2004 IEEE 6th Workshop on Multimedia Signal Processing

Physics-based modeling of musical instruments for interactive virtual reality

Matti Karjalainen*, Teemu Mäki-Patola†
Helsinki University of Technology
*Laboratory of Acoustics and Audio Signal Processing, and
†Laboratory of Telecommunication Software and Multimedia
P.O.Box 3000, FI-02015 HUT, Finland
Email: {firstname.lastname}@hut.fi

Abstract— **In this paper we describe how different physics-based modeling paradigms can be used to make sound synthesis of musical instruments. We also discuss different ways to construct user interfaces for interactive virtual reality applications. For multi-paradigm modeling we have investigated how digital waveguides (DWGs), finite difference time domain schemes (DFTDs), wave digital filters (WDFs), modal decomposition models, and source-filter models interrelate and mix together. We have investigated the user interface control of these models both in a cave-like virtual room and using special hand-held controllers. Modeling examples of a "virtual air guitar" and a virtual bell-xylophone are described and demonstrated.**

I. INTRODUCTION

Sound synthesis based on physical modeling has been recently one of the main research topics in computer music [1], [2]. It allows for realistic sound with highly flexible parametric control. Therefore it is well suited to interactive control of virtual instruments. The research challenge is to develop both advanced instrument synthesis models and experessive ways to control them.

In this paper we report on recent results from Helsinki University of Technology in an EU IST project ALMA (ALgorithms for the Modelling of Acoustic interactions). The modeling methodology as well as the physics-based models are developed primarily in the Laboratory of Acoustics and Audio Signal Processing, and the user interfaces and controllers are primarily studed in the Laboratory of Telecommunication Software and Multimedia. This paper is an overview on these methods with some case examples.

II. PHYSICS-BASED MODELING PARADIGMS

Several different approaches and paradigms have been applied to physics-based modeling of musical instruments. Recently there has been growing interest to deepen the knowledge to see a full picture how the different modeling paradigms are interrelated and how they can be optimally applied and mixed together [3], [4]. In this section we present a brief overview of the most important modeling paradigms, including the Digital Waveguides (DWG), Finite Difference Time Domain schemes (FDTD), Wave Digital Filters (WDF), and some others, as well as how they can be applied within a single model.

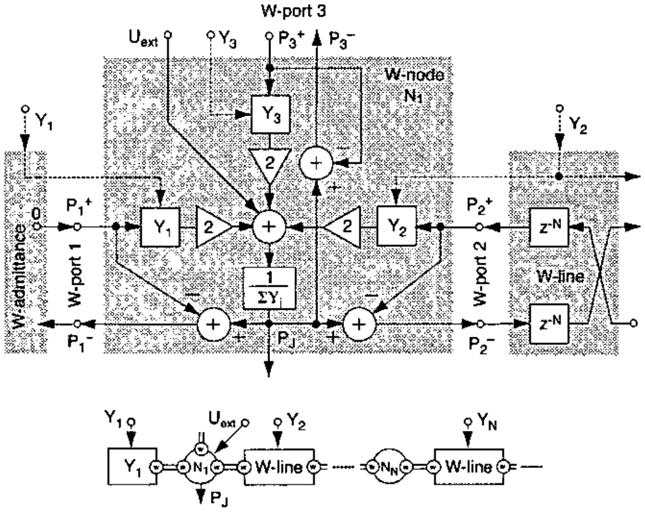

Fig. 1. *Top*: A 3-port parallel scattering junction for acoustic pressure waves. Incoming pressures are P_i^+, outgoing ones P_i^-, and P_J is common junction pressure. Port 1 (left) is terminated by admittance Y_1, port 2 (right) is connected to a delay-line having wave admittance Y_2, and port 3 (top) is not connected. Possible admittance controls are marked by dashed lines. *Bottom*: Block diagram with abstracted blocks and how they can be connected to form a 1-D digital waveguide.

A. Digital Waveguide (DWG) modeling

Digital Waveguide modeling [5] is based on the d'Alembert solution of the wave equation, i.e., the principle of wave propagation decomposed into two components traveling in opposite directions. In Fig. 1, the two-directional delay line (elements z^{-N}) at right-hand side implements it computationally. The shaded area in the middle of the figure shows how a (parallel) junction of elements in an acoustic system can be simulated to make a scattering junction so that the Kirchhoff continuity laws are obeyed. Three wave ports are depicted, two of them connected to ports of elements as described in the figure caption. The principle is applicable to connecting any number of ports of arbitrary admittances. The bottom part of the figure characterizes how the computation of the model can be abstracted into basic computational elements.

Digital waveguides are known as a robust and efficient way of realizing physical models and algorithms. They are now the most popular method for physics-based sound synthesis.

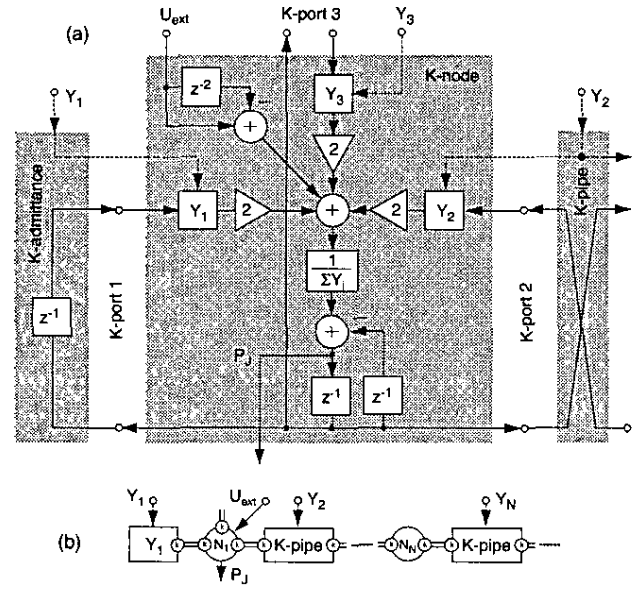

Fig. 2. *Top*: An acoustic three-port parallel connection of FDTD type, corresponding to the DWG in Fig. 1. *Bottom*: Abstraction as a blockwise description of a 1-D FDTD waveguide.

B. Finite Difference Time Domain (FDTD) modeling

Finite Difference Time Domain modeling [6], [7] is based on direct discretization of the wave equation. The second-order partial derivatives are replaced by symmetric second-order differences, both for time and place. This results in a recursion formula for each spatial position where the two neighboring nodes and the value of the node itself one time step before are used to compute the next value of the node.

We have generalized this to structures where the wave impedances/admittances of connected branches can be arbitrary [4]. Figure 2 depits this as a signal processing diagram. Notice the partial similarity of structures in Figs. 1 and 2. The main differences are that an FDTD junction has two internal unit delays and direct connection between junction nodes, while a DWG has unit delays between nodes. While DWGs are based on wave quantities (*W-modeling*), FDTDs are based on Kirchhoff quantities (*K-modeling*). We have shown these structures to be functionally equivalent [4] when FDTD connections are made as shown in Fig. 2. Notice that the external volume velocity U_{ext} needs to be fed through filter $1 - z^{-2}$ to make it behave in a physically relevant way.

FDTDs have the advantage that only two unit delays per node are needed in any dimensionality, while DWGs need $2 \times K$, where K is the dimensionality of the model. On the other hand, FDTDs are numerically less robust to compute.

C. Wave Digital Filter (WDF) modeling

Wave Digital Filters [8] originate from analog electric circuits, so that lumped element models can be elegantly simulated in discrete time by using them. Through analogy they can be applied as well in other physical domains such as the mechanical and acoustical ones. WDFs are based on wave quantities (thus *W-modeling*) so that ports to elements carry the incoming and the outgoing wave component. Lumped elements such as resistors, capacitors, and inductors can be connected to circuits through series or parallel connection elements called adaptors. Figure 3 shows how a series RCL-circuit is realized as a WDF model. It could represent as well a mass, spring, and damper system or an equivalent acoustical system.

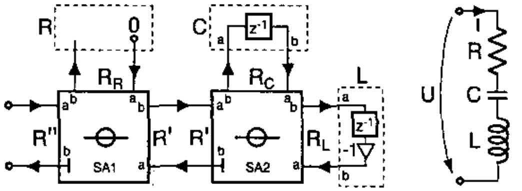

Fig. 3. (Left) A WDF series connection of resistor (R), capacitor (C), and inductor (L) constructed by two three-port series adaptors (SA1 and SA2). (Right) Equivalent analog circuit.

WDFs are know to be highly robust computationally. The formalism has been extended to include many element and adaptor types as well as to multidimensional networks. There are methods to model also nonlinearities in a physically 'correct' way [9]. WDFs are compatible with DWGs, actually these two formalisms are merely two different and complementary approaches to general W-modeling.

D. Mixed modeling

K- and W-modeling techniques are not directly compatible since they utilize different variable types. It would sometimes be useful, however, to combine them in a single model. In [10], [4] we have introduced a generalized method to do this through a *KW-converter* block. Figure 4 depicts such a converter between a K-modeling node (FDTD) and a W-modeling node (DWG). By applying the KW-converter elements it is now possible to realize arbitrary circuits and networks so that different modeling paradigms can be applied to different subparts of a model.

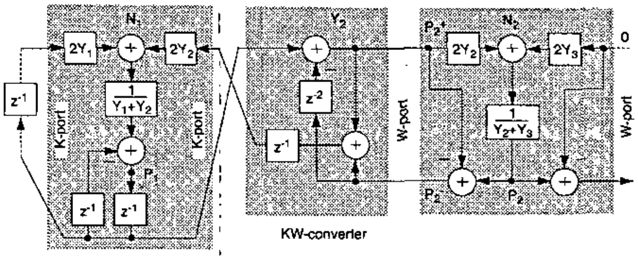

Fig. 4. KW-converter for mixed modeling with FDTDs (left-hand side) and DWGs (right-hand side).

E. Other modeling paradigms

While the DWG, WDF, and FDTD methods above are explicit time-domain simulation techniques, *modal decomposition* methods rely on frequency-domain formulations of systems under study. They decompose the behavior of a system into decaying exponentials, whereby oscillatory components represent resonances or eigenmodes of the system. Modal decomposition methods include the traditional *modal synthesis* [11] and a newer approach called the *Functional Transformation Method* (FTM) [13]. Modal decomposition models and modal synthesis can be realized as a technique by itself, such as the Modalys software [12], or by the WDF methodology.

Source-filter modeling is a reduced form of physics-based modeling so that two-directional causal physical interaction is reduced to one-way interaction, therefore allowing normal signal processing realizations using a signal generator (source) and a digital filter. Classical examples thereof are speech synthesis algorithms using formant filters or LPC (linear prediction) filter formulations. The Karplus-Strong model [14] is a classic case of modeling musical instruments, particularly string instruments, considered as an extremely reduced form of physics-based modeling.

F. Software tools for physics-based modeling

A block-based software tool called BlockCompiler [15] was developed for flexible yet efficient physics-based and DSP-based modeling, allowing for easy experimentation with the paradigms mentioned above. It is a C-code generator, which can run the models in real time, or from which the code can be exported to another run-time environment such as the Mustajuuri platform [16].

III. USER INTERFACES FOR CONTROLLING VIRTUAL INSTRUMENTS

As the sound of physics-based models can be produced in real-time, it becomes possible to alter any model parameter while playing. This creates a need for controllers that meet the sound model complexity by input flexibility. Virtual reality (VR) input technology, such as data gloves and location/orientation trackers with gesture analyses, is one possibility to offer several degrees of freedom. We have made several experimental VR interfaces for our sound models.

A. Virtual room environment (EVE)

The VR system that we used for making the interfaces is a cave-like virtual room, EVE (see Fig. 5). The virtual world is projected on the walls of a three meters wide cube. Our virtual room has three back-projected walls and a floor, onto which image is projected from above. The subjects wear shutter glasses that filter different images individually to each eye, letting the subject see three-dimensionally (3-D).

The location and orientation of the user's head and the data gloves is read by a Motionstar magnetic tracker at a rate of 100 Hz. The 5DT data gloves that we use return the flexure of each finger as one integer.

Fig. 5. Playing of a virtual xylophone in the EVE virtual room.

The sound models processed by the Mustajuuri software platform are controlled over local network from an SGI Onyx running the virtual reality application. Any MIDI input devices can be easily combined with the VR user interfaces.

B. Control interface issues

VR input hardware offers a very different kind of control medium than the physical interfaces that we are accustomed to. For instance, it is difficult to implement tactile feedback into the VR interfaces. The user's hand goes through virtual objects unobstructed. However, some tactile feedback can be added by introducing real objects to the interface.

Detecting small and precise finger movements is impossible for everything but the most expensive VR hardware. The practical precision of the magnetic tracking is also in the order of centimeters.

A VR system always introduces some latency. It is important to know how much latency can be allowed for different control paradigms and how it affects the control. We have studied the effect of latency with user tests [17], [18].

All of these issues shape the area for what kind of interfaces VR suits for. In addition to the actual interface, the mapping of input parameters to synthesis parameters defines much of the nature and expressiveness of the instrument [19].

The case studies below describe experiments where primarily the EVE environment has been used for controlling virtual instruments. Specialized experimental interfaces have been developed as well, as mentioned for the virtual air guitar.

IV. CASE STUDIES

A. Virtual air guitar

Following the idea of 'air guitar' shows, i.e., just acting the playing along music playback, our virtual air guitar is played with only hands in the air (see Fig. 6). However, it actually makes sounds through a model-based synthesizer. The control input is generated by two tracked data gloves that the player wears.

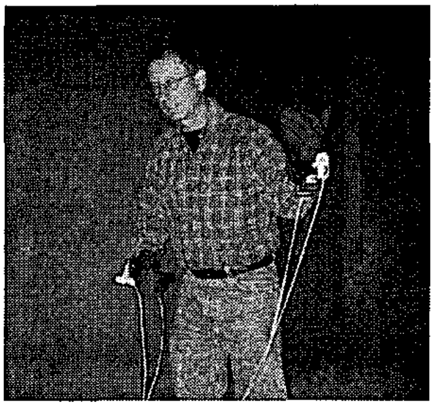

Fig. 6. Playing of a virtual air guitar.

The virtual guitar has different playing modes. In the simple mode the distance of the hands translates to pitch. Triggering is done with fingers or with shaking of the right hand. The instrument model generates an electric guitar sound, produced by coupling an extended Karplus-Strong guitar model with a distortion and loudspeaker simulation model.

In Fig. 6, data gloves are used for user interface. We are developing also other kind of controllers. One simple interface is based on measuring left-hand position for pitch control using high audio frequency pulses from a small loudspeaker to an electret microphone. Wave propagation delay indicates the left hand position. A microchip vibration sensor in a right-hand stick is used for plucking control.

Most of the playing modes of the virtual air guitar aim to be entertaining rather than a serious instrument. The user can, for instance, play rock classics. Each time he triggers a pluck the next note or chord of the song is played. With his actions the user defines the tempo of the song. The instrument implements also a mode where the user controls a Markov Chain based sound generation.

B. Virtual xylophone

A virtual xylophone interface uses interactive virtual objects, see Fig. 5. When the user closes his hands, virtual mallets appear as he would hold them. The mallets inherit the location and the orientation of the user's hands. The xylophone plates are located in the air around the user. The user can move relative to the plates. The hit location on the plate and the hit velocity are translated to sound model control parameters. The plates can be combined to make chords that are played by hitting through small plate piles.

The synthesis model is based on somewhat inharmonic digital waveguides that were originally developed to synthesize bell sounds.

Our recent, yet unpublished study suggests that people prefer a real mallet over a virtual one when hitting virtual objects. The use of a physical mallet results also in better temporal accuracy while playing a virtual percussive instrument.

V. SUMMARY

Physics-based modeling for real-time sound synthesis of musical instruments is well suited for interactive virtual reality. The parameters for model control are intuitive and closely related to the parameters used in controlling real instruments. In this overview paper we have discussed two aspects of developing such models and systems, based on recent research at HUT. First we described how to use different physics-based modeling paradigms, and secondly we discussed the user interface issues of controlling physics-based models for interactive playing of virtual instruments.

ACKNOWLEDGMENT

The authors would like to thank other ALMA project members, particularly Cumhur Erkut, Paulo A. A. Esquef, Jyri Pakarinen, Hannu Pulakka, Vesa Välimäki, Tapio Takala, and Lauri Savioja, for their contributions.

REFERENCES

[1] J. O. Smith, "Physical Modeling Using Digital Waveguides," *Computer Music J.*, vol. 16, no. 4, pp. 74–91, 1992. See: `http://www-ccrma.stanford.edu/~jos/pmudw/`.

[2] V. Välimäki, "Physics-Based Modeling of Musical Instruments," To appear in *Acta Acustica* united with *Acustica*, 2004.

[3] S. D. Bilbao, *Wave and Scattering Methods for the Numerical Integration of Partial Differential Equations*, PhD Thesis, Stanford Univ., May, 2001.

[4] M. Karjalainen and C. Erkut, "Digital Waveguides versus Finite Difference Schemes: Equivalance and Mixed Modeling," *J. Applied Sig. Proc.*, no 7, 2004, pp. 978–989.

[5] J. O. Smith, *Principles of Digital Waveguide Models of Musical Instruments*, Ch. 10 in *Appl. of Digital Sig. Proc. to Audio and Acoust.*, (ed. Kahrs and Brandenburg), Kluwer Academic Publishers, 1998.

[6] J. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth and Brooks, Grove, Ca, 1989.

[7] A. Chaigne, "On the Use of Finite Differences for Musical Synthesis. Application to Plucked Stringed Instruments," *J. Acoustique*, vol. 5, pp. 181–211, April 1992.

[8] A. Fettweis, "Wave Digital Filters: Theory and Practice," *Proc. of the IEEE*, vol. 74, no. 2, pp. 270–361, Feb. 1986.

[9] A. Sarti and G. De Poli, "Toward Nonlinear Wave Digital Filters," *Proc. IEEE Trans. Signal Processing*, vol. 47, no. 6, pp. 1654–1668, June 1999.

[10] M. Karjalainen, "Discrete-Time Modeling and Synthesis of Musical Instruments,"*Proc. Workshop on Applications of Signal Processing to Acoustics and Audio (WASPAA'04)*, New Paltz, N.Y., USA, Oct. 2003.

[11] J.-M. Adrien, *The Missing Link: Modal Synthesis*, Chapter in *Representations of Musical Signals*, (eds. de Poli, Piccialli, and Roads), MIT Press, Cambridge, Mass., 1991.

[12] http://www.ircam.fr/produits/logiciels/modalys-e.htm

[13] L. Trautmann, *Digital Sound Synthesis by Physical Modeling of Musical Instruments Using Functional Transformation Methods*, PhD Thesis, Univ. Erlangen-Nürnberg, Germany, 2002.

[14] K. Karplus and A. Strong, "Digital Synthesis of Plucked-String and Drum Timbres," *Computer Music J.*, vol. 7, nr. 2, pp. 43–55, 1983.

[15] M. Karjalainen, "BlockCompiler: Efficient Simulation of Acoustic and Audio Systems," *Preprints of AES114th Convention*, Paper 5756, Amsterdam, May 2003.

[16] http://www.tml.hut.fi/~tilmonen/mustajuuri/

[17] T. Mäki-Patola and P. Hämäläinen, "Latency Tolerance for Gesture Controlled Continuous Sound Instrument without Tactile Feedback," Proc. *Int. Computer Music Conference*, Miami, USA, Nov. 2004.

[18] T. Mäki-Patola and P. Hämäläinen, "Effect of Latency on Playing Accuracy of Two Gesture Controlled Continuous Sound Instruments Without Tactile Feedback," Proc. *Conf. on Digital Audio Effects*, Naples, Italy, Oct. 2004.

[19] A. Hunt, M. Wanderley, and M. Paradis, "The Importance of Parameter Mapping in Electronic Instrument Design," Proc. *Conf. on New Instruments for Musical Expression (NIME-02)*, Dublin, Ireland, May 2002.