AUTOMATIC MODELING AND AUTHORING OF NONLINEAR INTERACTIONS BETWEEN ACOUSTIC OBJECTS

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

The physical synthesis of musical sounds is based on the modeling of the nonlinear interaction between two separate structures (at least one of which with desirable resonating properties). Such structures and their interaction are usually modeled with a global approach (as a single system), but flexibility and scalability issues suggest us that it would be highly desirable to adopt a block-wise approach, based on the individual synthesis and discretization of the interacting parts. In this paper we address the problem of how to automatize the modeling of this nonlinear interaction and how to enable the on-the-fly fusion of model that corresponds to this interaction. We show how a solution based on binary connection trees can be fruitfully employed in an integrated modeling system that is able to automatize this sound synthesis approach. We also show that, with this solution, the modeling of physical interactions can be done in an entirely symbolic (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

The computer simulation of vibrational phenomena in musical instruments represents one of the earliest sound production techniques envisioned in the literature of machine-generated music. After the first experiments published in the early fifties, a long series of works followed, initially focused on the study of the physical phenomena that occur in musical instruments, and then oriented towards the possibility of generating sounds for musical applications. The interest in Sound Synthesis through Physical Modelling (SSPM) grew steadily after then, as more computational power became available at low cost. The reasons of this interest in SSPM are numerous, mostly related to the intuitive link that exists between control and model reaction; to the ability to generate musically interesting timbral spaces associated to the changing of just a few physically plausible parameters; and to the possibility to exploit a well-established modelling experience in musical acoustics. In order for SSPM to become of commercial interest, however, many issues still need to be settled. SSPM, in fact, needs to become more than just simulation: it must become a systematic, automatic, simple, physics-inspired approach to the construction of sound generation models.

In order to understand what to do to achieve this result, it is important to clarify a few things about SSPM and its meaning. A physical structure for generating sounds is generally made of two elements that interact with each other to produce the vibration. Usually one of the two elements acts as an *excitator* (e.g. bow, reed) and the other as a *resonator* (e.g. string, acoustic tube), but this distinction is of lesser importance. Each one of these elements, in turn, can be a combination of several simpler elements (e.g. string, bridge and soundboard in a traditional acoustic model structure). The timbre that is produced by a specific model is characterized by two aspects: the vibrational properties of the resonating structure; and the way the two elements interact with each other to generate and support the vibrations. The interaction mechanism characterizes a timbral class (e.g. strings, reeds, brass, percussions, etc.), while the properties of the resonating structure characterize the richness and the quality of the sound within that class. If we need to specify n parameters to produce a sound with a given model, then we say that the timbral space that the model offers is n-dimensional. This timbral space can be explored during the composition process or visited during the performance.

All sorts of resonating structures of musical interest have been studied and modeled quite extensively in the literature of SSPM. A variety of models are, in fact, available for this purpose, which range from finite elements modeling, to modal synthesis, to functional transformation methods [7, 8], to Digital WaveGuide (DWG) modeling solutions. Of particular interest for musical acoustics are the last two of such methods. The former, in fact, provides the user with a meaningful and minimal set of control parameters for the characterization of the vibrational phenomena, which results in responsive and musically rich timbral spaces that are easy to explore and navigate. The latter enables an efficient block-wise construction of resonating structures of all sorts.

On the other hand, one aspect that has received less attention in the literature of SSPM is the systematic characterization of the interactions between elements, which generate and support the vibrational phenomena in the resonator. In fact, 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.

One reasonable assumption that make in our approach is that the reference analog system is described in a lumped-parameter fashion, i.e. as an interconnection of blocks that communicate with each others through ports, each characterized by a pair of dual (Kirchhoff) variables: a *through* variable, such as velocity, flow, and current; and an *across* variable, such as force, pressure and voltage (which does not mean that the blocks cannot be internally described in a distributed-parameter fashion). This choice allows us to monitor the *energy* flowing through each port and keep the algorithmic stability under control. As a matter of fact, we do not use directly such pairs of variables but pairs of incident/reflected waves, which can be obtained as linear transformations of through/across pairs.

2. WORKING IN THE WAVE DIGITAL DOMAIN

The choice of working with waves instead of Kirchhoff variables is motivated by the need of working in a block-wise fashion while avoiding computability problems. The most popular approach that is based on this choice is is that of Wave Digital Filters (WDFs), developed by Fettweiss [1] over three decades ago. The interest of SSPM toward Wave Digital Filters 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], and to the popularity gained in the past decade by Digital WaveGuides (DWGs) [5], which can are close relatives of WDFs. In order for WDF/DWGs to be useful in musical acoustics, however, they need to be able to accommodate nonlinearities, which are key to the generation of vibrations in resonators.

WDFs are able to incorporate memoryless nonlinear elements by connecting their wave version to the adapted port of the structure. In addition to resistive nonlinearities (frictions), WDF can also accommodate reactive nonlinearities (e.g. nonlinear stiffnesses), or more general nonlinear elements with memory [4]. In order to do so, we 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.

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 [6]. 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.

3. AUTOMATIZING THE SYNTHESIS USING THE BCT METHOD

Some methods are already available for synthesizing macro-blocks, therefore the automatic synthesis procedure is based on the assumption that such elements are already available in the form of a collection of pre-synthesized structures. Currently, the family of blocks includes WD mutators [4] 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.

Currently two different methods are available: the first is inspired by the *tableau analysis method*, commonly employed in circuit theory to analytically determine the evolution of (analog, time-varying, linear) electrical circuits, while the second is based on a direct inspection of the numerical structure (the circuit) according to a tree-like structure that describes the interconnection topology of the elements. We refer to the first method as *Wave Tableau* (WT) method, and we call the second approach as *Binary Connection Tree* (BCT) method.

In spite of the dramatic improvements introduced in the WT method, we decided to explore a different and novel approach, organized in iterative form. This method turns out to be the most efficient one, as it is based on a direct inspection of the numerical structure. The method 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, in fact, we scan a tree that describes the circuit topology. 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). Like in the WT method, also with the BCT we do not necessarily need to use three-port adaptors. However, considering that any N-port adaptor can always be decomposed into the interconnection of $N-2$ three-port adaptors, we can use a BCT with no loss of generality. 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. 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.

4. MANAGING TIME-VARYING STRUCTURES AND TOPOLOGICAL CHANGES

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.

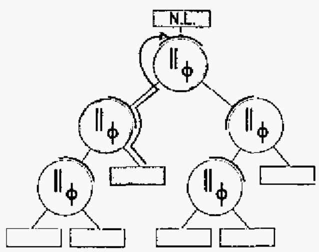

Figure 1: Tree updating after a bipole value change.

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 W domain, particularly if both circuits contain a 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.

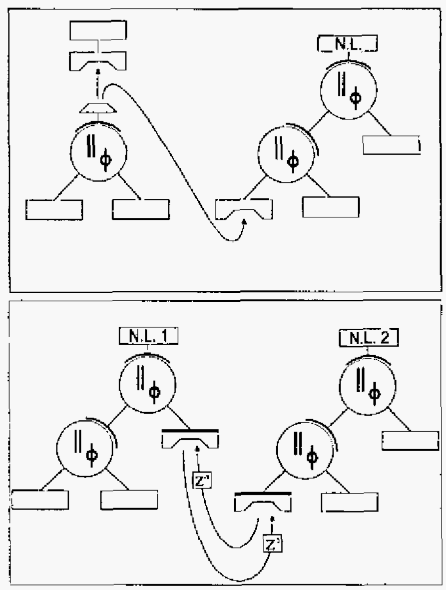

Figure 2: Memoryless (up) and dynamic (down) interconnection ports. The bold border indicates the instantaneous adaptation due to the memory

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

5. 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. By double-clicking any block, a local GUI will appear, for setting the parameters to be associated to that block. A particular interface has been implemented in order to edit the nonlinearities. On the axis are represented the pairs of variables (V,I), (V,Q) and (Φ, I) for the cases of a non-linear resistance, a non-linear capacitor and a non-linear inductance, respectively.

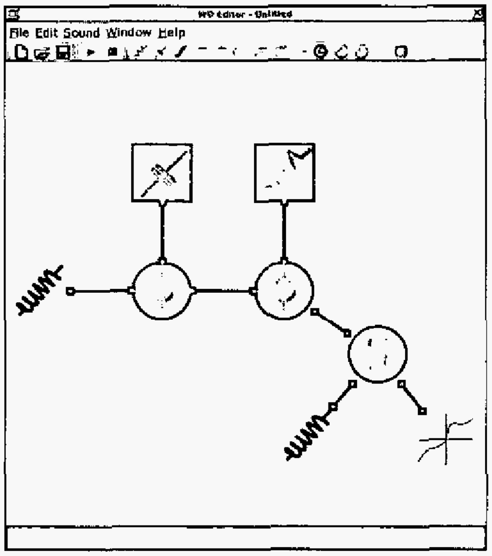

Figure 3: A Fender Rhodes model.

The GUI is able to accommodate a visual metaphor based on WDFs, but this can be easily replaced with an abstract one in which only BCT nodes appear: the user can select the appearance metaphor as he/she pleases. A more technically-oriented one could be more interested in WDF metaphor, while a more musically-oriented one could be more interested in an abstract appearance: in this case the software chooses automatically the correct ports when the user connects two elements. The BCT structure of a Fender Rhodes model is shown in figure 3.

Multiple-tree structures are fully supported: different trees and their respective non-linearities can be loaded and edited in a single sandbox, and their output directed to different audio channels. BCT-GUI is able to manage all the topological operations on trees (e.g. bridging) developed in BCT theory.

The implications of this choice are, perhaps, deeper that it may seem. The idea is that 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 (which communicate with each other in a delayed fashion).

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.

6. 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 [7, 8] and the Helsinki University of Technology [9], within the EC-funded ALMA project [?]. 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 a certain number of BCT models during the live performance. In addition to the performance mode, there is also the possibility to

can be either live (played using MIDI controllers) or "offline" (played using a MIDI sequencer). The 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. Since each stereo output can be thought as two outputs, the system can manage 8 independent audio channels, adding individual audio effects (reverbs, choruses, equalizations etc.). 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.

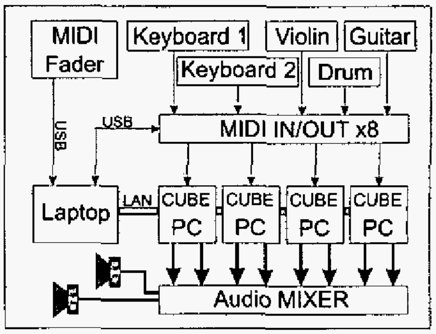

Figure 4: Block diagram of the live performance system.

7. 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.

8. 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] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.