Audio-Tactile Interaction at the Nodes of a Block-Based Physical Sound Synthesis Model

Cumhur Erkut, Antti Jylhä, Matti Karjalainen
Dept. Signal Processing and Acoustics
Helsinki University of Technology, Finland
{Firstname.Lastname}@tkk.fi

M. Ercan Altınsoy
Chair of Communication Acoustics
TU Dresden, Germany
Ercan.Altinsoy@ias.et.tu-dresden.de

ABSTRACT
We investigate how to link a physically-based digital sound synthesis structure to a haptic interface for constructing flexible and dynamic virtual audio-tactile environments and musical instruments. Our Block-based Physical Model (BBPM) imposes physicality on the audio signals by transforming them to a variable pair of effort and flow. With this pair, we can define the notions of energy and impedance/admittance for object blocks, and manage their physically-based interaction by using special elements we call nodes. We have recently ported a subset of the BBPM environment to Pure-Data (PD), where we can construct dynamic audio environments akin to a scene graph. Moreover, we can connect off-the-shelf haptic controllers to the entry node of a BBPM, excite the model with one physical variable, provide haptic feedback with the dual variable, and concurrently synthesize audio. We will demonstrate this system with two prototypes during the HAID'08 Workshop.

Author Keywords
Block-based physical modeling, digital sound synthesis, sonic interaction design, prototype, vibrotactile commodities.

ACM Classification Keywords
H.5 HCI, H.5.5 Sound and Music Computing, Signal analysis, synthesis, and processing

INTRODUCTION
In this contribution we investigate how to link a block-based physical model (BBPM) [1] for digital sound synthesis with a haptic interface [2]. Auditory cues frequently occur when we interact with everyday objects. Physically-based methods preserve this interaction fidelity in synthetic sound, and we can naturally control the physical parameters and variables of the models by our gestures and actions. These properties has made physical models favorable in integrated haptic and audio interfaces [3, 4]. Typical implementations, however, support interaction between two objects in a static virtual world, based on a single physical modeling paradigm (usually *modal synthesis*). A BBPM supports dynamic audio environments and multiparadigm physical modeling [1]. It imposes physicality on the audio signals by transforming them to a variable pair of effort and flow. With this dual variable pair, we can define the notions of energy, impedance, and admittance for object blocks, and manage their physically-based interaction by using special elements we call *nodes*.

In [5], construction and playing of virtual musical instruments based on BBPM techniques were investigated. For instance, a virtual xylophone that consists of data gloves, a magnetic tracker for head and hands positions, and audio-visual rendering in a cave-like virtual reality (VR) environment has been reported. The difficulty of implementing tactile feedback at the VR interface and the negative effects of latency has been pointed out. Our aim, therefore, is to experiment with simplified systems and off-the-shelf interfaces that can enhance audio-tactile interaction with low latency and low demands on the rendering architecture.

Here, we present our preliminary prototypes on audio-tactile interaction, i.e., synthesizing audio and haptic feedback at the same time by using a BBPM. We specifically deploy two prototypes that couple a BBPM environment implemented in PureData (PD)[1] with off-the-shelf controller devices. Our first prototype is built upon DIMPLE [6]; an implementation[2] that can render haptics (3D haptic rendering, physical dynamics and collision detection, plus soft-realtime graphics) and audio separately. DIMPLE allocates demanding resources and timing accuracy to the haptics part and enable an asynchronous, event-based communication protocol based on the Open Sound Control[3] (OSC) between these processes.

IMPLEMENTATION AND PROTOTYPES
Both prototypes are built upon *nodes*[4], which enable us to construct dynamic audio environments akin to a scene graph. The nodes can also be interfaced to the external physical world via *terminals* or *ports* [1]. As a haptic experience, a bidirectional port makes us feel a grounded resistance and feedback, whereas a terminal can be used to supply non-grounded feedback (such as impulses and vibration).

A three-port K-node and an abstract dynamic audio environment are depicted in Fig. 1. This structure can be, for example, a part of a taut string terminated by an impedance Z_1. The force applied to a port (K-port 3 in the figure) is distributed in the environment based on the total impedance of the haptic device itself plus other individual branches (string and termination). The junction velocity V_J provides feedback immediately, as well as when the reflections from the terminations arrive. Alternatively, a force F_ext can be applied to a terminal for nongrounded interaction. The unit

[1] `http://puredata.info`
[2] `http://www.idmil.org/software/dimple`
[3] `http://opensoundcontrol.org`
[4] Technically, there are two types of nodes: the *wave and K-nodes*, but here we focus on the K-nodes only [1].

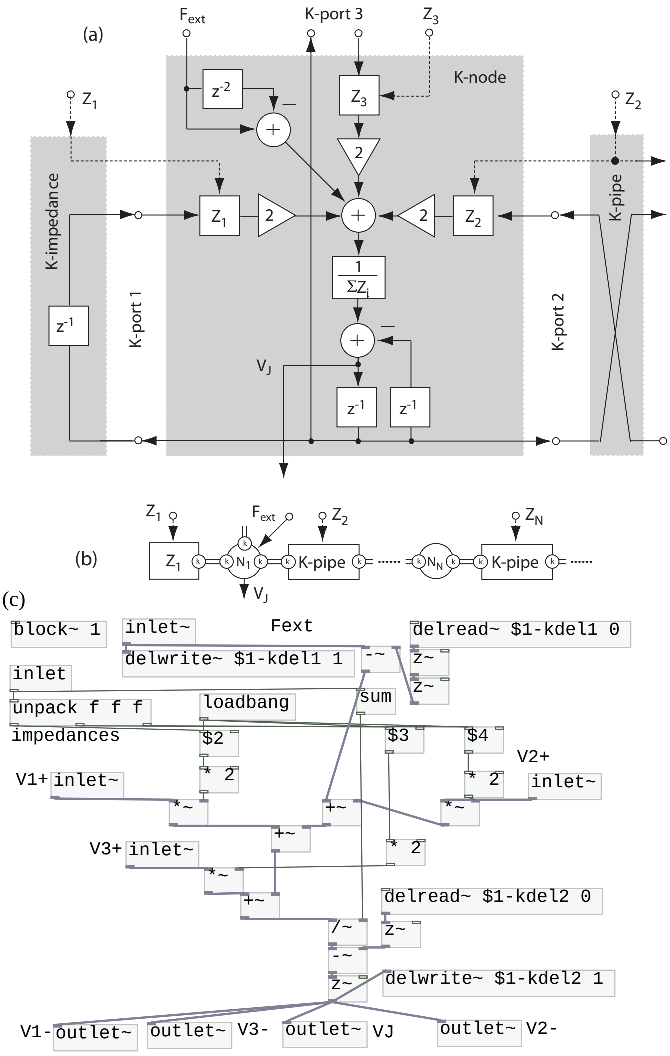

Figure 1. (a) A 3-port K-node with port impedances Z_i, after [5]. A haptic device that can be described by an impedance can be connected to a K-port. Otherwise, a force F_ext can be applied to a terminal to calculate the total velocity V_J. (b) Abstract representation of the K-node in (a). (c) PD implementation of the 3-port K-node in (a). Notice the block-size of one.

delays around V_J implement finite-difference approximation of the ideal wave equation. See [1] for further details.

Recently we have ported the nodes and some other BBPM elements to PD (see Fig. 1(c)). We run the models with minimum audio block-size for bidirectional interaction between the blocks. Albeit inefficient, this heavy duty computation is within the capabilities of current computer systems. We alter the signal propagation by supplying different impedance values, or by dynamically adding new ports or nodes with the internal messaging mechanism of PD.

In our first prototype we extend DIMPLE's OSC namespace to initialize and communicate with our BBPM via a port within PD, as illustrated in Fig. 2. The interface between DIMPLE and the haptic device is well-described in [6], we create a default BBPM with the following command: `/object/bbpm/create myModel`.

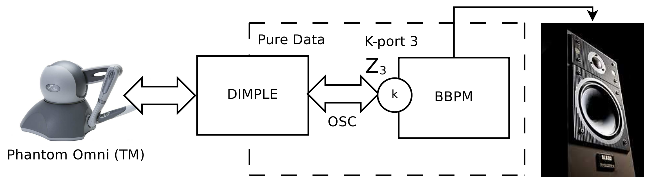

Figure 2. Our BBPM system linked with an impedance-based haptic device via a port (K-port 3 of Fig. 1).

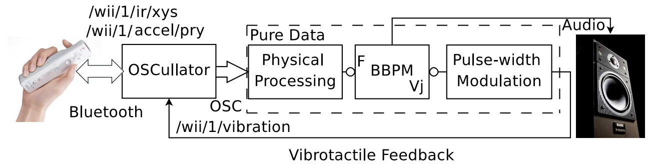

Figure 3. Our BBPM system linked with a Wiimote for simplistic audio-tactile interaction via a terminal force (F_ext in Fig. 1).

The default BBPM thus created is a 3-port K-node previously depicted in Fig. 1. Note that we need to convert the OSC control stream to an audio stream before applying it to a port. Each port except the one that interfaces DIMPLE via OSC can be nested. Port impedances may be changed with messages within PD in runtime.

Our second prototype, illustrated in Fig. 3, couples the Wiimote - the Bluetooth-based controller of the Nintendo's Wii console - to a terminal of our BBPM. We first convert the orientation, position, and the acceleration of the Wiimote (tracked by the OSCulator[5]) to the terminal force F_ext and apply it to the BBPM. We then modulate the only parameter available, namely the pulse-width of the signal feeding the vibrotactile motor by the calculated junction velocity for a nongrounded feedback substituting the first prototype.

REFERENCES

1. R. Rabenstein, S. Petrausch, A. Sarti, G. De Sanctis, C. Erkut, and M. Karjalainen, "Block-based physical modeling for digital sound synthesis," *IEEE Sig. Proc. Mag.*, vol. 24, no. 2, pp. 42–54, 2007.

2. V. Hayward and K. MacLean, "Do it yourself haptics: Part I," *IEEE Robot. Automat. Mag.*, vol. 14, no. 4, pp. 88–104, 2007.

3. D. DiFilippo and D. Pai, "The AHI: An audio and haptic interface for contact interactions.," in *Proc. ACM Symp. UIST*, (San Diego, CA USA), pp. 149–158, 2000.

4. F. Avanzini and P. Crosato, "Integrating physically based sound models in a multimodal rendering architecture," *Computer Animation and Virtual Worlds*, vol. 17, pp. 411–419, July 2006.

5. M. Karjalainen and T. Mäki-Patola, "Physics-based modeling of musical instruments for interactive virtual reality," in *Proc. Int. Workshop on Multimedia Signal Processing*, (Siena, Italy), pp. 223–226, Sep 2004.

6. S. Sinclair and M. M. Wanderley, "Using PureData to control a haptically-enabled virtual environment," in *Proc. PureData Conv.*, (Montreal, Canada), Aug 2007.

[5] `http://www.osculator.net`