Practical Hardware and Algorithms for Creating Haptic Musical Instruments

Edgar Berdahl
CCRMA/Stanford University
eberdahl@ccrma.stanford.edu

Hans-Christoph Steiner
ITP/NYU
hans@at.or.at

Collin Oldham
CCRMA/Stanford University
coldham@ccrma.stanford.edu

ABSTRACT

The music community has long had a strong interest in haptic technology. Recently, more effort has been put into making it more and more accessible to instrument designers. This paper covers some of these technologies with the aim of helping instrument designers add haptic feedback to their instruments. We begin by giving a brief overview of practical actuators. Next, we compare and contrast using embedded microcontrollers versus general purpose computers as controllers. Along the way, we mention some common software environments for implementing control algorithms. Then we discuss the fundamental haptic control algorithms as well as some more complex ones. Finally, we present two practical and effective haptic musical instruments: the haptic drum and the Cellomobo.

Keywords

haptic, actuator, practical, immersion, embedded, sampling rate, woofer, haptic drum, Cellomobo

1. INTRODUCTION

A haptic musical instrument consists of actuators that exert forces on the musician, sensors that detect the gestures of the musician, an algorithm that determines what forces to exert on the musician, and a controller that runs the algorithm and interfaces with the sensors and actuators. The instrument often synthesizes sound as well. Figure 1 illustrates how the musician is included in the haptic feedback loop.

NIME08, Genova, Italy

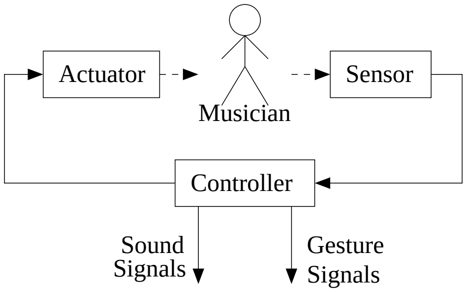

Figure 1: A musician interacting with a haptic musical instrument

There has been a wide array of research into haptics over the past decades, the vast majority taking place in specialized research labs with elaborate and custom equipment. Haptic feedback plays a key role in playing traditional instruments, and with the push to further develop electronic instruments, musicians have begun integrating haptic feedback into electronic instruments.

Recently, a number of developments have opened up haptic exploration to projects with smaller budgets and more common facilities. Additionally, as it becomes easier to access haptics equipment, it becomes possible to create haptics platforms oriented to musical instrument designers. This is especially interesting to designers looking to create their own instruments, since it means that they can design and employ useful haptic feedback in their own instruments.

2. ACTUATORS

Actuators form the core of any haptic musical instrument. The ideal actuator is linear and time invariant (LTI), has infinite bandwidth, can render arbitrarily large forces, and is accompanied by an LTI sensor with infinite resolution. In practice, the actuator usually limits the performance of haptic feedback in a haptic musical instrument. One effective design approach is to choose the actuator so that it directly complements the metaphor of the target haptic musical instrument. For instance, for a haptic drum, use a woofer to mimic a vibrating drum membrane.

2.1 Vibrotactile Actuators

Marshall and Wanderley [23] and Hayward and MacLean [20] provide good overviews of some actuators, so here we cover only the most effective and practical actuators for musical instrument designers.

2.1.1 Vibrating Motors

Vibrating motors are the most common haptic actuators. They are widely used in mobile phones and other communications devices. They are built using a motor with an unbalanced weight attached to the spindle. They are almost always used to generate a fixed frequency vibration, but some variation is possible. They are cheap, simple, and easy to obtain, but they have a slow ramp up time, which limits their application.

2.1.2 Tactors

Tactors are specialized motors that produce vibrations in a frequency range appropriate for sensing by the skin. They are included in devices like the iFeel mice. Immersion builds their tactors using "Inertial Harmonic Drive", which basically means a motor with a very small gear ratio whose spindle is attached to a surface by a somewhat flexible nylon linkage. The motor yanks on the linkage to generate a pulse. Another type of tactor is made using a piezoelectric element to actuate a plate under tension. It is also possible to build low-cost tactors using vibrating motors [17].

2.2 Force Feedback Actuators

In order to provide force feedback in practice, it is necessary to measure the behavior of the haptic device in the same dimension as it is actuated, making force feedback setups more complex.

2.2.1 Motorized Faders

Alps Electric Co. and other manufacturers make motorized faders designed for use in digital control surfaces. These faders consist of a belt motor drive attached to a linear slider potentiometer. The potentiometer can serve as the position sensor for the haptic feedback loop controlling the motor. Since the motor is relatively small, these faders cannot exert large forces, but they are cheap, pre-assembled and relatively easy to procure.

2.2.2 Servomotors with Optical Encoders

To produce relatively large forces, we have been using servomotors with built-in optical encoders that sense position [27]. We use the Reliance Electric ES364 servomotor with a peak-torque specification of 6.5 kg-cm and encoder resolution of 1000 pulses/rev (4000 counts).[1] An arm attached to the motor shaft makes it possible to interface the motor effectively with the hand. A force-sensitive resistor placed at the end of the shaft provides an additional sensed quantity useful in further fine-tuning the force feedback.

2.2.3 Woofers and Shakers

In contrast with rotational servomotors, woofers and shakers are *linear* actuators. As a consequence, the maximum displacements they provide are typically limited to a couple centimeters or less. Nevertheless, these actuators can be easily obtained at low-cost. Shakers are similar to woofers, but they have no cone for pushing air. Instead they mount to and shake a piece of furniture so that a listener can feel bass and infrasonic frequencies in music and movie soundtracks. The challenge in applying woofers and shakers effectively typically lies in integrating a sensor with the actuator.

Table 1: Approximate Actuator Costs in U.S. $

Device	**Price**
Vibrating motor	$1-$20
Tactor	$5-$200
Alps motorized fader	$30
Woofer/shaker	$40
Servomotor with encoder	$400
Novint Falcon	$200
SensAble Omni	$1000

2.2.4 Multi-DOF Haptic Devices

Commercial robotic arms like the 6DOF[2] SensAble Phantom have been available for a number of years now. They are typically designed to be held in the hand like a pen. They have traditionally been expensive and relatively rare; however, advancements in teleoperation and minimally-invasive surgery in particular have driven production costs per unit down significantly so that the Phantom Omni can be obtained for $1000.

The Novint Falcon is a more limited 3DOF haptic device that is designed for gaming. While it does not provide the flexibility or fidelity of the cheapest Phantom, it is available for less than $200.

3. CONTROLLERS

To provide force feedback, a control loop is usually called every $1/f_S$ seconds, where f_S is the sampling rate. This control loop reads inputs from the sensors, computes appropriate outputs to the actuators, and then immediately sends the outputs to the actuators. In order to have a responsive haptic musical instrument, the controller must be quick. In other words, the system delay (also known as input-output delay) should be short, and the sampling rate should be high. For most operating systems, these requirements are mutually exclusive, so in the following sections, we consider common control hardware implementations.

The sampling rate is an important factor. Typical haptics applications do not require sampling rates as high as audio. For instance, the CHAI 3D haptics framework does not support sampling rates above 1kHz for most devices [8]. However, some haptic musical instruments send audio signals through the feedback loop. The human range of hearing spans roughly 20Hz to 20kHz. According to the Nyquist-Shannon sampling theorem, the sampling rate must be at least 40kHz so the whole bandwidth that humans hear can be sampled and reconstructed within the feedback loop.

Haptic musical instruments taking full advantage of feeding aurally-relevant acoustic signals back through the haptic device must run at much higher sampling rates on the order of 40kHz. It is true that these higher frequencies are very poorly sensed by the human tactile system, but in a bowed string experiment, users reported that the system nevertheless felt much more real when the haptic sampling rate was 44kHz instead of 3kHz. They made comments regarding the "strong presence of the string in the hand," "the string in the fingers," and "the string is really here" [22].

---

[1] While the ES364 is now out of production, Applied Motion sells the comparable VL23-030D with an optical encoder for $400. It provides a maximum peak torque of 5.9kg-cm. This type of motor can be obtained surplus for prices as low as $7 each.

[2] six degrees of freedom

4. EMBEDDED MICROCONTROLLERS

Embedded microcontrollers can be run without any operating system or extraneous processes, which might interfere with the control loop timing. In addition, they are small, allowing them to be easily embedded within musical instruments, and they can be configured to interface with a wide variety of sensors and actuators. Atmel processor-based microcontrollers such as the AVR [5] and especially the Arduino [7] have recently become popular in computer music.

Note that these microcontrollers do not natively support floating-point calculations.[3] This is generally of no concern for simple algorithms, but more complex algorithms become much more difficult to implement without loss of fidelity.

4.1 Generic Programming Tools

Sometimes it is most convenient to program the control loop directly using generic tools. Microcontroller libraries such as AVR-lib make reading data from the sensors and writing data to the actuators straight-forward [5]. For teaching purposes, we use a combination of the AVRMini Atmel-based microcontroller board, the spyglass unit for producing debugging output, and the AVR motor controller board [3].

4.2 Immersion and USB PID

Immersion, Inc. sells a number of tools to make designing haptic feedback easier. Immersion devices use “effects”, which are built upon wavetables and envelopes and are handled by embedded microcontrollers. These effects can either be linked directly to button and position data using the microcontroller, or they can be controlled by the host computer via USB. The latency and jitter of USB are too high to handle the feedback loop,[4] so the microcontroller maintains the feedback loop. In Immersion-compliant devices, the feedback loop controlling the motors probably runs at 1kHz or faster.[5] The data sent over USB is used purely for configuring and triggering the microcontroller.

While a number of Immersion’s haptic devices are not easily procured, such as the tools for the medical and automotive industries, the tools aimed at video game development are practical for creating haptic feedback in musical applications. Joysticks and steering wheels provide kinesthetic and vibrotactile feedback using motors and position sensors; mice and gamepads provide vibrotactile feedback using tactors and vibrating motors.

4.2.1 USB Physical Interface Devices

Immersion, Inc. has worked to get their protocol into the USB Human Interface Devices (HID) [11] standard in a new subsection called PID (Physical Interface Devices) [12]. To program USB PID devices, each operating system has its own API: Apple has the *HID Manager* and *ForceFeedback* APIs, Microsoft has the *DDK HID* and Immersion APIs, and the Linux kernel has the *iforce* module and the *libff* API.

4.2.2 Immersion Studio

Immersion Studio, proprietary software only for Windows, is required to create and edit Immersion effects. The available effects are classified by Immersion thusly: Vibrational (Periodic), Positional (Texture, Enclosure, Ellipse, Spring, Grid), Directional (Constant, Ramp), and Resistive (Damper, Friction, Inertia) [1]. *Immersion Studio* allows designers to experiment with the set of effects and build them into an object that can be integrated into and triggered within a program. It is possible to create more elaborate compound effects by combining effects with waveforms and envelopes into an object that can be triggered as a single unit.

Immersion also produces more specialized versions of its *Studio* program for other markets, including medical and automotive applications. Additionally, they have the *VibeTonz* SDK for controlling the vibrating motor in some mobile devices and the *VirutalHand* SDK for controlling their glove systems. In general, this special software and equipment is targeted at specific markets and is not easy to obtain.

5. GENERAL PURPOSE COMPUTERS

In contrast with the aforementioned microcontrollers, general purpose computers are much faster and support native floating-point calculations. However, general purpose computers face a considerable drawback when controlling haptic feedback: the operating system schedulers, the bus systems, and device interface protocols can interfere with the ideally deterministic timing of the control loop. Using an RS232 serial port directly can help, but the maximum sampling rate will still be limited by the scheduler.

5.1 DIMPLE

Allowing musical instrument designers to incorporate a wide range of haptic behaviors into an instrument [25], DIMPLE takes full advantage of the CHAI 3D [8] and the Open Dynamics Engine (ODE) [10] libraries. ODE models the state of the virtual world, and CHAI 3D renders visual feedback and mediates the link between the virtual and the haptic worlds. The CHAI 3D library is compatible with Windows and GNU/Linux, and it supports a wide variety of haptic interfaces including the SensAble devices and the Novint Falcon. With the SensAble Phantom Omni, the maximum sampling rate is 1kHz, which limits haptic interaction at audio frequencies. The most recent release of DIMPLE incorporates a method for sending downsampled audio-frequency data to the actuators, but the delay, which is probably longer than 5ms,[6] prevents practical implementation of high-bandwidth feedback control.

5.2 TFCS

In contrast, the open-source Toolbox for the Feedback Control of Sound (TFCS) facilitates the implementation of haptic algorithms with large feedback bandwidths when using general purpose computers [14]. Virtual musical instrument models are provided via the Synthesis Toolkit (STK). Since they are implemented efficiently using digital waveguide technology, they can operate in synchrony with the haptic device at sampling rates as high as 40kHz with less than one sample of delay. The TFCS ensures that the con-

---

[3] They emulate floating-point calculations using integer arithmetic, which is too slow to be useful in most haptic algorithms.

[4] USB HID devices usually communicate with the host computer every 8-10ms; some devices can communicate faster, up to 1ms intervals.

[5] The manufacturers do not publish these rates.

[6] This theoretical lower limit has been derived during personal communication with Stephen Sinclair and still needs to be measured.

Table 2: Control Hardware

Control hardware	Maximum sampling rate	Approximate minimum delay	Native floating point
ATMEL-based	≈ 20kHz	$\doteq 50\mu$s	N
DIMPLE	1kHz	< 1ms	Y
TFCS	40kHz	$\approx 20\mu$s	Y
ASP	96kHz typ.	10**ms typ.**	Y

trol loop is called regularly by using the Real-Time Application Interface (RTAI) for Linux [6] and the Linux Control and Measurement Device Interface (Comedi) [9]. In multiprocessor machines, the control loop runs isolated on one processor, while all other code is executed on the remaining processors.

5.3 Audio Signal Processing (ASP) Environments

Most general purpose computers also come equipped with sound interfaces, so designers should consider whether a sound interface can be used for implementing the control loop. However, sound interfaces are not designed for very low-latency applications. Besides employing block-based processing, sound interfaces use sigma delta modulator converters that add considerable system delay [13]. The smallest system delay we were able to achieve on a 4.4GHz dual core AMD-based machine[7] was 4ms, where $f_S = 96$kHz. Nevertheless, this hardware/software solution is acceptable for some kinds of instruments. For example, haptic instruments that respond slowly to the environment can be implemented without problems.

6. ALGORITHMS

6.1 Standard Haptic Algorithms

6.1.1 Spring

An actuator induces a force F on the haptic device. Most haptic devices measure the movement of the device in response as a displacement x. Hence, the most fundamental (i.e. memoryless) haptic algorithm for these devices implements a virtual spring with spring constant k.

$$F = -kx \tag{1}$$

The virtual spring in combination with the physical mass and damping of the haptic device forms a damped harmonic oscillator, which can be plucked or bowed. By obtaining estimates of the haptic device's velocity or acceleration, the device's damping and mass can be controlled analogously.

6.1.2 Wall

An algorithm similar to the spring implements a wall at $x = 0$:

$$F = -kx \cdot (x > 0) \tag{2}$$

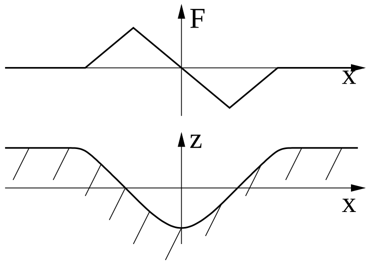

Figure 2: Force profile $F(x)$ (above) and terrain height profile $z(x)$ (below) for a simple detent.

Whenever the haptic device is pushed inside the virtual wall (i.e. $x > 0$), a spring force acts to push the device back out of the wall. So that the wall feels stiff, k should be chosen large. The maximum stiffness that a haptic device can render is governed by a fundamental limit, which is chiefly a function of the system delay, the sampling rate, and the internal physical damping of the device [19]. In general, more expensive haptic devices are required for rendering especially stiff virtual springs and walls.

6.1.3 Detents And Textures

Detents can help the musician orient himself or herself within the playing space of the instrument. Detents can be created even with 1DOF haptic devices. Figure 2 illustrates how to implement a simple detent. Near the origin, the force profile looks like that of a spring, while the force goes to zero when the position x moves further from the detent [27]. A simple potential energy argument implies that the force profile $F(x)$ is proportional to the derivative of the terrain height $z(x)$ (see Figure 2), allowing arbitrary terrains and textures to be created.

6.1.4 Event-Based Haptics

Another effective algorithm uses the sensors to detect certain events. When an event occurs, a stored waveform is sent to the actuators. A common example in gaming is sending a recoil force waveform to the actuators when the user fires a weapon. Since virtual walls cannot be made infinitely stiff, some musical instrument designers may consider sending ticks or pulses to the haptic interface whenever the interface enters a virtual wall. This type of event-based feedback is known to improve the perception of hardness [21].

6.2 Algorithms Requiring High Sampling Rates

6.2.1 Virtual Instruments

Extensive studies on the physical modeling of acoustic musical instruments have led to the development of many different acoustic musical instrument models. One simple way to create a haptic musical instrument is to interface a haptic device with a virtual instrument according to the laws of physics [18]. For efficiency reasons, it is often convenient to run the haptic control loop at a standard haptic sampling rate, while the musical instrument model runs at a higher sampling rate to provide high-quality audio. For example, the Association pour la Création et la Recherche sur les Outils d'Expression (ACROE) often employs a haptic sampling rate of about 3kHz, while audio output is often synthesized at standard audio sampling rates, such as 44kHz. However,

---

[7] The machine was running the Planet CCRMA distribution of Fedora Core, which has a patched kernel allowing low-latency audio.

Figure 3: Haptic drum

ACROE sometimes employs their ERGOS device with dedicated DSP hardware to run both the haptic and audio loops at 44kHz in real-time [22].

6.2.2 Actively Controlled Acoustic Instruments

An actively controlled acoustic musical instrument is an acoustic musical instrument that is augmented with sensors, actuators, and a controller. These instruments can be considered a special case of haptic musical instruments where the interface is the entire acoustic instrument itself. For example, a monochord string can be plucked and bowed at various positions as usual, while its acoustic behavior is governed by the control hardware. Simple and appropriate control algorithms emulate passive networks of masses, springs, and dampers or implement self-sustaining oscillators [15].

7. EXAMPLES

7.1 Haptic Drum

The haptic drum is a haptic musical instrument that can be constructed out of components found in practically any computer music laboratory [2]. It employs an event-based haptics algorithm that is implemented using a woofer actuator, a general purpose computer, and an ASP environment.

The woofer actuator conforms to the metaphor of a vibrating drum membrane. A sunglass lens is attached rigidly to the cone but held away from the sensitive surround part by way of a toilet roll (see Figure 3). Whenever a drumstick strikes the sunglass lens, it makes a loud "crack" sound. A nearby microphone (not shown) provides an input signal to a sound interface. A Pure Data patch detects drumstick collisions by checking the threshold of the microphone signal envelope. Whenever a collision is detected, an exponentially-decaying pulse is sent to the woofer that effectively modifies the coefficient of restitution of the collision. The haptic drum can be configured to make it easier to play (one-handed) drum rolls. It also facilitates playing various "galloping" and "backwards" drum rolls, which are otherwise nearly impossible to play using one hand [16]. If instead a ping pong ball is placed on the lens, and if the lens is driven sinusoidally, various period-doubling and apparently chaotic effects may be observed.

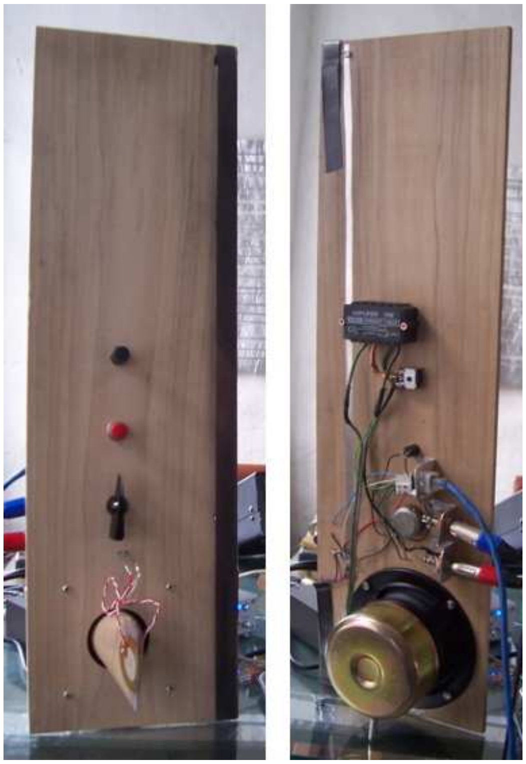

Figure 4: Cellomobo front (left) and back (right)

7.2 Cellomobo

The Cellomobo is an instrument allowing the musician to bow a virtual string using a haptic interface [4]. The length of a the string is adjusted by a resistive ribbon controller (see Figure 4, left). The vibrating string element consists of a piezoelectric disc pickup (see Figure 4, bottom left), which is mounted upon a shaker (see Figure 4, bottom). The haptic feedback and sound synthesis algorithms run at the audio rate in Pure Data.

Figure 5 shows a diagram of the the Cellomobo's combined haptic feedback/sound synthesis engine. The dotted box encloses the digital waveguide model of a lightly damped vibrating string. N/f_S is the period of the note being played. The internal feedback loop gain g is between 0.9 and 0.999 and is controlled by a knob. $H_{LP}(z)$ is a lowpass filter causing the higher partials to decay more quickly [26]. The outer feedback loop is closed around the shaker and piezoelectric pickup, which provides the excitation input to the instrument. $H_{2LP}(z)$ is a second order lowpass filter to remove upper partials from the feedback loop. The cut-off frequency of the filter is controlled by left hand finger pressure, to give the musician control of tone color. Before the output signal reaches the actuator, a hard clipping nonlinearity clips off the tops of the wave form. This gives the haptic signal more of a square shape, causing the bow to release from the bowing surface more easily.

The novel addition of the inner feedback loop is nonphysical, but it allows the instrument to be less sensitive to the dynamics of the sensor and actuator. This structure en-

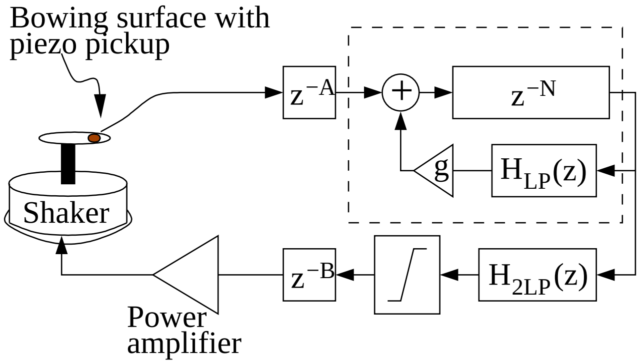

Figure 5: The Cellomobo block diagram

hances the playability of the instrument and differentiates the Cellomobo from previous research efforts [22]. In fact, the behavior is so robust, that the instrument functions despite the large ASP system delay $(A+B)/f_S \doteq$20ms. Note that this delay is an order of magnitude longer than the period of the highest note that can be played on the instrument, which is about 1ms.

8. CONCLUSIONS

Making haptic musical instruments is not so difficult given some forethought and knowledge about the field! Incorporating haptic feedback is also often worth the effort—haptic feedback has been shown to improve the user's impression of playing a haptic musical instrument [22]. Haptic feedback has been informally found to make it easier for users to play various types of drum rolls [16]. Finally, haptic feedback has been further shown to improve the accuracy of musicians playing a haptic musical instrument [24].

In this paper, we presented ideas on how to practically implement such instruments given today's technology. We hope our efforts will help make haptic technologies more accessible to designers and musicians. We expect more superior haptic technologies to become even more accessible as other fields drive haptic device development.

9. ACKNOWLEDGEMENTS

We would like to thank all of the people at or from CCRMA who have helped us and inspired us to study haptics: Bill Verplank, Julius O. Smith III, Günter Niemeyer, Chris Chafe, Sile O'Modhrain, Brent Gillespie, and Charles Nichols.

10. REFERENCES

[1] Immersion fundamentals. `http://www.immersion.com/developer/downloads/ImmFundamentals/HTML/ImmFundamentals.htm`.

[2] `http://ccrma.stanford.edu/~eberdahl/Projects/HapticDrum`.

[3] `http://cm-wiki.stanford.edu/wiki/AVR`.

[4] `http://homepage.mac.com/coldham/klang/cellomobo.html`.

[5] `http://hubbard.engr.scu.edu/embedded/`.

[6] `https://www.rtai.org/`.

[7] `http://www.arduino.cc/`.

[8] `http://www.chai3d.org/`.

[9] `http://www.comedi.org/`.

[10] `http://www.ode.org/`.

[11] USB HID. `http://www.usb.org/developers/hidpage/`.

[12] USB PID. `http://www.usb.org/developers/devclass_docs/pid1\_01.pdf`.

[13] M. Antila. Contemporary electronics solutions for active noise control. In *Proc. Int. Symposium on Active Noise and Vibration Control*, September 2004.

[14] E. Berdahl, N. Lee, G. Niemeyer, and J. O. Smith. Practical implementation of low-latency DSP for feedback control of sound in research contexts. In *Proc. of the Acoustics '08 Conference*, June 2008.

[15] E. Berdahl, G. Niemeyer, and J. O. Smith. Applications of passivity theory to the active control of acoustic musical instruments. In *Proc. of the Acoustics '08 Conference*, June 2008.

[16] E. Berdahl, B. Verplank, J. O. Smith, and G. Niemeyer. A physically-intuitive haptic drumstick. In *Proc. Internat'l Computer Music Conf.*, volume 1, pages 363–366, August 2007.

[17] A. Bloomfield and N. I. Badler. A low cost tactor suit for vibrotactile feedback. Technical Report 66, Center for Human Modeling and Simulation, University of Pennsylvania, 2003.

[18] N. Castagne and C. Cadoz. Creating music by means of 'physical thinking': The musician oriented genesis environment. In *Proc. of the Fifth Int. Conf. on Digital Audio Effects*, pages 169–174, 2002.

[19] N. Diolaiti, G. Niemeyer, F. Barbagli, J. K. Salisbury, and C. Melchiorri. The effect of quantization and coulomb friction on the stability of haptic rendering. In *Proc. of the First Joint Eurohaptics Conf. and Symp. on Haptic Interfaces*, pages 237–246, 2005.

[20] V. Hayward and K. MacLean. Do it yourself haptics, part 1. *IEEE Robotics and Automation Magazine*, 14(4):88–108, December 2007.

[21] K. Kuchenbecker, J. Fiene, and G. Niemeyer. Improving contact realism through event-based haptic feedback. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):219–230, March/April 2006.

[22] A. Luciani, J.-L. Florens, D. Couroussé, and C. Cadoz. Ergotic sounds. In *Proc. of the 4th Int. Conf. on Enactive Interfaces*, pages 373–376, November 2007.

[23] M. Marshall and M. Wanderley. Vibrotactile feedback in digital musical instruments. In *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression (NIME06)*, Paris, France, 2006.

[24] S. O'Modhrain and C. Chafe. Incorporating haptic feedback into interfaces for music applications. In *Proc. of ISORA, World Automation Conference*, 2000.

[25] S. Sinclair and M. Wanderley. Extending dimple: a rigid body haptic simulator for interactive control of sound. In *Proc. of 4th Int. Conf. on Enactive Interfaces*, November 2007.

[26] J. O. Smith. *Physical Audio Signal Processing: For Virtual Musical Instruments and Audio Effects.* http://ccrma.stanford.edu/˜jos/pasp/, 2007.

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