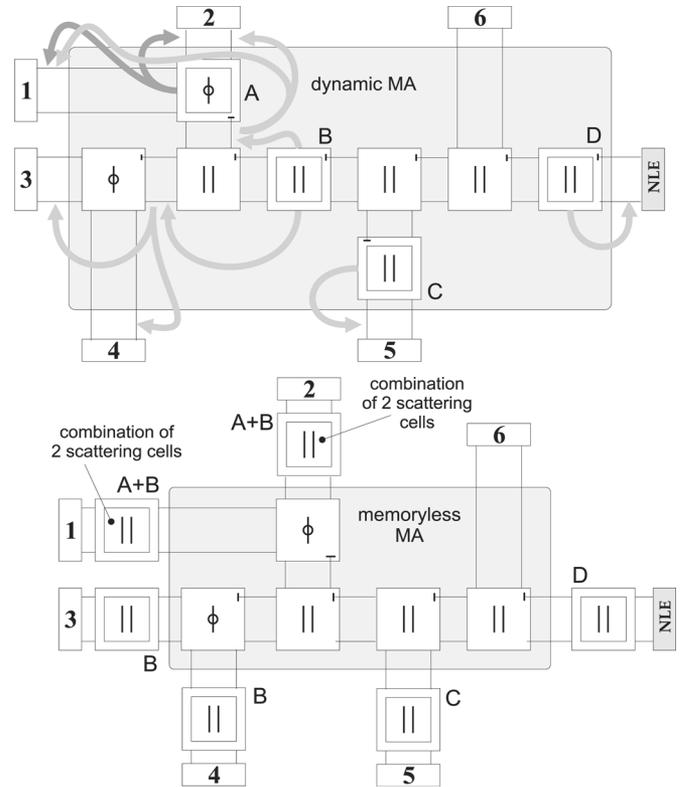


Fig. 2. Extracting the dynamics from a MA. Again, the double-box symbol is used here to distinguish between an adaptor with memory and a traditional (instantaneous) one.

1) *Structural Equivalences:* A MA can always be transformed into a new structure made of a memoryless MA surrounded by DSCs as shown in Fig. 2. This can be achieved by having all the DSCs “slide through” the inner adaptors according to specific rules, until they reach the periphery of the MA. In order to characterize these “sliding rules,” we need to find the equivalence that exists between a three-port memoryless adaptor that has one port connected to a DSC and another three-port memoryless adaptor of the same type that has two DSCs connected to the other two ports. We approached this problem in [18] and [19], where we proved that this can be done in a rather straightforward fashion. In fact, we found that a structure made of a memoryless adaptor and a DSC connected to the junction’s adapted port is equivalent to the same adaptor whose ports that are nonadapted are connected to DSCs of the same type. The earlier equivalency rule holds true also in the opposite direction. If, however, only one of the two nonadapted ports is connected to a scattering cell, we can connect the other nonadapted port to the cascade of the same DSC with another having a port RTF of opposite sign. This results from the fact that two DSCs with opposite port RTFs (and same initial conditions) cancel each other out. The pair of DSCs that are connected to the nonadapted ports of the junction can now be moved to the adapted port. The outcome is similar to before: a structure made of a memoryless adaptor and a DSC connected to any of its ports is equivalent to the same adaptor whose other two ports are connected to similar DSCs.

2) *Initial Conditions:* When we replace a structure that includes a DSC with an equivalent one having two DSCs, such

elements with memory cannot be independent on one another as this would correspond to increasing the number of state variables (and of initial conditions). Consequently, the initialization of both DSCs will depend on that of the original one.

Intuitively, one would expect that the original initial condition would be divided among the two new scattering cells. In fact, as we showed in [18] and [19], the two new initial conditions are a weighted version of the original one, the weights being the reflection coefficient at each port. This is plausible, as the sum of these two coefficients is  $\gamma_1 + \gamma_2 = 1$ .

### III. WT AND STATE-UPDATE EQUATIONS

Solving a circuit in the Kirchhoff domain means determining the values of the voltage/current pairs for each of its ports. The classical tableau analysis [21] for analog linear circuits is based on the construction of a linear system in the  $2N$  unknowns. The equations needed to do so are divided into two groups: 1)  $N$  local equations (*I/O* relationships) that characterize the individual bipoles and 2)  $N$  global equations derived from the laws of continuity (Kirchhoff laws), which describe the interconnection topology.

In this section, we show how this approach can be readily modified in order to accommodate WD structures [11], bringing it to a level of practical usability.

We will begin by constructing a wave tableau (WT) in quite a straightforward fashion, i.e., using the definition of scattering parameters and incorporating all the WD elements, including the adaptors. Treating the adaptors as circuit elements means transforming the global relationships (the topology) into local relationships. In so doing, the second part of the system is replaced by simple port-connection rules to form the new global relationships. This choice, albeit increasing the size of the system, simplifies the writing process, and more importantly, it represents the key step for the achievement of the results presented in this paper. In fact, in the reduced WT discussed later, the state-update equations of the circuit elements are extracted from the WT, which is then formed only by the MA equations. This separation between local and global relationships allows us to fully benefit from the advantages that are inherent in the WD solutions. Although the reduced tableau has never been used in practice, it is a fundamental intermediate step from which the state-update equation of the MA and the binary connection tree (BCT) has been devised.

In the following sections, we will assume, without loss of generality, that all junctions are three-port adaptors. In fact, it is quite easy to see that any  $N$ -port adaptor can always be thought of as the interconnection of  $N - 2$  three-port adaptors.

#### A. WT

Let us collect all the scattering equations of all the WD elements, including the adaptors, organizing them in a system. With this approach, the number of unknowns is exactly twice the number  $P = 4N - 6$  of ports,  $N$  being the number of bipoles. We thus need  $2P$  equations, half of which are provided by local equations (those of the structure elements), and the other half come from global equations (those that describe how the elements are interconnected).

TABLE I  
SCATTERING MATRICES OF COMMONLY USED COMPONENTS

ELEMENT	BLOCK IN $M_0$	BLOCK IN $M_1$	$\mathbf{u}$
Resistor	0	0	0
Voltage Source	0	0	$V_g$
Current Source	0	0	$R_g I_g$
Capacitor	0	1	0
Inductor	0	-1	0
Parallel Adaptor	$\begin{bmatrix} 1-\delta_i & \delta_{i-1} & \delta_{i-1} \\ -\delta_i & \delta_i & -1 \\ -\delta_i & \delta_{i-1} & 0 \end{bmatrix}$	0	0
Series Adaptor	$\begin{bmatrix} \gamma_i-1 & \gamma_i & \gamma_i \\ 1-\gamma_i & -\gamma_i & 1-\gamma_i \\ -1 & -1 & 0 \end{bmatrix}$	0	0
R-C Mutator	$\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$	0

The local relationships for WD descriptions can be written in matrix form as a set of scattering equations  $\mathbf{b} = \mathbf{M}_0 \mathbf{a} + \mathbf{u}$ , which can be rewritten as

$$[\mathbf{I} \quad -\mathbf{M}_0] \begin{bmatrix} \mathbf{b} \\ \mathbf{a} \end{bmatrix} = \mathbf{u}$$

where  $\mathbf{a}$  and  $\mathbf{b}$  are the vectors of the incident and reflected waves, respectively,  $\mathbf{u}$  is the vector of inputs (generators), and  $\mathbf{M}_0$  is the scattering matrix of all the elements of the structure. The dimension of such vectors equals the number of ports.

If the circuit contains dynamic elements, then  $\mathbf{b}$  depends also on the past values of  $\mathbf{a}$  and  $\mathbf{b}$

$$\mathbf{b}(n) = \mathbf{M}_0 \mathbf{a}(n) + \mathbf{u}(n) + \sum_{k=1}^{K_a} \mathbf{M}_k \mathbf{a}(n-k) + \sum_{k=1}^{K_b} \mathbf{M}'_k \mathbf{b}(n-k)$$

which, in matrix form, becomes

$$[\mathbf{I} \quad -\mathbf{M}_0] \begin{bmatrix} \mathbf{b}(n) \\ \mathbf{a}(n) \end{bmatrix} = \mathbf{u}(n) + \sum_{k=1}^K [\mathbf{M}'_k \quad \mathbf{M}_k] \begin{bmatrix} \mathbf{b}(n-k) \\ \mathbf{a}(n-k) \end{bmatrix}$$

where the matrix  $[\mathbf{M}'_k \quad \mathbf{M}_k]$  expresses this dependence. As we can see from Table I, elementary blocks have  $K = \max(K_a, K_b)$  that is seldom greater than one. More complex blocks obtained connecting simpler ones exhibit larger summations.

Global equations are specified by a pair of equations for each port interconnection. For example, if port  $i$  is connected to port  $j$ , then we have  $a_i = b_j$  and  $a_j = b_i$ . In a structure with  $P$  ports, we have a total of  $P$  such equations (two equations per port interconnection). Once we have scattering and interconnection relationships of all the elements, we can assemble the whole system of  $2P$  equations in  $2P$  unknowns

$$\mathbf{T}_0 \begin{bmatrix} \mathbf{b}(n) \\ \mathbf{a}(n) \end{bmatrix} = \begin{bmatrix} \mathbf{u}(n) \\ \mathbf{0} \end{bmatrix} + \sum_{k=1}^K \mathbf{T}_k \begin{bmatrix} \mathbf{b}(n-k) \\ \mathbf{a}(n-k) \end{bmatrix}$$

where  $\mathbf{T}_0$  is the memoryless portion of the tableau matrix, which is made of four  $P \times P$  blocks

$$\mathbf{T}_0 = \begin{bmatrix} \mathbf{I}_P & -\mathbf{M}_0 \\ \mathbf{C} & \mathbf{I}_P \end{bmatrix}$$

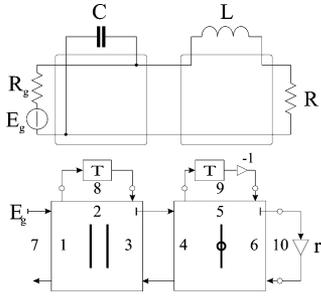


Fig. 3. Example of (top) reference analog circuit and (bottom) corresponding WD structure.

where  $\mathbf{M}_0$  is the block-diagonal matrix containing the instantaneous portions of the RTFs, which are the I/O relationships of the circuit elements;  $\mathbf{C}$  is the interconnection matrix; and  $\mathbf{I}_P$  is the order- $P$  identity matrix. The matrix  $\mathbf{T}_k$  represents the component of the tableau matrix that acts on those input and output samples that are delayed  $k$  time steps. In practice, the matrices  $\mathbf{T}_k$  describe the dynamics (i.e., the *history*) of the network elements

$$\mathbf{T}_k = \begin{bmatrix} \mathbf{M}'_k & \mathbf{M}_k \\ \mathbf{0}_P & \mathbf{0}_P \end{bmatrix}.$$

Finally,  $\mathbf{u}(n)$  is the  $P$ -dimensional vector of known terms representing, for example, the generator's values at time  $n$ .

In order to better clarify what are described earlier, let us consider the simple circuit shown in Fig. 3, which is made of  $N = 4$  linear bipoles (the generator and its resistance are treated as a single bipole) and needs  $L = 2$  adaptors to be correctly assembled. We thus have a total of  $P = 4N - 6 = N + 3L = 10$  ports.

As the circuit exhibits order-one dynamic elements (capacitor and inductor), we have to build not just  $T_0$  but also  $T_1$ . The first  $P$  rows of  $T_0$  represent the instantaneous portion of the RTFs of all circuit elements. With reference to the port numbering of Fig. 3, the matrix  $\mathbf{M}_0$  turns out to be a  $10 \times 10$  block-diagonal matrix of the form

$$\mathbf{M}_0 = \text{block}(\mathbf{P}_0, \mathbf{S}_0, 0, 0, 0, -r)$$

where [17]

$$\mathbf{P}_0 = \begin{bmatrix} 1 - \delta_1 & \delta_1 - 1 & -1 \\ -\delta_1 & \delta_1 & -1 \\ -\delta_1 & \delta_1 - 1 & 0 \end{bmatrix}$$

$$\mathbf{S}_0 = \begin{bmatrix} \gamma_1 - 1 & \gamma_1 & \gamma_1 \\ 1 - \gamma_1 & -\gamma_1 & 1 - \gamma_1 \\ 1 & 1 & 0 \end{bmatrix}$$

are the scattering matrices corresponding to the series adaptor (of reflection coefficient  $\delta_1$ ) and of the parallel adaptor (of reflection coefficient  $\gamma_1$ ), respectively. Notice that  $r$  is the reflection coefficient of the resistance  $R$ , which becomes zero only when the reference resistance of port six is equal to  $R$ . We also have

$$\mathbf{u}(n) = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ E_g \ 0 \ 0 \ 0]^T.$$

The matrix  $\mathbf{T}_1$  is zero everywhere except for those two diagonal elements of  $\mathbf{M}_1$  that correspond to the inductor and to the capacitor, which are the only dynamic elements of the circuit

$$\mathbf{T}_1 = \begin{bmatrix} \mathbf{M}'_1 & \mathbf{M}_1 \\ \mathbf{0}_P & \mathbf{0}_P \end{bmatrix}$$

where  $\mathbf{M}'_1 = \mathbf{0}_P$  and

$$\mathbf{M}_1 = \text{block}(\mathbf{0}_3, \mathbf{0}_3, 0, 1, -1, 0).$$

Finally, the  $P$  interconnection equations are

$$\begin{aligned} b_7 - a_1 &= 0 & b_8 - a_2 &= 0 & b_4 - a_3 &= 0 \\ b_3 - a_4 &= 0 & b_9 - a_5 &= 0 & b_{10} - a_6 &= 0 \\ b_1 - a_7 &= 0 & b_2 - a_8 &= 0 & b_5 - a_9 &= 0 \\ b_6 - a_{10} &= 0 \end{aligned}$$

which can be written in matrix form as

$$[\mathbf{C} \ \mathbf{I}_P] \begin{bmatrix} \mathbf{b} \\ \mathbf{a} \end{bmatrix} = \mathbf{0}$$

where

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Notice that the symmetry of  $\mathbf{C}$  is due to the fact that each connection is described by equations that involve two pairs of waves with swapped indexes. At this point, the system is completely specified and can be solved to compute the circuit wave pairs, provided that it is properly initialized, i.e., that the vector  $[\mathbf{b}^T \ \mathbf{a}^T]^T$  at time zero is determined.

With minor changes in the earlier approach, it is possible to solve circuits containing one nonlinearity, as long as the NLE is explicitly described in the WD domain as  $b = f(a)$  and is connected to the adapted port available in the MA (in the earlier example, port six is the adapted one). This can be done by replacing the product between the coefficient of  $\mathbf{T}_0$  in position  $(P, 2P)$  by the incident wave  $\mathbf{a}_{2P}$  at the port  $2P$  with the term  $f(\mathbf{a}(n))$ . In order to preserve the linearity of the system, this terms need to be moved onto the other side of the equation and treated as a previously computed variable. Indeed, this is possible by exploiting the fact that the NLE is connected to the MA's adapted port. The modified system thus becomes

$$\mathbf{T}_0 \begin{bmatrix} \mathbf{b}(n) \\ \mathbf{a}(n) \end{bmatrix} = \begin{bmatrix} \mathbf{u}(n) \\ \mathbf{0} \end{bmatrix} + \sum_{k=1}^K \mathbf{T}_k \begin{bmatrix} \mathbf{b}(n-k) \\ \mathbf{a}(n-k) \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{P-1} \\ f(\mathbf{a}_{2P}(n)) \\ \mathbf{0}_P \end{bmatrix}.$$

If  $R$  in the previous example were nonlinear, we could express the corresponding reflection coefficient  $r$  as a function of  $a$ , and

we could replace  $r$  with  $f(a_P)$  or the nonlinear function mapped to the wave domain. Moving this nonlinear function on the other side of the tableau equation (removing the corresponding intrinsic description from the tableau matrix) and treating it as a known variable means using the wave reflected by the NLE like an input to the adapted port.

By exploiting the instantaneous adaptation property [14], we can extend this idea to all the circuit bipoles. This leads to a second formulation of the WT method in which the matrix of the coefficients describes only the cluster of all the circuit adaptors. In fact, all interconnections in WD systems are done in such a way as to avoid instantaneous (noncomputable) loops; therefore, the adaptation condition (no instantaneous reflection) is satisfied either by the bipole or by the port that it connects to. It is not difficult to realize that the set of all adaptors form an energy-transparent multiport adaptor, which in the following will be referred to as MA. The second formulation of WT method thus consists of splitting the state-update process in two phases: in the first one, we compute the waves reflected by the MA (using the tableau system driven by the waves produced by the bipoles), and in the second one, we compute the waves reflected by the bipoles when driven by the waves coming from the MA.

The new tableau system becomes

$$\begin{bmatrix} \mathbf{I}_{3L} & -\hat{\mathbf{M}}_0 \\ \hat{\mathbf{C}} & \mathbf{I}_{3L} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}}(n) \\ \hat{\mathbf{a}}(n) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3L} \\ \hat{\mathbf{u}}(n) \end{bmatrix} + \sum_{k=1}^K \hat{\mathbf{T}}_k \begin{bmatrix} \hat{\mathbf{b}}(n-k) \\ \hat{\mathbf{a}}(n-k) \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{6L-1} \\ f(a_{3L}(n)) \end{bmatrix}$$

where a hat has been put on every vector and matrix to stress the fact that they still have the same meaning but different dimensions with respect to their previously defined counterpart.

In particular,  $\hat{\mathbf{M}}_0$  is now a block-diagonal matrix composed of  $3 \times 3$  blocks only (the system has only three-port adaptors);  $\hat{\mathbf{C}}$  describes just the  $3L-N$  interconnections between adaptors; and  $\hat{\mathbf{u}}(n)$  is the vector of the  $3L$  inputs. The input vector has a total of  $N$  nonzero elements that are placed in the second half of the vector of “known variables,” which allows us to distinguish the incident waves that are coming from the bipoles. As the MA can be assumed as nondynamic (we can extract any memory from it as seen in Section II.B), all the  $\hat{\mathbf{T}}_k$  are null, therefore

$$\begin{bmatrix} \mathbf{I}_{3L} & -\hat{\mathbf{M}}_0 \\ \hat{\mathbf{C}} & \mathbf{I}_{3L} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}}(n) \\ \hat{\mathbf{a}}(n) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3L} \\ \hat{\mathbf{u}}(n) \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{6L-1} \\ f(a_{3L}(n)) \end{bmatrix}.$$

In conclusion, the tableau system of the MA is made of the following components.

- 1)  $3L$  intrinsic I/O relationships (which can be expressed as  $L$  matrix equations of the form  $\mathbf{b} = \mathbf{M}\mathbf{a}$ ).
- 2)  $N$  interconnection equations between MA and bipoles ( $a_i = u_i$ ).
- 3)  $3L-N$  interconnection equations between adaptors ( $-b_i + a_k = 0$ ).

The terms on the right-hand side of the earlier equation, which are known at time  $n$ , may be grouped together into a single vector  $\hat{\mathbf{t}}(n)$

$$\begin{bmatrix} \mathbf{I}_{3L} & -\hat{\mathbf{M}}_0 \\ \hat{\mathbf{C}} & \mathbf{I}_{3L} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}}(n) \\ \hat{\mathbf{a}}(n) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{t}}(n) \end{bmatrix}. \quad (2)$$

Let us consider, once again, the example proposed before. Using this modified tableau description,  $\hat{\mathbf{M}}_0$  and  $\hat{\mathbf{C}}$  become  $6 \times 6$  matrices, while the input vector ends up containing just one nonzero element, corresponding to the voltage generator

$$\begin{aligned} \hat{\mathbf{M}}_0 &= \text{block}(\mathbf{P}_0, \mathbf{S}_0) \\ \hat{\mathbf{C}} &= \text{block}\left(\mathbf{0}_2, \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}, \mathbf{0}_2\right) \\ \hat{\mathbf{u}} &= [E_g \ 0 \ 0 \ 0 \ 0 \ 0]^T. \end{aligned}$$

### B. State-Update Equations in the WD Domain

The tableau system that describes the MA, shown in (2), emphasizes the strict relationship between the input vector  $\hat{\mathbf{t}}(n)$  and the vector of incident waves  $\hat{\mathbf{a}}(n)$ . In fact, eliminating from  $\hat{\mathbf{a}}$  the elements that do not correspond to the inputs, we are left with  $\hat{\mathbf{t}}$ . In other words, if we eliminate the equations that correspond to internal interconnections in the MA, it is possible to write  $\hat{\mathbf{b}}$  as a function of  $\hat{\mathbf{t}}$ . The elimination of the nonnecessary equations leads to the scattering equation of the MA, whose dimension is  $N \times N$ , just like in the analog case.

At this point, we can decompose the system in the two groups of local and global equations

$$\begin{aligned} \hat{\mathbf{b}}(n) - \hat{\mathbf{M}}_0 \hat{\mathbf{a}}(n) &= \mathbf{0} \\ \hat{\mathbf{C}} \hat{\mathbf{b}}(n) + \hat{\mathbf{a}}(n) &= \hat{\mathbf{t}}(n) \end{aligned}$$

which can be combined together to eliminate  $\hat{\mathbf{a}}(n)$

$$(\mathbf{I} + \hat{\mathbf{M}}_0 \hat{\mathbf{C}}) \hat{\mathbf{b}}(n) = \hat{\mathbf{M}}_0 \hat{\mathbf{t}}(n).$$

If the matrix  $\mathbf{I} + \hat{\mathbf{M}}_0 \hat{\mathbf{C}}$  is nonsingular, then we have

$$\hat{\mathbf{b}}(n) = (\mathbf{I} + \hat{\mathbf{M}}_0 \hat{\mathbf{C}})^{-1} \hat{\mathbf{M}}_0 \hat{\mathbf{t}}(n) = \overline{\mathbf{M}} \hat{\mathbf{t}}(n). \quad (3)$$

This way, we obtain a state-update equation in explicit form, where  $\hat{\mathbf{t}}(n)$  replaces the role of  $\hat{\mathbf{a}}(n)$ .

Notice that  $\overline{\mathbf{M}}$  is a  $3L \times 3L$  matrix, while what we need to find is an  $N \times N$  scattering matrix  $\mathbf{M}$  for the MA. In fact, the vector  $\hat{\mathbf{b}}(n)$  obtained from (3) contains also the waves reflected by the internal ports of the MA. The scattering matrix  $\mathbf{M}$  can be obtained by simply eliminating the equations relative to such internal ports. We can do so by using an appropriate  $N \times 3L$  matrix  $\mathbf{Q}$  obtained by eliminating rows from the  $3L \times 3L$  identity matrix

$$\mathbf{M} = \mathbf{Q}[(\mathbf{I} + \hat{\mathbf{M}}_0 \hat{\mathbf{C}})^{-1} \hat{\mathbf{M}}_0] \mathbf{Q}^T = \mathbf{Q} \overline{\mathbf{M}} \mathbf{Q}^T.$$

The premultiplication by  $\mathbf{Q}$  eliminates the unnecessary rows while the postmultiplication by  $\mathbf{Q}^T$  eliminates the corresponding columns, in order to obtain again a square matrix. If we did not perform this column removal, the elements of such columns would multiply those elements of  $\hat{\mathbf{t}}(n)$  that are zero anyway. Such elements correspond to the incident waves at the internal interconnection ports. Indeed, the input vector will have to be resized accordingly, by eliminating all the zero elements  $\mathbf{a}(n) = \mathbf{Q} \hat{\mathbf{t}}(n)$ , which results in the state-update equation of the form  $\mathbf{b}(n) = \mathbf{M} \mathbf{a}(n)$ , where  $\mathbf{b}(n)$  and  $\mathbf{a}(n)$  are now made of  $N$  elements.

In the example of the previous section, we have a  $4 \times 6$  matrix  $\mathbf{Q}$ , which is derived by eliminating rows 3 and 4 from the order-six identity matrix

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

### C. Initialization Issues

As the circuit to be simulated will, in general, contain dynamic elements, it must be properly initialized. Following, once again, the classical network-theory approach, the initialization of a WT implementation of a WD structure can be addressed in a matrix-oriented fashion. This is done [21] by replacing the relationships of the dynamic elements with ideal generators that force their given initial conditions. We will see that this can be accomplished directly in the matrix representation in a fully automated fashion.

Let us consider again the WT system for the linear case

$$\begin{bmatrix} \mathbf{b}(n) \\ \mathbf{a}(n) \end{bmatrix} = \mathbf{T}_0^{-1} \begin{bmatrix} \mathbf{u}(n) \\ \mathbf{0} \end{bmatrix} + \mathbf{T}_0^{-1} \sum_{k=1}^K \mathbf{T}_k \begin{bmatrix} \mathbf{b}(n-k) \\ \mathbf{a}(n-k) \end{bmatrix} \quad (5)$$

here, we notice that the solution is made of two contributions: 1) the solution due just to the inputs  $\mathbf{u}(n)$  or *forced evolution* and 2) the solution that we have when the input is zero,  $\mathbf{u}(n) = \mathbf{0}$ , or *free evolution*, which depends on the values of the solution vector in the  $K$  previous time steps.

At time  $n$ , the vector  $\mathbf{u}$  is known and the solution can be computed as long as we know the values of the vectors  $[\mathbf{b}^T(n-k)|\mathbf{a}^T(n-k)]^T$ , for  $k = 1, \dots, K$ . Thus, initializing the system means determining such  $K$  vectors at time zero, starting from the  $K$  variables that we want to set on the circuit. This can be done by writing a system of  $K$  linear equations obtained by replacing some of the equations in the WT system in such a way as to account for both the initial conditions and the properties of the circuit. In practice, not all WT equations will depend on the history of the circuit but only those in which the corresponding row of at least one of the matrices  $\mathbf{T}_k$  contains nonzero elements. We will thus discriminate between *instantaneous relationships*, whose elements in the matrices  $\mathbf{T}_k$  are all zero, and *delayed relationships* (with delay  $i \leq K$ ), which are those that depend on the first  $i$  matrices  $\mathbf{T}_k$ . All the delayed relationships have no meaning as they depend on previous wave samples that have not been computed. Therefore, they need to be replaced by the instantaneous relationships which describe the ideal generators that force the initial conditions. For those circuits whose dynamic elements are just capacitors and inductors ( $K = 1$ ), we only need to compute the vector  $[\mathbf{b}^T(0)|\mathbf{a}^T(0)]^T$  with one of the two possible instantaneous relationships given by the initial conditions

$$\frac{a_m(0) + b_m(0)}{2} = v_m(0)$$

$$\frac{a_m(0) - b_m(0)}{2R_m} = i_m(0)$$

where  $v_m$  and  $i_m$  are the voltage and the current at the port  $m$ , respectively, whose reference resistance is  $R_m$ , while  $a_m$  and  $b_m$

are the corresponding waves at that port. In order to obtain the new system matrix  $\mathbf{T}'_0$ , we just need to modify the WT matrix, replacing the rows corresponding to the delayed equations, with the new relationships

$$a_m(0) + b_m(0) = 2v_m(0)$$

$$a_m(0) - b_m(0) = 2R_m i_m(0).$$

If  $\mathbf{T}'_0$  is nonsingular, then the resulting system of equations turns out to have a unique solution  $[\mathbf{b}^T(0)|\mathbf{a}^T(0)]^T$ ; otherwise, if  $\det \mathbf{T}'_0 = 0 \neq \det \mathbf{T}_0$ , then the cause is due to a wrong choice of initial conditions, for example, trying to set all voltage values in a closed loop or trying to set all current values crossing a closed surface.

In general, when  $K \geq 1$ , we will need to compute all vectors  $[\mathbf{b}^T(n-k)|\mathbf{a}^T(n-k)]^T$  up to the order  $K$  by solving the systems

$$\mathbf{T}'_k \begin{bmatrix} \mathbf{b}(n-k) \\ \mathbf{a}(n-k) \end{bmatrix} = \begin{bmatrix} \mathbf{c}(n-k) \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{u}(n-k) \\ \mathbf{0} \end{bmatrix} + \sum_{i=1}^{k-1} \mathbf{T}_i \begin{bmatrix} \mathbf{b}(n-i) \\ \mathbf{a}(n-i) \end{bmatrix}$$

under the initial conditions vector  $\mathbf{c}(n-k)$ , which will contain the terms  $v_m(n-k)$ ,  $R_m i_m(n-k)$ , or any other suitable initial condition.

If the circuit contains an NLE, then this must be treated as an input

$$\mathbf{T}'_k \begin{bmatrix} \mathbf{b}(n-k) \\ \mathbf{a}(n-k) \end{bmatrix} = \begin{bmatrix} \mathbf{c}(n-k) \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{u}(n-k) \\ \mathbf{0} \end{bmatrix} + \sum_{i=1}^k \mathbf{T}_i \begin{bmatrix} \mathbf{b}(n-i) \\ \mathbf{a}(n-i) \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{P-1} \\ f(a_P(n-k)) \\ \mathbf{0}_P \end{bmatrix}.$$

We will thus need to precompute the value of  $a_P(n-k)$  in order to be able to compute  $f(a_P(n-k))$  and include it among the known variables of the system.

In conclusion, the setting of the initial conditions in the WT method can be automated, and it is always based on the solution of a linear set of equations and, if needed, the evaluation of a nonlinear characteristics.

## IV. BCT

The approach presented in the previous section is based on the automatic derivation of the state-update equations that describe the MA. In this section, we propose an alternative solution that allows us to readily implement a WD structure through the inspection (scanning) of the treelike topological representation of the reference model (connection tree). We will assume, once again, that the structure is made just of three-port adaptors; therefore, the connection tree turns out to be binary (hence, the name BCT).

### A. WD Structure Inspection

A very simple WD structure that we can use to illustrate the method of topological inspection of this section is made of a chain of  $L$  adaptors, each connected to the adapted port of the previous one, except for adaptors at the extremes of the chain,

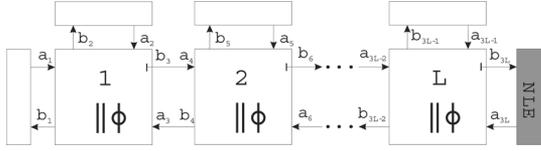


Fig. 4. Chain structure. Each adaptor is connected to at least one bipole, and there is no branching.

which are connected to two bipoles. The adapted port, in particular, is connected to an NLE, as shown in Fig. 4. Understanding how to implement this circuit through direct inspection will help us show how to proceed in more general cases.

Our goal is to devise a method to compute the vector  $\mathbf{b}$  from the known elements of  $\mathbf{a}$ . One should keep in mind, in fact, that the only elements of  $\mathbf{a}$  that are known are those that correspond to the ports connected to linear bipoles. In fact, we have the following expressions.

- 1)  $a_i(n) = 0$ , when the bipole is an adapted resistor.
- 2)  $a_i(n) = V_i$ , when the bipole is an adapted real generator.
- 3)  $a_i(n) = \pm b_i(n-1)$ , if the bipole is a reactance.

The steps for computing  $\mathbf{b}$  from the known elements of  $\mathbf{a}$  can be as follows.

- 1) Once the circuit is initialized, the vector of the incident waves will be

$$\mathbf{a} = [a_1, a_2, x, x, a_5, x, \dots, x, a_{3L-1}, x]^T$$

where  $x$  denotes an unknown, while  $a_i$  represents a known variable.

- 2) We start from the first adaptor, which is the only one in which both the inputs of the nonadapted ports ( $a_1$  and  $a_2$ ) are known. The output of the adapted port can be readily computed as  $b_3 = f(a_1, a_2) = k_{11}a_1 + k_{12}a_2$ ,  $k_{11}$ , and  $k_{12}$  being appropriate transmission coefficients that depend on the reference resistances of the other two ports of the adaptor. Notice that  $b_3$  does not depend on  $a_3$  because port three is adapted.
- 3) Port three is connected to port four; therefore, now, we have both  $a_5$  (from the bipole) and  $a_4 = b_3$ , computed in the previous step.
- 4) We can now repeat steps 1) and 2) until we reach the last adaptor. This way, we are able to compute some of the elements of  $\mathbf{a}$  and  $\mathbf{b}$

$$\mathbf{a} = [a_1, a_2, x, a_4, a_5, x, \dots, a_{3L-2}, a_{3L-1}, x]^T$$

$$\mathbf{b} = [x, x, b_3, x, x, b_6, \dots, x, x, b_{3L}]^T.$$

- 5) Using the characteristics of the NLE, we can compute  $a_{3L} = f_{NL}(b_{3L})$ .
- 6) We can now compute, if necessary,  $b_{3L-1}$  and/or  $b_{3L-2}$  and go through the whole circuit computing  $b_{3i-1}$  and  $b_{3i-2}$  until we arrive at the first adaptor. Notice that not all the reflected waves  $b_{3i-k}$  ( $k = 1, 2$ ) need to be computed. For example, we do not need to compute the reflection at those (nonadapted) ports that are connected to a resistor (no reflection).
- 7) Once  $\mathbf{b}$  is specified, it is possible to update  $\mathbf{a}$  using the I/O descriptions of the bipoles. We can thus go back to the beginning of the procedure to compute the next value of  $\mathbf{b}$ .

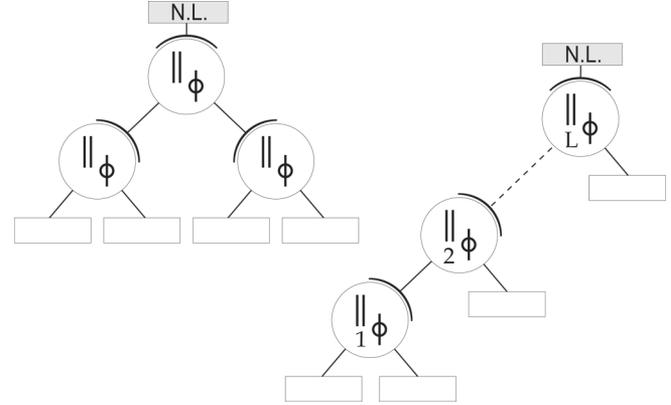


Fig. 5. Two examples of BCTs. (left) Generic one and (right) chainlike circuit. The circular box represents an instantaneous adaptor, in which the adapted port is clearly specified. This particular notational choice simplifies the drawing of connection trees with a great amount of branching.

## B. BCT

The chainlike circuit described earlier can be generalized by allowing the other two nonadapted ports to be connected to adapted ports of other three-port junctions. We can quite easily see that the topology remains treelike and that using only three-port adaptors allows us to use a binary tree.

In general, we describe the interconnection topology of a circuit using the so-called BCT, which is constructed by following some simple rules.

- 1) The **root** of the binary tree corresponds to the adaptor that the NLE connects to.
- 2) The **nodes** of the tree are three-port standard WDF adaptors and the branching topology matches the actual adaptor's interconnection topology.
- 3) The **leaves** correspond to the linear bipoles.

The method that we described for the chainlike circuit can be readily extended to this new situation by defining a “forward scan” (from the leaves toward the NLE) and a “backward scan” (from the NLE toward the leaves). In fact, the computation starts from the leaves of the tree, which contain the “memory elements” with the initial conditions. The computation then proceeds by following a forward scan, and when the working point of the NLE is found, the backward scanning propagates the computation back toward the leaves, where all memory cells are refreshed.

As shown in Fig. 5, the adapted port of a junction is either connected to a nonadapted port of another adaptor or to an NLE.

One major advantage of implementing the algorithm as described earlier lies in the fact that its computational cost and its memory requirements increase linearly with the number of adaptors. This is a big improvement with respect to the WT method based on the implementation of the state-update equation, whose complexity grows quadratically with the number of circuit elements.

Notice that, as shown in Fig. 5 and in the following ones, we adopted a nonstandard representation of the WD adaptors. In fact, we use a circular box in which the adapted port is clearly specified. This particular notational choice simplifies the drawing of connection trees with a great deal of branching.

### C. BCT Initialization

If the circuit includes dynamic elements, such as capacitors (springs) or inductors (masses), then it is necessary to determine the values to put into the memory cells that model such elements as a function of the initial conditions to match. In order to determine the wave value that initializes a memory cell, we need the whole voltage/current pair that pertain that element. In other words, we need to solve the circuit in which all dynamic elements have been replaced by ideal generators to find the corresponding Kirchhoff variable. As ideal generators do not admit a WD-adapted representation, it is not possible to use wave variables directly; otherwise, the structure would turn out to be noncomputable. However, we can still use the tree structure that describes the circuit topology, regardless of whether we are working in the WD domain or in the Kirchhoff domain. In particular, it is quite simple to compute the voltage/current characteristics of the subtree at a generic node, given those of the two “children nodes.” In fact, as the circuit portion described by a subtree is linear, such characteristics are *straight lines* that can be modeled according to the Thevenin or Norton equivalent circuit corresponding to the subtree. Knowing the lines that model the two subtrees and the type of node that combines them (series or parallel), we can easily compute the line that describes their combination.

What we know initially are the characteristics of all bipoles (we replaced capacitors and inductors with ideal voltage or current generators, respectively). As we go through the tree to reach the root, we compute and store the characteristics at each node. Once we know the characteristics at the root (which is the one of the whole tree), we can think of the circuit as decomposed into its linear part (represented by the equivalent circuit we just computed) and its nonlinear bipole. The characteristic line that describes the linear portion of the circuit intersects the voltage–current characteristics of the nonlinear bipole in its “workpoint” (see Fig. 6). The search for this point of intersection can be done with a classical numerical algorithm such as Newton–Raphson’s.

Now that we know the workpoint of the NLE (and, thus, also that of the whole linear portion of the circuit, as seen from the same port terminals), we can begin a second scan to visit all the tree nodes out as far as the leaves.

With reference to Fig. 7, if the root adaptor is a parallel one, then the value  $V_0$  of the voltage at the workpoint will be the same at all three ports. Going to the left node (port one of the adaptor), we have  $V_1 = V_0$ , while the value of  $I_1$  can be computed using the equivalent of the left subtree scanned in the forward pass. On the right branch, we also have  $V_2 = V_0$ , but  $I_2$  can now be computed by simply applying Kirchhoff’s current law  $I_2 = I_0 - I_1$ . If the root adaptor is a series one, then the procedure will be the same as long as we swap the roles of currents and voltages. This time, in fact, the variable that does not change is the current, while the voltage is computed from the characteristics of the subtree corresponding to the first branch using Kirchhoff’s voltage  $V_2 = V_0 - V_1$ .

There are cases in which it is not possible to determine voltages and currents using the equivalents. This happens, for example, when trying to compute the current from an ideal voltage

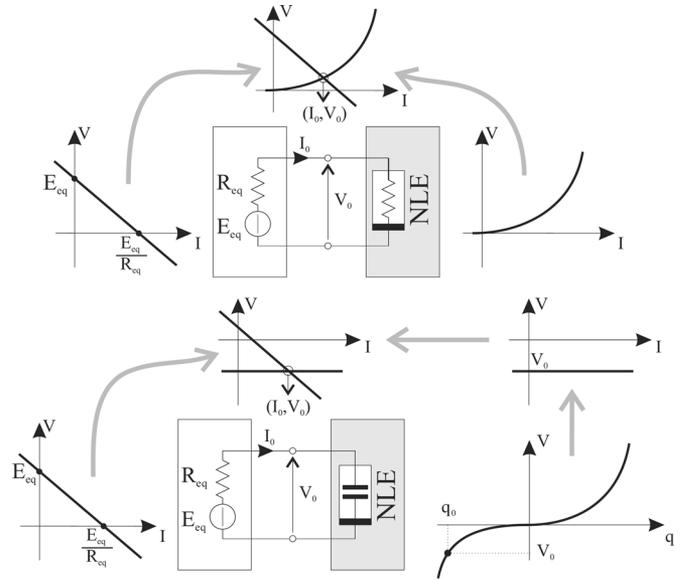


Fig. 6. Computing the workpoint  $(V_0, I_0)$  of the NLE as the intersection of its characteristics with that of the equivalent circuit. If the NLE is not instantaneous (see bottom figure), we must refer to its equivalent circuit at the initialization time.

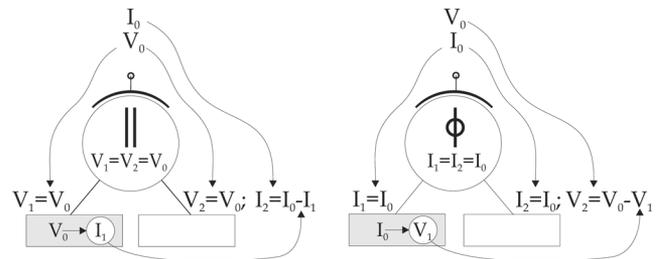


Fig. 7. Determination of the initial values during the backward scan.

generator. In order to avoid this problem, we can always explore the second branch and, then, derive the unknown value of the first branch using Kirchhoff’s laws. Notice that it will not be possible to have undetermined cases on both subtrees; otherwise, the circuit would exhibit degeneracies such as loops of voltage generators and capacitors or nodes (cut-sets) of current generators and inductors.

At this point, we have all voltages and currents at the second-level nodes of the tree. If these nodes correspond to dynamic bipoles, then we can compute the value of the wave that initializes the memory cell. If, on the other hand, we have an instantaneous bipole, then the algorithm stops at this branch; otherwise, if there is another adaptor, then we can repeat the earlier-described procedure using the newly computed currents and voltages and scanning the tree until we reach a bipole.

In conclusion, also the initialization process can be split into two phases: a forward scan (from leaves to root) and a backward scan (from root to leaves). In the first phase, we derive the characteristic lines that describe the relationship between current and voltage at each node. In this way, during the backward scan, knowing one of the two variables, we can compute the other one using these characteristics.

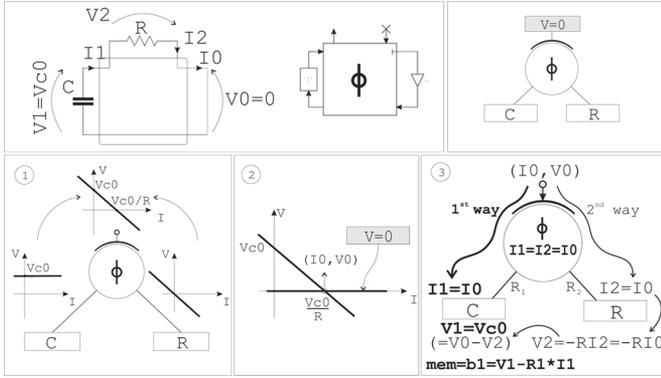


Fig. 8. Implementation of the RC circuit using a series adaptor.

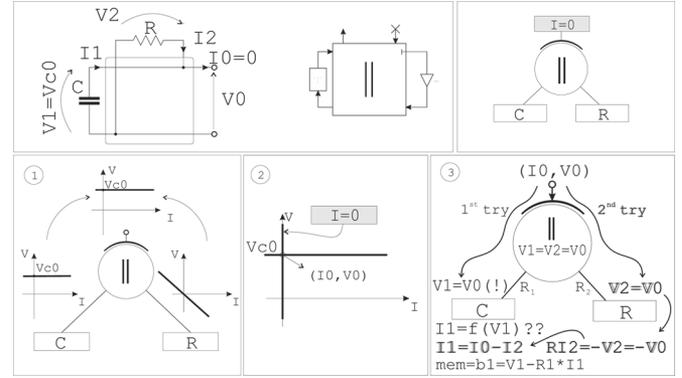


Fig. 9. Implementation of the RC circuit using a parallel adaptor.

*Example: RC circuit*—The initialization of an RC circuit consists of determining the current, given the initial condition  $V_C(0) = V_{C0}$ . This allows us to determine the reflected wave at the capacitor’s port  $b = V_{C0} - R_1 I_{C0}$ , which will then be assigned to the memory cell.

There are several ways to implement this circuit in the WD domain. One of these, shown in Fig. 8, uses a series adaptor whose adapted port is connected to a short-circuit, i.e., it is connected to an ideal generator of voltage  $V = 0$ . The forward scanning consists of deriving the equivalent bipole as seen from the root of the tree, which is given by the series of an ideal voltage generator  $V_{C0}$  (the equivalent of the capacitor) and the resistor  $R$ . Its characteristics, using the generator’s convention, is a line crossing the voltage axis at the point  $V_{C0}$ , with slope  $-R$ . Once we find the workpoint ( $V_0 = 0, I_0 = V_{C0}/R$ ) as the intersection between the line and the characteristics of the short-circuit, we can initiate the backward scan.

We can choose, for example, to visit first the left branch (the one that goes toward the capacitor) where we have  $I_{C0} = I_2 = I_0$  because the series adaptor “preserves” the currents. This equation, together with the initial condition  $V_{C0}$ , allows us to compute the initial wave that we need. Notice that, by following this path, it was not necessary to visit the leaves corresponding to the resistor, because it does not contain any memory cell to initialize. However, we took this resistor into account during the forward scan, when we derived the characteristics of the whole tree.

We could have arrived at the same result by visiting the resistor’s branch first (see gray arrows in Fig. 8). In this case, we would have had to compute the value of  $V_2$  as a function of  $I_2 = I_0$  by using the characteristics of the resistor  $R$ , i.e.,  $V_2 = -RI_2$ . Using the other Kirchhoff law, we would have found  $V_C(0) = V_1 = V_0 - V_2 = 0 + RI_2 = 0 + RI_0 = V_{C0}$ .

The same circuit can be implemented using a parallel adaptor (see Fig. 9), whose adapted port is connected to an open circuit, i.e., to an ideal current generator of  $I = 0$ . The equivalent circuit seen at the root of the tree is an ideal voltage generator (the capacitor), because the resistor  $R$  that is in parallel with it does not affect its characteristics. Intersecting this characteristics with the line  $I = 0$  (characteristics of the open-circuit bipole) allows us to find the workpoint ( $V_0 = V_{C0}, I_0 = 0$ ). This time, the resistor  $R$  did not give us any information for the derivation of the (sub)tree’s characteristics, therefore it is not possible to

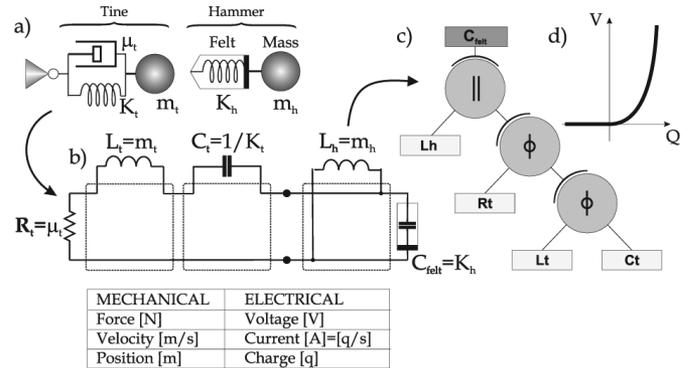


Fig. 10. Modeling of a simple hammer–tine interaction. (a) Mechanical reference model. (b) Equivalent electrical circuit. (c) BCT implementation. (d) NL characteristics of the felt (capacitor). A table with some mechanical–electrical equivalences has been added, for ease of reference.

initialize the circuit without visiting the corresponding tree leaf. As the parallel adaptor preserves the voltages, all we can write is  $V_C(0) = V_1 = V_0 = V_{C0}$  (no information from this equation); therefore, we cannot determine  $I_1$  as a function of  $V_1$  because  $V_1 = V_{C0}$  is satisfied irrespective of  $I_1$ . On the other hand, by visiting the resistor’s leaf first, we obtain

$$V_2 = V_0 = V_{C0} \quad I_2 = -V_2/R$$

$$I_1 = I_0 - I_2 = 0 - (-V_2/R) = V_{C0}/R.$$

#### D. Example of Modeling Using the BCT

We introduce here a simple example of a BCT implementation, related to the model of the hammer–tine interaction, which is typical of electromechanical pianos. Fig. 10(a) shows the mechanical equivalent of this system. The hammer is modeled as a mass attached to a felt having a nonlinear compressibility function. The sound-generating element is the tine, a steel rod tied at one end. When hit by the hammer, the tine vibrates and a pickup positioned at its free-end transforms the oscillations into an electrical signal to be amplified. The characteristic sound of this type of musical instrument is mostly to be attributed to the nonlinear response of the pickup. As this can be implemented as a postprocessing filter, its implementation is not addressed here. Even though the tine is, in fact, a distributed-parameter system,

its modeling is here simplified into a simple  $RLC$  oscillator. This simplification, however, is good enough to produce very plausible results, particularly on higher notes. Using well-known equivalences between mechanical and electrical systems, the equivalent electrical circuit can be rather straightforwardly derived. For the sake of simplicity, the two elements of the system can be modeled by a single circuit, owing to the particular choice of the nonlinear function. This, in fact, serves a twofold purpose: to model the nonlinear compressibility (reactive force versus compression) and to model the contact condition. The felt's compression can be thought of as the relative position between hammer and tine. As a result, when they are far apart, the work-point of the NL capacitor lies on the negative half-plane, where the voltage is zero (infinite capacitance), resulting in a short-circuit of the hammer side. If we apply an initial velocity (current) to the hammer's mass ( $L_m$ ), a constant current begins flowing through the NL capacitor, which holds an initial negative charge. When this charge becomes positive, the contact condition is reached, and some of the current will start flowing through the  $RLC$  circuit. Successively, when the current through  $L_m$  changes its sign, the hammer bounces away. The  $RLC$  circuit, however, continues oscillating, until all of its energy is completely dissipated by  $R_t$ , as the hammer returns in its short-circuit behavior.

The circuit shown in Fig. 10(b), where the adaptors have been emphasized, can easily be redrawn with the tree representation as shown in Fig. 10(c). This is the starting point for our software implementation, where the model to be simulated is described in terms of circuit elements connected by adaptors. In the same description, we must specify which physical quantity is associated to the output sound. In our example, this can be identified in the tine displacement from its initial position, equivalent to the integral of the current flowing in the  $RLC$  circuit or the voltage across  $C_t$ . The model description is completed by an optional set of rules that link one or more parameter (e.g., a capacitance or the voltage of a source) to an external controller, using a communication protocol such as MIDI.

Despite its simplicity (five bipoles in total), the WT implementation would require dealing with matrices of large size (1 of  $28 \times 28$  elements or 4 blocks of  $14 \times 14$ ), which shows that the computational cost would, in this case, be much higher than in the BCT implementation. A more systematic comparison between computational complexities of WT and BCT implementations will be given in the next section.

## V. COMPARISON

As we are particularly interested in applications of WD modeling to musical acoustics, a reduced computational cost is a factor of utmost importance. We will show here that the iterative nature of the BCT method allows us to cut down on the computational cost quite dramatically. In fact, the number of operations per output sample in the BCT case increases linearly with the number of circuit ports, while the implementation cost of the state-update equations in the WT method tends to increase quadratically with the number of ports. In order to confirm the linear growth of the computational cost with the

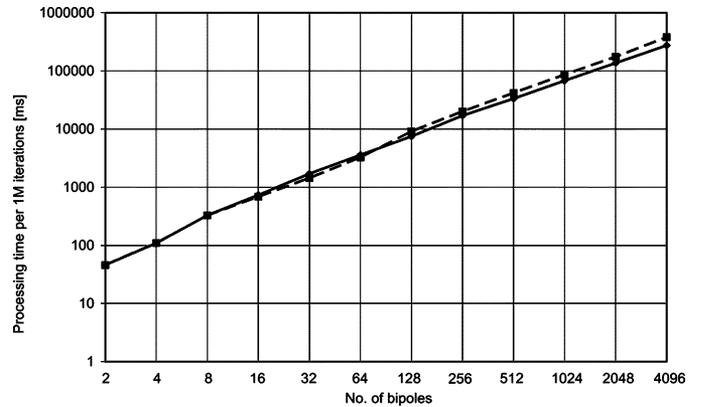


Fig. 11. Computational cost versus number of bipoles in the case of BCT implementation. The experiment consists of the construction of two circuits with a growing number of bipoles. The dashed curve refers to a WD structure that grows in a chainlike fashion (no branching), while the solid curve corresponds to a WD structure that grows in a balanced fashion (full branching).

number of circuit ports, we performed a simple test based on the construction of two circuits with a progressively increasing number of bipoles. We allowed the first structure to grow in a chainlike fashion (no branching), as opposed to a second structure, which was grown in a balanced fashion (full branching). In both cases, the computational complexity turned out to be linear (see Fig. 11). The difference in the computational time between a balanced structure and an unbalanced (chainlike) one are to be attributed to the fact that the unbalanced tree branches out about two orders of magnitude farther than a balanced one, which results in a far larger memory stack size. This explains a 3:4 ratio in the overall computational time for large structures.

In order to make the WT method more efficient, we could avoid implementing the state-update equation and refer to the tableau equation in its implicit form, as explained in Section III.A. In this case, in fact, we can take advantage of the sparsity of the WT matrix. As the degree of sparsity tends to increase with the size of the tableau matrix, a careful implementation can avoid a less than quadratic trend. The price to pay for this choice is a significant loss of flexibility in the implementation, as it would turn out to be quite difficult to have the structure interact with other physical models, which is a common requirement in musical acoustics. Notice that the explicit formulation based on the state-update equation cannot benefit from similar properties, as the state-update matrix is no longer sparse.

As far as the other properties are concerned, each one of the two described methods exhibits its own peculiarities. For example, the WT matrix can be constructed in a blockwise fashion by “pasting” WT matrices of standard three-port adaptors into a larger WT matrix, which makes it easy to automate the process. The aforementioned reasons of flexibility would require that the implicit equations be turned into a state-update equation. On the other hand, having to perform a conversion from implicit to explicit form prevents us from performing any on-the-fly changes to the interconnection topology, with consequent flexibility limits. All such limitations are overcome with the BCT method. Moreover, the fact that a connection

tree is scanned “on the fly” makes it particularly suitable for modeling dynamic topologies.

One more comment concerns the generality of the methods. We should notice that the range of applicability of the WT method in the case of state-update equations is the same as that of the BCT. The first version of the implicit WT method, on the other hand, is more general, as it does not exploit the WDF properties. For this reason, it can implement structures that would turn out to be noncomputable otherwise. For example, it allows us to incorporate ideal generators, which do not admit an adapted WD counterpart.

Finally, both methods provide us with an immediate feedback on the computability of the interconnections, and their topological correctness. Such properties can be checked through a visual inspection of the WT matrix (once turned into a state-update equation, this visual inspection is not so immediate anymore), or of the BCT structure.

## VI. CONCLUSION

In this paper, we approached the problem of automatically modeling nonlinear interactions between circuit elements, with reference to physical modeling of acoustic interactions in the musical-application scenario. The method that we developed allows us to model such interaction without altering the modeling structure of the interacting blocks, without ever violating global continuity laws such as Kirchhoff’s laws and energy conservation.

One other crucial aspect that has been assessed in this paper is the automation of the modeling process. This was achieved here through the definition of a novel strategy that exploits the inherent treelike structure of a minimal circuit. The reference connection tree is used here as a modeling paradigm that allows us to achieve a number of goals, including scalability, structural scalability, and implementability of time-varying topologies.

The proposed approach has proven effective for the automatic and modular synthesis of a wide class of physical structures encountered in musical acoustics. In fact, the BCT approach that we implemented makes the construction and the implementation of the interaction topology systematic and its implementation efficient. In its current state, the implementation of the described synthesis system is able to assemble the synthesis structure from a syntactic description of its objects and their interaction topology, opening the way to a first computer-aided design approach to the construction of interactive sound environments.

One aspect that has not been addressed in this paper is the impact of the nonlinearity on energy conservation and stability preservation of the implemented structure. This problem has been widely addressed in [5], where the use of power-normalized waves has been proposed to overcome possible problems. This is an issue that we are currently investigating.

## REFERENCES

[1] A. Fettweis, “Wave digital filters: Theory and practice,” *Proc. IEEE*, vol. 74, no. 2, pp. 327–270, Feb. 1986.

[2] G. Borin, G. De Poli, and A. Sarti, C. Roads, S. T. Pope, A. Piccialli, and G. De Poli, Eds., “Musical signal processing,” in *Musical Signal Synthesis*. Lisse, The Netherlands: Swets and Zeitlinger, 1997, pp. 5–30.

[3] J. O. Smith, M. Kahrs and K. Brandenburg, Eds., “Principles of digital waveguide models of musical instruments,” in *Applications of Digital Signal Processing to Audio and Acoustics*. Norwell, MA: Kluwer, 1998, pp. 417–466.

[4] F. Pedersini, A. Sarti, and S. Tubaro, “Object-based sound synthesis for virtual environments,” *IEEE Signal Process. Mag.*, vol. 17, no. 6, pp. 37–51, Nov. 2000.

[5] S. Bilbao, J. O. Smith, J. Bensa, and R. Kronland-Martinet, “A power-normalized wave digital piano hammer,” in *Proc. Program 144th Meeting J. Acoust. Soc. Amer.*, Cancun, Mexico, Dec. 2–6, 2002.

[6] R. Rabenstein and L. Trautmann, “Digital sound synthesis of string instruments with the functional transformation method,” *Signal Process.*, vol. 83, no. 8, pp. 1673–1688, Aug. 2003.

[7] S. Petrausch and R. Rabenstein, “Interconnection of state space structures and wave digital filters,” *IEEE Trans. Circuits Systems II, Exp. Briefs*, vol. 52, no. 2, pp. 90–93, Feb. 2004.

[8] V. Välimäki, J. Pakarinen, C. Erkut, and M. Karjalainen, “Discrete-time modeling of musical instruments,” *Rep. Prog. Phys.*, vol. 69, no. 1, pp. 1–78, 2006.

[9] T. Felderhoff, “Simulation of nonlinear circuits with period doubling and chaotic behavior by wave digital principles,” *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 41, no. 7, pp. 485–491, Jul. 1994.

[10] T. Felderhoff, “A new wave description for nonlinear elements,” in *Proc. IEEE Int. Symp. Circuits Syst.*, Atlanta, GA, May 12–15, 1996, pp. 221–224.

[11] F. Pedersini, A. Sarti, and S. Tubaro, “Block-wise physical model synthesis for musical acoustics,” *Electron. Lett.*, vol. 35, no. 17, pp. 1418–1419, Aug. 1999.

[12] A. Hirschberg, J. Kergomard, and G. Weinreich, *Mechanics of Musical Instruments*. Berlin, Germany: Springer-Verlag, 1995.

[13] A. H. Benade, *Fundamentals of Musical Acoustics*. New York: Dover, 1990, Reprint edition.

[14] A. Sarti and G. De Poli, “Toward nonlinear wave digital filters,” *IEEE Trans. Signal Process.*, vol. 47, no. 6, pp. 1654–1668, Jun. 1999.

[15] K. Meerkötter and R. Scholz, “Digital simulation of nonlinear circuits by wave digital filter principles,” in *Proc. IEEE Int. Symp. Circuits Syst.*, Portland, OR, May 8–11, 1989, vol. 1, pp. 720–723.

[16] A. Sarti and G. De Poli, “Generalized adaptors with memory for nonlinear wave digital structures,” in *Proc. 8th EUSIPCO*, Trieste, Italy, Sep. 1996, vol. 3, no. 10–13, pp. 1941–1944.

[17] A. Fettweis and K. Meerkötter, “On adaptors for wave digital filters,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-23, no. 6, pp. 516–525, Dec. 1975.

[18] A. Sarti and G. De Sanctis, “Structural equivalences in wave digital systems based on dynamic scattering cells,” in *Proc. 14th EUSIPCO*, Florence, Italy, Sep. 4–8, 2006.

[19] A. Sarti and G. De Sanctis, “Memory extraction from dynamic scattering junctions in wave digital structures,” *IEEE Signal Process. Lett.*, vol. 13, no. 12, pp. 729–732, Dec. 2006.

[20] Algorithms for the Modeling of Acoustic Interactions [Online]. Available: <http://www-dsp.elet.polimi.it/alma/IST-2001-33059> ALMA Project

[21] V. Belevitch, *Classical Network Theory*. San Francisco, CA: Holden Day, 1968.

[22] D. Franken and J. Ochs, “Generation of wave digital structures for networks containing multiport elements,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 3, pp. 586–596, Mar. 2005.

[23] A. Fettweis, “Multidimensional wave digital principles: From filtering to numerical integration,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Adelaide, Australia, Apr. 19–22, 1994, vol. 6, pp. 173–181.

[24] S. Bilbao, *Wave and Scattering Methods for Numerical Simulation*. Chichester, U.K.: Wiley, 2004.

[25] S. Petrausch and R. Rabenstein, “Wave digital filters with multiple nonlinearities,” in *Proc. 12th EUSIPCO*, Vienna, Austria, Sep. 10, 2004.



**Augusto Sarti** (M'04) was born in 1963. . He received the “Laurea” degree (*cum laude*) and the Ph.D. degree in electrical engineering with research on nonlinear communication systems from the University of Padua, Padova, Italy, in 1998 and 1993, respectively. His graduate studies included a joint graduate program with the University of California, Berkeley, where he worked on nonlinear system theory.

Since 1993, he has been with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy, where he is currently an Associate Professor and where he cofounded the Image and Sound Processing Group, which has contributed to numerous national and international research projects. His research interests are in the area of digital signal processing, with particular focus on sound analysis and processing, image analysis, and computer vision. He coauthored over 150 scientific publications on international journals and congresses as well as numerous patents in the image and sound processing areas.



**Giovanni De Sanctis** received the “Laurea” degree in electronic engineering from the Politecnico di Milano, Milano, Italy, in 2003. He is currently working toward the Ph.D. degree on parameters estimation for physical modeling in the Sonic Arts Research Centre, Queen’s University Belfast, Belfast, U.K.

Prior to beginning his doctoral studies, he worked for three years as a Contract Researcher with the Image and Sound Processing Group, Politecnico di Milano, working on sound synthesis through physical modeling and acoustic source localization for human–computer interaction. His research interests include signal processing, with an emphasis on sound-source modeling and tactile interfaces.