# AN INTEGRATED SYSTEM FOR THE AUTOMATIC BLOCK-WISE SYNTHESIS OF SOUNDS

*Giovanni De Sanctis, Augusto Sarti, Gabriele Scarparo, Stefano Tubaro*

Dipartimento di Elettronica e Informazione
Politecnico di Milano - Piazza L. da Vinci 32,20133 Milano,Italy
email: desancti/sarti/scarparo/tubaro@elet.polimi.it

## ABSTRACT

Current physics-based synthesis techniques tend to synthesize the interaction between different functional elements of a sound generator by treating it as a single system. However, when dealing with the physical modeling of complex sound generators this choice raises questions about the resulting flexibility of the adopted synthesis strategy. One way to overcome this problem is to approach it by individually synthesizing and discretizing the objects that contribute to the generation of sounds. In this paper we address the problem of how to automate the process of physically modeling the interaction between objects, and how to make this interaction time-varying in its topology. We show how a solution based on binary connection trees can be fruitfully employed in an integrated modeling system that is able to automate the synthesis of interactions between objects. We also show that, with this approach, the modeling of physical interactions can be done in an entirely graphical fashion. We finally provide a description of the Graphical User Interface for a user-friendly authoring of interactional models and an overview of a live performance system based on this technology.

## 1. INTRODUCTION

In the past few years the interest in Wave Digital Filters (WDFs) has grown a great deal, as the research in musical acoustics started to turn toward synthesis through physical modeling. This is, in part, due to the fact that WDFs are able to preserve many properties of the analog systems that they model, with particular reference to passivity and losslessness [1]. This renewed interest in WDFs, however, is also due to the popularity gained in the past decade by Digital WaveGuides (DWGs) [6], which can be seen as close relatives of WDFs. Classical WDFs are able to incorporate nonlinear elements by connecting their wave version to the adapted port of the structure. In addition to resistive nonlinearities (frictions), it is possible to use Wave Digital (WD) principles to accommodate reactive nonlinearities (e.g. nonlinear stiffnesses), or more general nonlinear elements with memory [4]. In order to do so, we can define new waves with respect to which the description of the nonlinear elements becomes memoryless. The wave transformation is performed by dynamic multiport junctions and adaptors with memory that can be proven to be non-energetic [4]. Such multiport junctions are called *dynamic adaptors*, as their reflection coefficients are, in fact, reflection filters.

The importance of being able to model a wide range of nonlinear elements and accommodate them into WD structures is particularly relevant in musical acoustics. Vibrational phenomena in physical models, in fact, are generated and supported by a nonlinear interaction between structures (at least one of which must have resonating properties). As an example, a bow and a string interact in a highly nonlinear fashion (stick-and-slide interaction, discovered long ago by Helmoltz) to produce the sound of a violin. Similarly, a hammer interacts with a membrane through a highly nonlinear (percussive) fashion. It is, in fact, by the structure of this nonlinear interaction that a timbral class (strings, brass, woodwinds, percussions, ...) is completely characterized. Although the literature is rich with ad-hoc solutions for excitational interactions of various nature, no attempts, that we are aware of, have been made to develop systematic and automatic strategies for modeling and implementing such interactions. The need for solutions in this direction is, in fact, very strong when we need to manage arbitrary interactions between many pre-existing models in a simple and automatic fashion. Situations of this sorts are encountered, for example, in the sonification of acoustic events in virtual reality, or just in the modeling of rich percussion sets.

Excitational interactions of acoustic interest are aimed at generating vibrations in a resonating structure through some nonlinear action such as friction, percussion, positional setting, plucking or rolling. It is the interaction nonlinearity that is responsible for the timbral dynamics of the model. We need to remember, however, that modeling this nonlinear interaction raises stability-preservation issues in the model discretization process. As our goal is to develop a SSPM technique of practical interest, we need to have strict requirements on the computational complexity and avoid having to increase the sampling rate just to make sure that the numerical implementation of the model will preserve the stability properties of the analog reference model.

We recently showed that it is possible to use such principles in order to model physical structures in a block-wise fashion through a systematic and automatic procedure [7]. Working in a block-wise fashion means constructing a number of individually synthesized blocks and connecting them together using a properly defined interconnection network. In this paper we show that this automatic procedure can be implemented for dynamically changing topologies, and in a very cost-effective fashion.

## 2. AUTOMATING THE SYNTHESIS USING THE BCT METHOD

As we are interested in modeling the linear and the nonlinear interaction between blocks in a physical fashion, we assume that a library of blocks is available for the construction of the model. This is not a restrictive assumption, as the blocks of this library can, in fact, be constructed using our own approach. Furthermore, the literature is rich with methods for the physical modeling of blocks in the WD domain, therefore our method allows us to easily exploit such solutions for the construction of arbitrary interactional models.

Our library of blocks already includes a variety of blocks constructed as a physical interconnection of elementary WD blocks, other obtained through the discretization of PDE, or using the Functional Transformation method [8, 9]. It also accommodates various mutators [4], i.e. two-port adaptors (scatterers) with memory, and other types of adaptors developed for modeling typical nonlinear elements of the classical nonlinear circuit theory (both resistive and reactive).

In order to devise a systematic approach to the implementation of W structures we need an appropriate data structure and a *method* that allows us to compute incident and reflected waves at each bipole. If the circuit were memoryless, we would only need to apply our method to our data structure once in order to derive the solution vector (i.e. a configuration of waves that complies with the intrinsic I/O relationships of the blocks and the global continuity laws). In all practical cases of interest, however, our circuit is not instantaneous, therefore the solution vector ends up containing the system's *memory*. Once we assign such vector an initial configuration, at each iteration we update its content, to produce the next instance of the solution.

Our method is organized in an iterative fashion and is based on the direct inspection of the numerical structure. The method, in fact, starts from the reflected waves on the bipoles and follows their path throughout the whole structure once every time sample. In order to generate the path, we scan the tree that describes the circuit topology [5]. If the structure is based just on three-port junctions, the resulting connection tree turns out to be binary (hence the name binary connection tree). The BCT formally describes the interconnection topology of the adaptors under the following rules:

- the **root** corresponds to the adaptor that the nonlinear (NL) element connects to;
- the **nodes** are 3-port standard WDF adaptors and the branching topology matches the actual adaptor's interconnection topology;
- the **leaves** correspond to the bipoles.

Once the connection tree is built, the computational procedure can be constructed in two steps: a *forward scan* of the tree (from the leaves to the root), followed by a *backward scan* (from the root to the leaves). In fact, the computation begins from the memory cells, which are in the leaves of the tree and contain all the initial conditions of the system and keeps nesting function calls until we reach the root (NL element), obtaining the reflected waves at the adapted ports of each adaptor. In the backward scan, once we have the wave reflected by the NL element, all other reflected waves can be computed, reaching the leaves again and updating their content with the reflected wave of the adaptor they are connected to. In other words, following this path we always have all necessary data to compute the waves we need.

The initialization procedure follows a similar approach [5]. Determining the initial condition means solving a set of equations, one of which is nonlinear. Indeed, the solution of this set of equation is rather simple, as it requires a search for a fixed point. The problem is to specify the set of equations starting from the connection tree. Since during this phase the reactances are formally replaced by ideal generators, it is not possible to use W variables directly, because they do not have an adapted representation and the structure would turn out to be non-computable. However, we can still use the tree structure that describes the circuit topology, which works irrespectively of whether we are working in the W domain or in the K domain. The process can again be splitted 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. This way, during the backward scan, knowing one of the two variables, we can compute the other one using these characteristics.

One key feature of this approach is that its computational cost and memory requirements increase linearly with the number of adaptors. Of course, this improved efficiency costs in terms of evocative power of the structure.

## 3. MANAGING TIME-VARYING STRUCTURES

Changing any model parameter in a WD structure usually affects all the other parameters as they are bound to satisfy global adaptation conditions. Temporal variations of reference resistances, on the other hand, are implemented through a re-computation of the model parameters on the behalf of a process that works in parallel with the simulator. Using the BCT method, when the value of a leaf changes, the adaptors that need to be updated are only those lying on the path that link the leaf to the root (fig. 1).

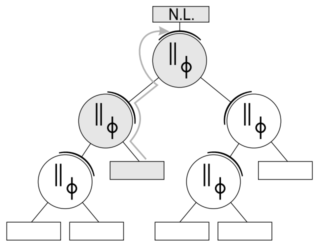


Figure 1: Tree updating after a bipole value change.

Let us consider an object that could potentially interact with a number of other objects in a sound environment. For example, we could think of a mallet that could potentially collide with a number of drum-like resonators. Indeed, this situation cannot be implemented with a fixed interaction topology. In order to be able to implement this dynamic topology, we need to be able to connect or disconnect objects on the fly. This can be achieved by exploiting the fact that a connection between systems becomes *irrelevant* when their contact condition is not satisfied.

Working with BCTs, in fact, is simpler, as they naturally offer an enhanced flexibility in managing topological changes. Assuming, for the sake of simplicity, that the two circuits connect with each other through a single *interconnection port*, we would like their port to become "transparent" when the objects are isolated (no contact). This means that the port resistance is zero if it comes from a series adaptor, or infinity if it comes from a parallel one. We must remember, however, that the interconnection of two circuits could originate computability problems in the wave (W) domain, particularly if both circuits contain a nonlinear element (NLE). In a wide variety of acoustic physical models, however, NLEs are separated by instantaneously decoupling multiports, such as DWGs, therefore they can be safely connected together.

Even when we need to interconnect a linear W system with a nonlinear one, we still need to have some element that enables the connection. Since we are in a situation in which we do not need any decoupling, this interconnection element could also be memoryless. In a linear circuit the root of the BCT could be any of the bipoles (if have a BCT and have it *dangling* from another one of its nodes, we will end up with another BCT). If a linear circuit has an interconnection port, we can take that as the root of the BCT, so that it can act as the "shoot" (subroot) to be "grafted" to the receiving tree. Notice, however, that the state update equation does not treat the instantaneous interconnection port as a bipole, as it does not "contain" a numerical value but a pointer to another structure.

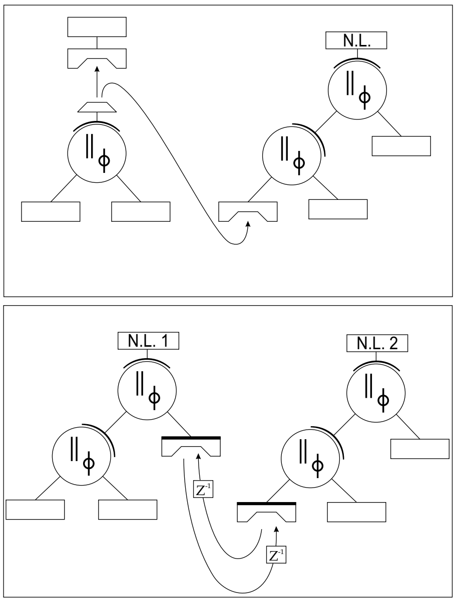


Figure 2: Memoryless (up) and dynamic (down) interconnection ports. The bold border indicates the instantaneous adaptation due to the memory

Let us consider a W hammer model interacting with the W model of a string. The W hammer is made of a mass and a nonlinear spring that models the lossless and instantaneous limited compressibility of the felt. Both systems can be modeled with a single circuit but, to explain the above method, are here kept as separate through memoryless interconnection ports. During the interaction we can identify the following

1. initially the objects are far apart and their ports are disconnected. Such ports are transparent with respect to their circuits. In fact, the string port is a series one, therefore it is a short circuit; while the hammer port is a parallel one, therefore it is an open circuit.
2. When hammer and string are close to each other (proximity condition) we can establish a connection, and the string BCT can be grafted into the hammer BCT, originating a single structure. As far as the circuit behavior is concerned, however, nothing has changed, as the series adaptor is still short-circuited by the NLE, which is working on the linear portion of its characteristics with slope -1.
3. The situation changes when the hammer comes in contact with the string (contact condition), i.e. when the working point on the NLE characteristics begins changing slope. From now on, there is a non-zero power exchange between elements, therefore the hammer will begin bouncing against the string until it will be push away from it.
4. When the hammer is sufficiently far apart from the string, the proximity condition ceases to be valid, therefore the connection can be removed and the circuits are once again isolated.

Notice that although the interconnection ports and the particular behavior of the NLE (a step function in the K domain) play a similar role, irrelevant interconnections and absence of connection have consequences on the organization of the implementation. In fact, when the hammer is disconnected, it can be used elsewhere. Roughly speaking, a piano harp can use a limited amount of shared hammers

## 4. THE BCT GUI

We developed a Graphical User Interface for a user-friendly construction and testing of BCT-based models. Models are constructed in a block-wise fashion by creating a visual network of physical elements with the help of a graphical parser whose aim is to make sure that the resulting model will be consistent with the requirements set forth by the BCT methodology.

Each block is picked from a palette (dynamically constructed from a properly defined library) and connected using BCT nodes. Double-clicking a block will prompt a local GUI to appear, in order to set the parameters of that block. One special local GUI enables the editing of the nonlinearities. The editing can be done directly in the Kirchhoff domain for nonlinear resistances, capacitors and inductances.

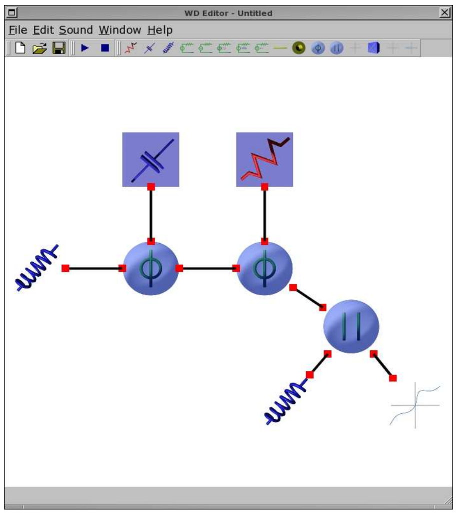


Figure 3: A Fender Rhodes model.

The GUI has replaceable "skins", as it can accommodate a visual metaphor based on WDFs, or an abstract one where only BCT nodes are shown. The BCT structure of a Fender Rhodes model (with a WDF "skin") is shown in figure 3.

Both synthesis authoring environment and synthesis engine are able to accommodate multiple-tree structures. Several trees (with their own nonlinearities) can, in fact, be loaded and edited in the same sandbox, and their output directed to different audio channels. The BCT GUI is able to handle all the topological operations on trees (e.g. bridging, grafting, etc.) described above.

There are several motivations behind the need to accommodate multiple-trees. For example, if we have two interconnected structures that interact with each other in a non-instantaneous fashion, then we are in a situation in which the two trees are connected with each other through a bridging mechanism (leaves of the trees are interconnected in a delayed fashion). This, for example, is the situation in which two excitation blocks (for example mallets) interact in a nonlinear fashion (percussion) with a single resonating structure in differen points (delayed interconnection).

As far as the real-time parameters control is concerned, the synthesis engine fully supports and integrates the MIDI protocol. It is, in fact, possible to modify any parameter with MIDI controls (such as pitch bend, modulation wheel, volume control) in addition to GUI-specific controls (e.g. sliders or spin boxes). All such settings are defined within a single dialog window as they can be loaded from and saved into a model file, within a specific section.

## 5. THE LIVE PERFORMANCE SYSTEM

In order to test the performance effectiveness and the playability of the system, we developed a live performance system that is based on this interactional approach. The system integrated a variety of categories of blocks developed in cooperation with the University of Erlangen-Nurenberg [8, 9] and the Helsinki University of Technology [10], within the EC-funded ALMA project[1]. Using the live performance system all such blocks are allowed to freely interact with each other in a nonlinear fashion through BCT interconnections. The system is able to simultaneously run many BCT models during a live performance. The BCT synthesis engine is fully compatible with both MIDI controllers and commercial sequencers. The live performance system is based on 4 "cube" PC (Intel Pentium IV 3GHz) running a BCT engine each. These PC are connected through a LAN to a laptop that acts as the main console and routes the MIDI data. The laptop also runs the sequencer software. The audio generated by the PCs is collected through stereo audio connections and sent to an audio mixer. The system thus produces 8 independent audio channels, and is able to add independent audio effects to each one of them (reverbs, choruses, equalizations etc.). Such effects are modeled directly in the WD domain and are treated as blocks of the BCT system. This setup can be expanded connecting up to 8 PC directly to the MIDI interface. Moreover, as the BCT engine can read MIDI data over the LAN, any number of PCs can be connected. The system has already been used in two live performances by professional musicians, which are currently collaborating to improving it further.

## 6. CONCLUSIONS

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, both the Wave Tableau approach and the Binary Connection Tree approach we implemented make the construction and the implementation of the interaction topology systematic. 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 CAD approach to the construction of an interactive sound environment.

[1] http://www-dsp.elet.polimi.it/alma

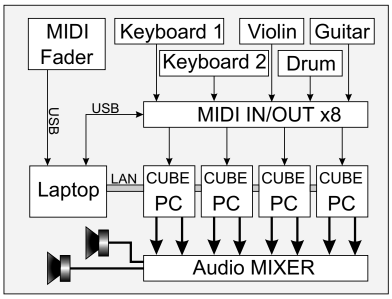


Figure 4: Block diagram of the live performance system.

## REFERENCES

[1] A. Fettweis: "Wave digital filters: theory and practice". *Proceedings of the IEEE*, Vol. 74, No. 2, pp. 327–270, Feb. 1986.

[2] F. Pedersini, A. Sarti, S. Tubaro: "Block-wise Physical Model Synthesis for Musical Acoustics". *IEE Electronic Letters*, Vol. 35, No. 17, Aug. 1999, p. 1418-19.

[3] A. Sarti, G. De Poli: "Generalized Adaptors with Memory for Nonlinear Wave Digital Structures". *VIII European Signal Processing Conference*, 1996, Trieste, Italy, Vol. 3, pp. 1773-6.

[4] A. Sarti, G. De Poli: "Toward Nonlinear Wave Digital Filters". *IEEE Transactions on Signal Processing*. Vol. 47, No. 6, June 1999.

[5] G. De Sanctis, A. Sarti, S. Tubaro: "Automatic synthesis strategies for object-based dynamical physical models in musical acoustics". 6th Intl. Conf. on Digital Audio Effects DAFx-03, Sept. 8-11, 2003, London, UK.

[6] J.O. Smith, "Principles of digital waveguide models of musical instruments", in *Applications of digital signal processing to audio and acoustics*, edited by M. Kahrs and K. Brandenburg, Kluwer, 1998, pp.

[7] G. De Sanctis, A. Sarti, S. Tubaro, "Automatic synthesis strategies for object-based dynamical physical models in musical acoustics."6th Intl. Conf. on Digital Audio Effects DAFx-03. Sept. 8-11, 2003, London, UK.

[8] L. Trautmann, S. Petrausch, R. Rabenstein, "Physical modeling of drums by transfer function methods". Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP '01). Vol. 5, May 7-11, 2001 p. 3385-3388, Vol.5.

[9] Stefan Petrausch, Rudolf Rabenstein, "Interconnection of State Space Structures and Wave Digital Filters". To appear on IEEE Tr. on Circuits and Systems II, Volume: PP, Issue: 99, 2005 p. 1-4.

[10] M. Karjalainen, C. Erkut, "Digital waveguides versus finite difference structures: Equivalence and mixed modeling". EURASIP Journal on Applied Signal Processing, Vol. 2004, No. 7, July, 2004, pp. 978-989.