Workshop on Haptic Interaction Design for Musical Applications Using Physical Models

ABSTRACT

A workshop for NIME 2013 is proposed on haptic interaction design for musical applications. The focus lies primarily on using physical models for programming haptic force feedback. During the workshop, participants will program the FireFader haptic device using the Synth-A-Modeler physical modeling framework, which combines elements from digital waveguide, mass-interaction, and modal synthesis. The source code for Synth-A-Modeler will be officially released to the open-source community at the NIME 2013 conference.

Keywords

Novel musical instruments, haptic and force-feedback devices, haptic interaction design, FireFader, Synth-A-Modeler

1. INTRODUCTION

1.1 Background

While traditional musical instruments provide haptic touch-oriented feedback, this kind of feedback is lacking in many digital musical instruments. This is one reason why the community is generally interested in endowing new digital musical instruments with haptic force feedback. In addition, active haptic feedback has some interesting properties that passive systems do not have. For instance, haptic feedback can enable musicians to make gestures that would otherwise be difficult or impossible, and it can assist musicians in playing more accurately.

Considerable research has been carried out on force feedback for musical systems, for instance for rendering forces for interacting with virtual acoustic musical instruments [6]. However, besides merely imitating pre-existing musical instruments, new virtual instruments can be designed with a computer by simulating the acoustics of hypothetical situations, creating a "metaphorisation of real instruments [5, 4]." For example, the GENESIS software package allows users to design physical models using a *modular* approach. In other words, a user specifies the physical parameters of a network of virtual mechanical elements. Then, the digital signal processing (DSP) algorithm for simulating the system is implemented automatically to compute the haptic force feedback and sound synthesis signals [5, 4].

NIME'13, May 27 – 30, 2013, KAIST, Daejeon, Korea.

1.2 Necessity of Programming using Physical Models

Prior work emphasized the value of programming haptic feedback using physical models [5, 4]. In other words, this work suggested that it was a *sufficient* design approach to prototype musical force feedback using physical models; however, it is also possible to program haptic feedback using a signal processing approach, in which the haptic feedback force is simply a function of the sensed variables [11].

However, in fact it is actually *necessary* to program haptic force feedback using physical models to guarantee certain stability properties, in particular for portability to other haptic devices and user hand impedances (i.e. hand stiffnesses, hand masses, etc.) [1]. The work provides a mathematical explanation why designing haptic force feedback using physical models appears to be the best approach: even though one could design stable and valid force feedback laws using a signal processing approach, one can do so equally well using physical models, which simplify programming because of the ease of proving the feedback controller is energy-conserving. The parameterization with physical parameters further enables a physical interpretation of the feedback controller, and it promotes portability to other haptic devices and user hand impedances.

2. CURRENT PLATFORM

2.1 Iterative Development

We have now taught a laboratory exercise in a musical instrument design course for three consecutive years. The exercise was originally based on an exercise by Bill Verplank [11], which was based on an earlier exercise by Okamura et al. [8]; however, each year, we have made changes to the platform in one way or another while evaluating how well students were able to design prototypes with it.

The first major change was programming the force feedback using physical models rather than a using signal processing approach. Informal testing suggested that, rather than students getting distracted by attempting to write error-free code in C, students were more successful at prototyping new models, and in general were able to program more complex designs using physical models than by programming "manually"/"by hand" in C.

The second major change was in improving the efficiency of the physical models. Originally, the physical models were implemented using primitives in Pure Data (pd), but we found that in order to achieve maximum computational power, it was necessary to compile the physical models directly into binary code. To achieve this, we started to use the Synth-A-Modeler physical modeling compiler.

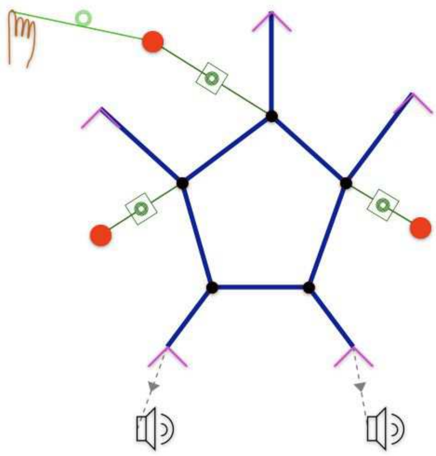

Figure 1: *Pentagonal percussion*—an example model for Synth-A-Modeler

2.2 Open-Source Software: Synth-A-Modeler

Synth-A-Modeler allows a user to specify a network of virtual mechanical elements and their physical parameters, which can then be compiled into a Faust DSP file [9]. In turn, the Faust DSP file can be compiled into a wide variety of targets, including Max/MSP, Pure Data (pd), and SuperCollider externals, as well as VST plug-ins and even generic Jack audio applications. In the workshop, Synth-A-Modeler is used to generate a modified generic Jack audio application, which can also communicate with a haptic device via USB.

Besides compiling into Faust DSP code, Synth-A-Modeler also differs from prior physical modeling sound synthesis environments in that it, for the first time, allows users to interconnect physical elements from the digital waveguide, mass-interaction, and modal synthesis schools of physical modeling [3]. This provides users with maximum flexibility but also optionally increased computational efficiency, for example via digital waveguide technology [10]. For example, consider the model depicted in Figure 1, which specifies a virtual membrane made of five serially connected digital waveguides that are rigidly terminated via five outwardly oriented waveguides. Two snare masses (in red) bounce on top of the leftmost and rightmost waveguide junctions, while a user can interact with the membrane by touching the uppermost red "drumstick" mass against a waveguide junction.

2.3 Open-Source Hardware: The FireFader

Other researchers have designed single degree-of-freedom devices for haptics education. However, these devices have either not been designed using off-the-shelf parts [11, 8], and/or they have required hand winding actuator coils and been subject to falling apart, which is not feasible for in a workshop environment [7].

Hence in the workshop, the FireFader device will be used. The FireFader is a low-cost haptic force-feedback device that is based around motorized faders, as found in some high-end mixing consoles. The motorized faders are interfaced with an Arduino, which allows the faders to be controlled from a laptop computer over a USB connection [2]. A FireFader with two motorized faders is shown in Figure 2.

3. WORKSHOP

We now describe the format of the workshop proposed for NIME 2013:

Figure 2: FireFader design incorporating two motorized faders

3.1 Format

- First half hour: The authors will present a lecture to explain the virtual mechanical elements available in Synth-A-Modeler. Example models will be employed to promote the workshop participants' understanding.
- Second half hour: The authors will lead the workshop participants through installation of the Faust DSP programming language and Synth-A-Modeler.
- Second hour: Participants will start working through a series of laboratory exercises. The authors will visit the participants individually or in small groups to answer questions.
- Third hour: Each participant will design a novel audio-haptic interaction by desiging a new Synth-A-Modeler model. At the close of the third hour, participants will share their designs with each other and the workshop organizers will present some closing words about additional features of the platform, such as compiling models into Max/MSP, Pure Data (pd), or SuperCollider externals. Participant will be encouraged to follow up at home and/or ask questions via a Google group.

3.2 Cost

The workshop organizers do not require a fee for any participant who does not wish to take any hardware home. However, we believe that most participants will want to take a FireFader home. To reduce the cost of the workshop, FireFaders will be provided that have only a single degree of freedom (i.e. have only one fader).

Each participant who wishes to take his or her FireFader hardware home should be prepared to pay the workshop organizers $125 USD in cash with exact change (or 96 Euros in cash or 150,000 KRW in cash with exact change). This fee only reimburses for the cost of the parts.

The total number of kits will be limited to 15. This means that more than 15 participants should be allowed to enroll ONLY if they are willing to share kits.

4. CONCLUSIONS

Following the workshop, participants can work further on their projects. The workshop organizers will continue to provide support via the project website and the Google group.

This workshop with its unique focus on physical modeling has never been taught at NIME before, and we believe that participants will come away from the workshop with a new understanding of haptic interaction design and modular physical modeling for sound synthesis. We would like to note that the quality of the audio from the demonstration models in Synth-A-Modeler is fully what one should expect from high-quality physical models computed using modern DSP techniques, which will appeal to the NIME community. The focus of the workshop lies more in musical instruments than in mechanical design applications of haptics. Some of the twenty example physical models that will be given to workshop participants are depicted in the appendix. The source code for Synth-A-Modeler will be officially released to the open-source community at the NIME 2013 conference.

4.1 Acknowledgments

There are many people who we would like to acknowledge, but their names have been removed until the camera-ready submission for the purposes of anonymization.

5. REFERENCES

[1] E. Berdahl, J.-L. Florens, and C. Cadoz. Using physical models is necessary to guarantee stable analog haptic feedback for any user and haptic device. In *Proceedings of the 8th Sound and Music Computing Conference*, Padua, Italy, July 6-9 2011.

[2] E. Berdahl and A. Kontogeorgakopoulos. The FireFader Design: Simple, open-source, and reconfigurable haptics for musicians. In *Proceedings of the 9th Sound and Music Computing Conference*, pages 90–98, Copenhagen, Denmark, July 11-14 2012.

[3] E. Berdahl and J. Smith III. An introduction to the Synth-A-Modeler compiler: Modular and open-source sound synthesis using physical models. In *Proceedings of the Linux Audio Conference*, Stanford, CA, USA, April 2012.

[4] C. Cadoz, A. Luciani, and J.-L. Florens. CORDIS-ANIMA: A modeling and simulation system for sound and image synthesis—The general formalism. *Computer Music Journal*, 17(1):19–29, Spring 1993.

[5] N. Castagne and C. Cadoz. Creating music by means of 'physical thinking': The musician oriented Genesis environment. In *Proc. 5th Internat'l Conference on Digital Audio Effects*, pages 169–174, Hamburg, Germany, Sept. 2002.

[6] J.-L. Florens, C. Cadoz, and A. Luciani. A real-time workstation for physical model of multi-sensorial and gesturally controlled instrument. In *Proceedings of the International Computer Music Conference*, pages 518–526, Ann Arbor, MI, USA, July 1998.

[7] B. Gillespie. Haptic interface for hands-on instruction in system dynamics and embedded control. In *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 410–415, March 22-23 2003.

[8] A. Okamura, C. Richard, and M. Cutkosky. Feeling is believing: Using a force-feedback joystick to teach dynamic systems. *ASEE Journal of Engineering Education*, 92(3):345–349, 2002.

[9] Y. Orlarey, D. Fober, and S. Letz. *New Computational Paradigms for Computer Music*, chapter FAUST: An Efficient Functional Approach to DSP Programming. Edition Delatour, Sampzon, France, 2009.

[10] J. O. Smith. *Physical Audio Signal Processing: For Virtual Musical Instruments and Audio Effects*. W3K Publishing, `http://ccrma.stanford.edu/~jos/pasp/`, December 2010.

[11] B. Verplank. Haptic music exercises. In *Proc. of the Int. Conf. on New Interfaces for Musical Expression*, pages 256–257, Vancouver, BC, Canada, 2005.

APPENDIX

A. MORE EXAMPLE MODELS