The FireFader Design: Simple, Open-Source, and Reconfigurable Haptics for Musicians

Edgar Berdahl
Audio Communication Group
Technical University of Berlin
Germany

Alexandros Kontogeorgakopoulos
Cardiff School of Art and Design
Cardiff, Wales
United Kingdom

ABSTRACT

The FireFader is a simple haptic force-feedback device that is optimized for introducing musician to haptics. It has one degree of freedom and is based upon a mass-produced linear potentiometer fader coupled to a DC motor, also known as a "motorized fader." Lights are connected in parallel with the motor to help visually communicate the strength of the force. Compatible with OS X, Linux, and Windows, the FireFader consists of only open-source hardware *and* software elements. Consequently, it is also relatively easy for users to re-purpose it into new projects involving varying kinds and numbers of motors and sensors.

An open-source device driver for the FireFader allows it to be linked to a laptop computer via USB so that the computer can perform the feedback control calculations. For example, the laptop can simulate the acoustics of a virtual musical instrument to calculate the motor force as a function of the fader position. The serial connection over USB causes delay of the control signal, but the serial connection facilitates easier programming via the laptop, and the force feedback can be disabled when the user is not touching the fader. Some new devices derived from the FireFader design are presented.

1. INTRODUCTION

1.1 Overview

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 feedback. Haptic feedback can take many forms [1], and one important form is *force feedback*, which allows a user to interact kinesthetically with a virtual mechanoacoustical system.

While considerable research has been carried out on force feedback for musical systems [2,3], it has not trickled down into systems that are widely available to musicians. One of the main reasons for this is that most prior force feedback systems were too expensive for most musicians. For example, the force-feedback devices from Ergos Technologies are designed with the highest quality musical applications in mind: the devices have a position resolution of $<1\mu m$, mechanical bandwidth of 20kHz, and maximum force of up to 200N. [1] We would love to teach workshops with them; however, they cost tens of thousands of Euros, so they are beyond the cost of most musicians' budgets [2].

Force-feedback devices from Sensable Technologies are less expensive. For example, the least expensive device from Sensable is the Phantom Omni, which costs about $1000 USD. However affordable this price may or may not be, the devices from Sensable are designed with medical applications in mind, such as surgery, and a musician on stage may not be so interested in sitting at a table and performing with a pen-like object.

For instructional purposes, several universities have made simpler haptic force-feedback devices that are less expensive [4]. The series of "Haptic Paddles" are single degree-of-freedom devices based upon a cable connection to an off-the-shelf DC motor. However, such designs are problematic because of the unreliable supply of surplus high-performance DC motors [5]. We ourselves contacted four companies selling DC motors with integrated sensors, and we could not find a low-price motor of this type that was always available new. Surplus motors can be obtained, for instance from stock of old hard disks [6]; however, removing the motors from the disk drives themselves requires some additional work. New, identical DC motors can be ordered from companies such as Maxon motors, but they tend to cost more than $100 or even $200 USD each when they include a position sensor. [2] In contrast, the iTouch device at the University of Michigan instead contains a voice-coil motor, which is hand-wound by students [5]. However, making a large number of devices is time intensive, and the part specifications are not available in an open-source hardware format.

In our own prior work, we have used the least expensive general-purpose force-feedback robotic arm called the NovInt Falcon. Over the past several years, its price has ranged between $100 USD and $250 USD. The Falcon is designed primarily for gaming, so it is inexpensive and is accessible to musicians [7]. Conveniently, it has an open-

[1] http://acroe.imag.fr/ergos-technologies
[2] http://www.maxonmotor.com/

source driver compatible with Mac OS X, Windows, and Linux [8]; however, the appearance and size of the device can be less appealing to musicians. Also in our own prior work, we have alternatively emphasized that radically different actuator configurations can be artistically interesting, such as the haptic drum, which uses a woofer to actuate a drumstick [9]; however, the Falcon's geometry cannot easily be modified because of the complex interconnections of the cables and other parts.

Standardized configurations are useful as they can be employed in and abstracted to a wider variety of applications, and indeed this consideration has influenced the FireFader's design.

1.2 Requirements

In consideration of prior work and our own experiences, we arrived at the following requirements for the device design. The device should

- implement standardized force-feedback control with collocated sensing,
- be relatively inexpensive yet still perform well enough for intriguing musical applications,
- have access to high-quality audio converters,
- be compatible with OS X, Linux, and Windows,
- be controllable via double-precision floating-point physical models [10],
- be easy to assemble,
- be reconfigurable,
- allow additional sensors to be incorporated with ease,
- inspire musicians with its design,
- be simple enough that students can understand and reconfigure it, and
- consist of open-source hardware *and* open-source software elements.

2. HARDWARE DESIGN

2.1 Motor

To keep the cost down, we decided to center the design about a motorized fader, which is a linear potentiometer coupled to a DC motor. Motorized faders are mass-produced by multiple companies, including ALPS Electric and Penny + Giles, so the competition can tend to drive down the price. Furthermore, motorized faders can be found in many audio mixing consoles, so most professional musicians are familiar with them already. For that reason, we thought that musicians might on average be more interested in force-feedback interaction with a motorized fader than with something more foreign-looking like a haptic paddle, and we have noticed this trend among our own students.

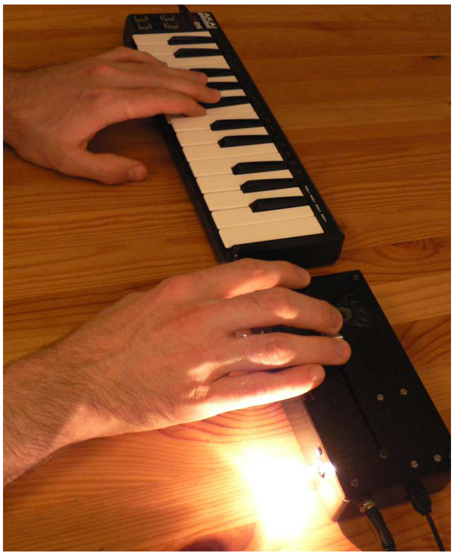

Figure 1. FireFader (below) with MIDI keyboard (above)

We compared several different faders, and we eventually decided to primarily use a motorized fader from ALPS, which currently can often be purchased at a price of about $25 USD each new.

We would like to note that the idea of using a motorized fader for new musical applications is not new. Bill Verplank has maintained a stock of them in the laboratory at CCRMA for several years, and other human-computer interface researchers have experimented with them [11], even for audio applications [12–14]. However, our design differs due to the availability of new open-source hardware, an expanded sensing design, and we agree with the philosophy that the user should be able to use powerful floating point computations for the feedback control, to prevent getting distracted by a need to write efficient firmware.

2.2 Concept

For musical applications, it could be interesting to combine the FireFader's force feedback with other more common user interfaces. In other words, building a keyboard with all force-feedback keys is very impressive [15]; however, one can still obtain interesting interactions with the combination of a single degree-of-freedom force-feedback device and a standard MIDI keyboard. Figure 1 shows a FireFader prototype (below) and a small MIDI keyboard (above).

2.3 Lights

While force feedback can easily change the nature of interaction from a performer's perspective, in many situations the audience might not perceive the presence of the force feedback. Thus, we thought the device might benefit from a method for communicating the force level to the audience, in line with the concept of multisensory simulation with coherent multimodal feedback [16]. A bright light helps suggest the metaphor of *fire*, suggesting power,

Figure 3. Specification image for top plate for one fader

excitement, and extraordinariness. This explains why we named the device the *Fire*Fader, which helps to further distinguish it from other user interfaces that lack active haptic feedback.

We employed a light to indicate the force level for each fader. Even if the audience would not realize that the lights communicated the force level, the lights could probably at least draw attention to the force itself. Furthermore, we thought this feature might seem exciting and inspiring to musicians, emphasizing the distinction between a common digital musical interface and an interface with force feedback.

We experimented with over 50 different lights in order to find a bright light that looks appealing over the motor driver's dynamic range. One good solution is a 5W Halogen lamp, which has a slightly yellowish tinge to it as seen in Figure 1. Other lights may be less delicate, such as a 20mm LED in series with a 1W 100 ohm resistor, or a 12V LED replacement lamp.

2.4 Enclosure

We wished to provide an open-source method for creating enclosures for the FireFader. Laser-cut parts are appealing due to low cost, ease of reconfigurability, and the wide variety of materials available for laser cutting. While it is possible to build entire enclosures using laser cutting methods, [3] these enclosures tend to involve a lot of screws and look unusual due to the inability to laser-cut threaded screw holes. For this reason, we decided to laser-cut only the top plate for the enclosure, including the long groove for the fader itself, and to fix the top plate to part of a standard plastic box. Figure 3 shows the open specification image for the single-fader top plate that matches the Strapubox 2003 SW bottom with outer dimensions 160mm by 83mm by 52mm. Using open-source software such as Inkscape, the top plate specification could easily be modified to fit another mass-produced box. The final enclosure design is shown in Figure 4, which incorporates two motorized faders.

2.5 Electronics

The electronics are based around the Arduino Nano 3.0 microcontroller, which plugs directly into the 2MOTOR

[3] http://support.ponoko.com/entries/20344437-laser-cut-project-box-tutorial

Figure 4. Final FireFader enclosure design incorporating two motorized faders

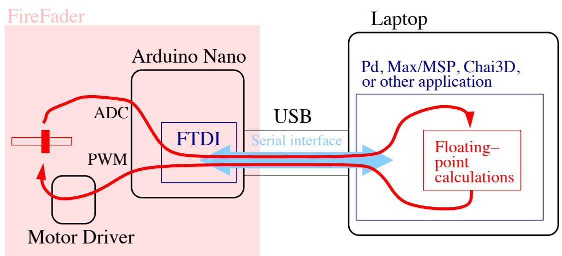

Figure 5. Signal flow for the control loop

H-bridge add-on board. The schematics for both of these boards are available on Gravitech's website. [4] The interconnecting wires for the remaining components are given in the schematic for the FireFader (see Figure 2).

3. SOFTWARE DESIGN

3.1 Signal Flow

The FireFader communicates with a laptop in order to gain access to fast real-time floating point computation and high-quality audio converters. The usage of the laptop also eases programming the FireFader because code efficiency is less of a concern.

The signal flow describing how the FireFader and laptop work together is shown in Figure 5. First the Arduino measures the position of the fader via the ADC. Then the Arduino uses its FTDI chip to send the measured position to the laptop over a serial interface provided by USB. The laptop runs an application to control the FireFader. Currently Max/MSP, Pure Data (pd), Chai3D, and a generic application are supported. This application receives the position of the fader and computes a reaction force. The application sends the reaction force over the serial interface via USB to the Arduino, which uses the 2MOTOR H-bridge board to exert the reaction force via the motor (see Figure 5).

3.2 Arduino Firmware

An Arduino firmware program runs on the Arduino to facilitate the signal flow shown in Figure 5. A main loop on

[4] http://www.gravitech.us/

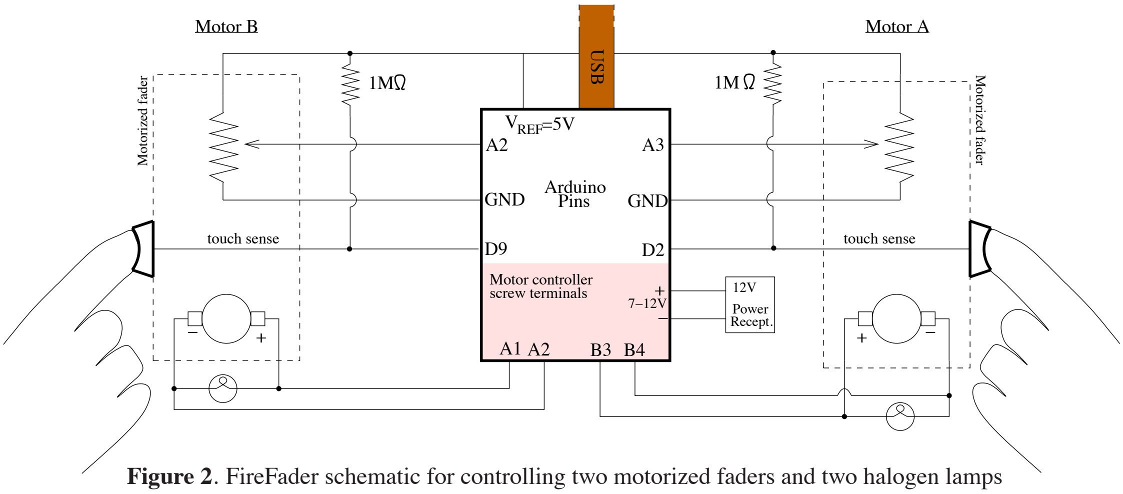

Figure 2. FireFader schematic for controlling two motorized faders and two halogen lamps

the Arduino repeats the following steps over and over. We describe them in detail here because the explanation could be helpful for users who want to modify the firmware when reconfiguring the device:

1. While input serial data from the laptop is available, read it in, and write the most recent force commands to the PWM pins for controlling the motors.
2. For each fader, sample its analog input multiple times and average the result to obtain an estimate of its position.
3. Sample the analog inputs for additional ADC pins, to be used with any generic sensors, while averaging the results.
4. Perform capacitive sensing for each of the faders.
5. Send the sentinel character 255 over the output serial interface to indicate that a new packet of information is available.
6. Send the position of each fader, clipping it to the range 0 to 254 to avoid confusion with the sentinel.
7. Send the values from the other sensors, clipping each value again to the range 0 to 254 to avoid confusion with the sentinel.
8. Send the low-pass filtered capacitive sensing value from each fader, clipping it to the range 0 to 254.
9. Wait for approximately 100 μs.

The wait statement helps prevent the serial bus from getting overrun with data. To slow down the firmware for debugging, the wait statement can be changed to values such as 100 ms or 1 s.

3.3 Host Software on Laptop

3.3.1 Custom Application

Physical models for controlling the FireFader can be generated using the Synth-A-Modeler software package [17]. It generates Faust DSP code that can be compiled into a custom application that accesses the FireFader directly via USB and produces output audio via Jack [18]. However, some users may prefer to employ the FireFader from within other common sound synthesis environments. For this reason, we have provided sample driver code for Max/MSP and Pure Data (pd).

3.3.2 Configuring the Feedback Control Delay

In order for the FireFader to work well, the feedback control delay needs to be short, yet each element in the feedback loop contributes some delay. We have optimized the delay of the elements in the loop as best possible subject to the low-cost design. The application running on the laptop contributes some delay, so we provide some tips here for reducing that delay.

In Max/MSP and Pure Data (pd), the position of the FireFader is received as an event. Thus, events need to be processed frequently, which can be configured by adjusting the scheduler. In Max/MSP 5 the scheduler interval can be set to 1ms in the Preferences window under "Scheduler." In Pure Data (pd), the scheduler interval can be set using the startup switch "-sleepgrain 1" although under some conditions better performance can be achieved using "-rt -sleepgrain 1" to also enable real-time scheduling if available on the operating system.

Finally, it is essential to set the default value for the FTDI chip's latency timer to 1ms. This cannot be set by the Arduino firmware, but it can be set via the FTDI driver on the laptop. Accordingly, it is essential to follow instructions [5] for setting the FTDI latency. Otherwise, the serial buffer on the FTDI chip on the Arduino may not be flushed as often, increasing the latency of the force feedback by as much as 16ms or more.

Under OS X with Max/MSP 5, we measured the round-trip control delay to be roughly 3.5ms on average. This is about the same as the latency for the open-source NovInt Falcon driver, which also uses a serial link over USB via an FTDI chip. Performance under Pure Data (pd) has been

[5] http://projectgus.com/2011/10/notes-on-ftdi-latency-with-arduino/

good on the Beagle Board xM but not as good under OS X. We are continuing to debug the performance in pd for OS X.

4. TOUCH SENSE ENABLE

If feedback control with a large gain is desired, which can be the case when modeling a very stiff spring or strong damper, then the force feedback can cause a haptic device to become unstable, especially if the delay around the control loop is significantly long. Instability is a potential problem because it can damage the device. However, a user generally provides a stabilizing influence when touching a device [19].

Thus, for devices such as the FireFader, which are subject to relatively long feedback control delays, it is a good idea to only enable the force feedback when the user is touching the fader knob. This "touch sense enable" feature allows for programming interactions with larger gains, because the feedback is disabled when the user releases the fader knob. The feature can be implemented via capacitive sensing of the "touch sense" pin that is electrically connected to the fader knob with a little help from the firmware and the 1MΩ pull-up resistor (see Figure 2).

5. TESTING

5.1 Instructional Prototypes

We evaluated the FireFader from an instructional perspective using five prototypes. Students in the Music 250A course at Stanford University used the prototypes to complete a musical design-oriented exercise in October 2011: https://ccrma.stanford.edu/wiki/250a_Haptics_Lab_2011

The laboratory exercise was motivated by a previous exercise by Bill Verplank [4] adapted for the new device. In the new exercise, students programmed by specifying physical models instead of writing plain C code [20]. The results indicated that students could program much more complex designs using physical modeling than when writing plain C code.

5.2 Example Projects

Example projects have provided more opportunity to refine the FireFader's design. Figure 6 shows the *Sound Flinger* by Chris Carlson, Eli Marschner, and Hunter McCurry. The four faders allow users to interact with a virtual mass sliding along the edges of a square, while the position of the mass set the angle at which an audio input was reproduced within a room [21].

For *PROJECT SQUEEZE*, Joel Sadler and Shruti Gupta constructed a single degree-of-freedom exoskeletal haptic pincher out of the FireFader parts [22]. The pincher was made primarily out of laser-cut acrylic, including a piece which links the fader handle with the thumb (see Figure 7).

Edgar Berdahl constructed *String-U-Topia* using an early FireFader prototype to implement a plucked interaction with a triplet of virtual strings [23].

Figure 6. The *Sound Flinger*

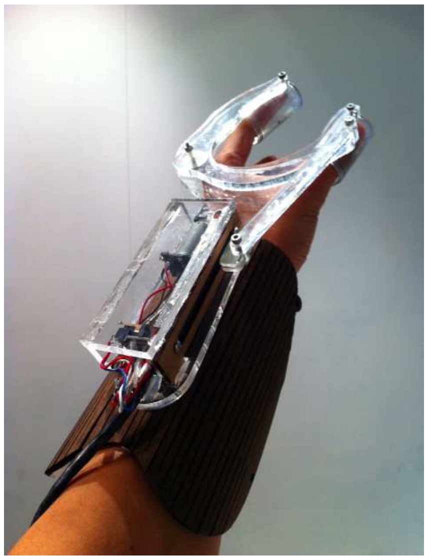

Figure 7. Exoskeletal haptic pincher from *PROJECT SQUEEZE*

The design of the light can also be important. We show a few frames from a video clip of an older prototype in Figure 9 to further illustrate the effect of the lighting.

6. CONCLUSIONS

The design of the FireFader has been an iterative process and was first announced in an abstract published by the Acoustical Society of America [24]. This paper serves to provide detail on the design, how musicians have been using it, and the full open-source disclosure of the hardware and software.

The FireFader hardware components, Arduino firmware, pictures, and drivers for Max/MSP, Pure Data, Chai 3D, and sample generic application can be found in the archive:

`https://ccrma.stanford.edu/~eberdahl/FireFader.zip`

The device is simple, open-source, and reconfigurable, so we hope that it will appeal to musicians as well as the broader DIY community. We have started a Google Group for providing support for the platform and generally trying to galvanize the DIY community:

`http://groups.google.com/group/open-source-haptics-for-artists`

We believe that the FireFader is the only completely open-source hardware and software system for building

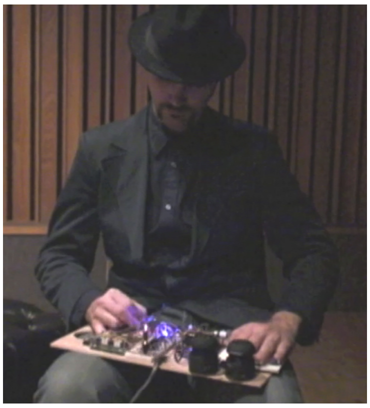

Figure 8. *String-U-Topia*

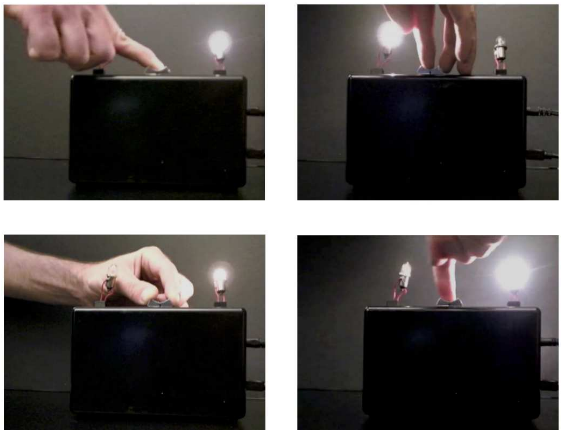

Figure 9. An older prototype of the FireFader at work

haptic musical instruments with force feedback. It is relatively inexpensive, with the total cost of the parts adding up to about \$150 USD for two faders. The capacitive sensing feature allows for the force-feedback to be enabled only when the user is touching the knob, allowing for implementing force feedback with relatively large control gains despite the relatively long feedback control delay of 3.5ms on average.

Finally, the FireFader is so small and convenient that we plan to teach workshops with it to introduce more members of the community to what force-feedback has to offer for them. In future work, we will create a library of physical models for controlling the FireFader to help teach users how to integrate force feedback into their own musical instrument designs.

A. ESTIMATING DOWNWARD PRESSURE

Adding motors is more expensive than adding sensors. In this appendix, we describe how to inexpensively sense the downward pressure that the user applies to the fader.

In fact, if one covers the fader knob with resistive fabric such as sold by EeonTex and presses on the knob [25], an analog signal related to the downward pressure is already

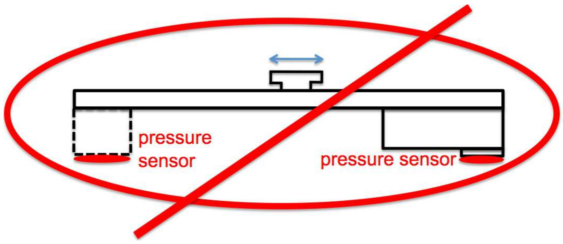

Figure 10. Motorized fader with FSRs under each end

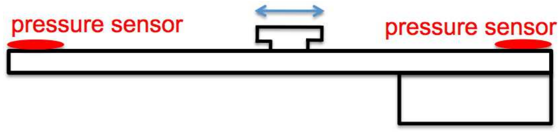

Figure 11. Motorized fader with FSRs placed *on top*

obtainable via the capacitive sensing circuit (see connections to pin D9 in Figure 2). However, we found this sensing approach to be noisy, and it only provided a small dynamic range.

We made further experiments using inexpensive force-sensing resistors (FSRs). At first we tried placing an FSR underneath each end of the motorized fader as shown in Figure 10, mounting the entire fader on top of the project box (not shown). The idea was to combine the sensed signals to estimate the downward pressure applied by the finger. However, we found that the measurements to be marginal, perhaps due to the compliance of the motor mounting.

Next we tried placing the FSRs *on top* of the motorized fader as shown in Figure 11, so that they could be sandwiched in between the motorized fader and the project box. We found this approach to work much better.

Figure 12 shows all of the sandwich elements, including also a white piece of compliant double-stick tape and a spring-loaded piece of acrylic to preload the FSR. When the user pushed downward on the fader knob, the fader pulled downward on the screw in the center (see arrow in Figure 12) to reduce the preload on the FSR. The orientation of the sandwich elements is shown in two profiles in Figure 13.

The dynamic range of the sensor signal needed to be increased in order to obtain a high-quality measurement. Figure 14 shows the circuit we used to obtain a gain of $1 + (Rf/Rg) = 8$. The trimpot *Rs* adjusted the offset voltage for the sensor signal, and it had a similar effect to changing the tightness of the four preload-adjustment screws (see Figure 12). In a more refined implementation, the trimpot could probably have been replaced with fixed resistors and the calibration performed using only the four preload-adjustment screws.

To estimate the downward pressure applied by the user, the sensed parameters were combined. FSR_{both} described the sum of the two FSR measurements:

$$FSR_{both}[n] = FSR_1[n] + FSR_2[n]. \qquad (1)$$

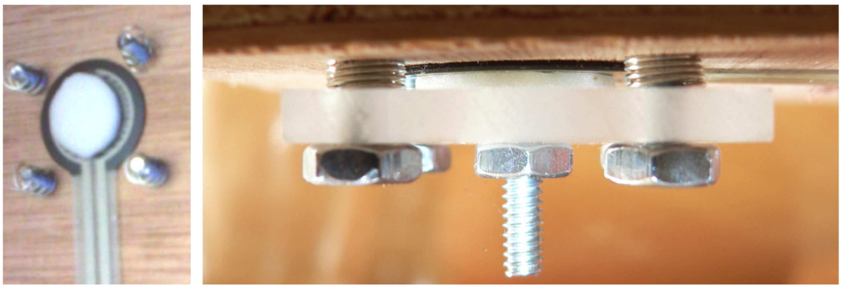

Figure 13. Sandwich elements shown with motorized fader removed (left: bottom view also with acrylic removed, right: side view with acrylic)

$$p_{FSR}[n] = \begin{cases} 0, & \text{if } c_{capsense}[n] < c_{thresh} \\ FSR_{both}[n_{touch}] - FSR_{both}[n], & \text{otherwise} \end{cases} \quad (2)$$

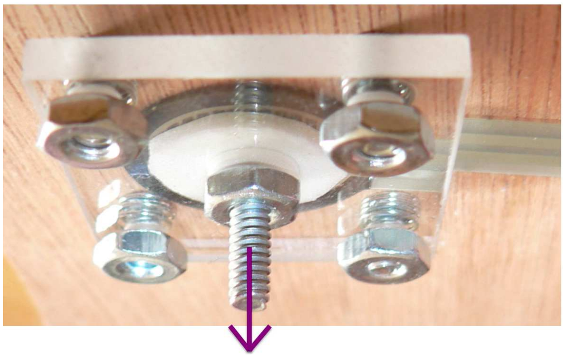

Figure 12. Sandwich elements with arrow showing the direction of the force that the fader end can exert on the sandwich

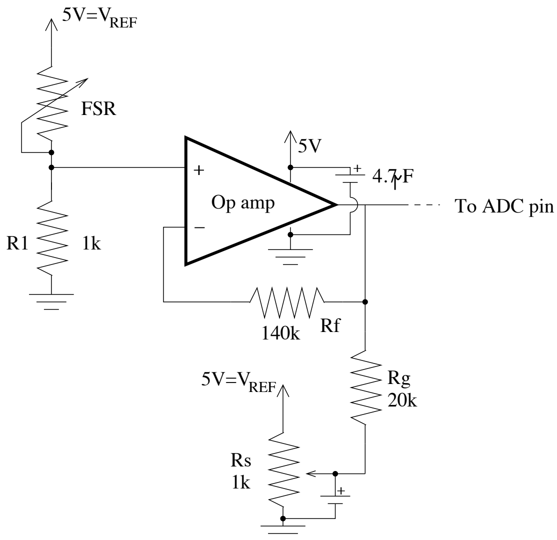

Figure 14. Operational amplifier circuit increases the dynamic range of FSR sensor signal while allowing the DC offset to be adjusted

With the amplification, the FSR signals were subject to noticeable hysteresis [26]. However, we could reduce the effect of the hysteresis using the capacitive sensing signal $c_{capsense}[n]$ to keep track of when the user was touching and releasing the fader knob (see (2) — c_{thresh} was a capacitive sensing threshold parameter to be calibrated depending on the electrical properties of the surrounding environment, and $n_{touch}+1$ indicated the most recent time index at which the signal $p_{FSR}[n]$ became non-zero).

Finally, by forming a linear combination of these two signals, we obtained a relatively inexpensive estimate of the downward force.

$$p_{total}[n] = \alpha \cdot p_{FSR}[n] + \beta \cdot c_{capsense}[n], \quad (3)$$

where $\alpha > 0$ and $\beta > 0$.

B. REFERENCES

[1] D. Birnbaum, “Musical vibrotactile feedback,” Master’s thesis, McGill University, Montreal, Canada, October 2007.

[2] 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*, Ann Arbor, MI, USA, July 1998.

[3] 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*, vol. 17, no. 1, pp. 19–29, Spring 1993.

[4] B. Verplank, “Haptic music exercises,” in *Proc. of the Int. Conf. on New Interfaces for Musical Expression*, Vancouver, BC, Canada, 2005, pp. 256–257.

[5] B. Gillespie and A. Okamura, “Haptic interaction for hands-on learning in system dynamics and controls,” accepted for publication in Con-

trol Systems Magazine, Last checked: March 30, 2012: http://www-personal.umich.edu/~brentg/Publications/Journal/2005e.htm.

[6] B. Verplank, M. Gurevich, and M. Mathews, "The Plank: designing a simple haptic controller," in *Proceedings of the 2002 conference on New Interfaces for Musical Expression*, Dublin, Ireland, May 24-26 2002.

[7] E. Berdahl, A. Kontogeorgakopoulos, and D. Overholt, "HSP v2: Haptic signal processing with extensions for physical modeling," in *Proceedings of the Haptic Audio Interaction Design Conference*, Copenhagen, Denmark, September 16-17 2010, pp. 61–62.

[8] E. Berdahl, G. Niemeyer, and J. Smith, "HSP: A simple and effective open-source platform for implementing haptic musical instruments," in *Proceedings of the Ninth International Conference on New Interfaces for Musical Expression*, Pittsburgh, PA, June 4-6 2009, pp. 262–263.

[9] E. Berdahl, H.-C. Steiner, and C. Oldham, "Practical hardware and algorithms for creating haptic musical instruments," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME-2008)*, Genova, Italy, June 5-7 2008, pp. 61–66.

[10] J. Castet, D. Couroussé, and J.-L. Florens, "A real-time simulator for virtual reality conceived around haptic hard constrains," in *Proc. International Conference on Enactive Interfaces*, Grenoble, France, November 19-22 2007.

[11] J. Rodriguez, A. Shahrokni, M. Schrittenloher, and M. Fjeld, "One-dimensional force feedback slider: Digital platform," in *IEEE VR 2007 Workshop on Mixed Reality User Interfaces: Specification, Authoring, Adaptation*, 2007, pp. 47–51.

[12] R. Gabriel, J. Sandsjo, A. Shahrokni, and M. Fjeld, "Bounceslider: Actuated sliders for music performance and composition," in *Proceedings of the Second International Conference on Tangible and Embedded Interaction*, Bonn, Germany, Feb 18-20 2008, pp. 127–130.

[13] T. Andersen, R. Huber, A. Kretz, and M. Fjeld, "Feel the beat: Direct manipulation of sound during playback," in *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, January 5-7 2006.

[14] B. Verplank and F. Georg, "Can haptics make new music? - fader and plank demos," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Oslo, Norway, May 30-June 1 2011.

[15] C. Cadoz, L. Lisowski, and J.-L. Florens, "A modular feedback keyboard design," *Computer Music Journal*, vol. 14, no. 2, pp. 47–51, 1990.

[16] C. Cadoz, A. Luciani, J.-L. Florens, and N. Castagné, "Artistic creation and computer interactive multisensory simulation force feedback gesture transducers," in *Proceedings of the 2003 Conference on New Interfaces for Musical Expression*, Montreal, Canada, 2003, pp. 235–246.

[17] 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.

[18] Y. Orlarey, D. Fober, and S. Letz, *FAUST: An Efficient Functional Approach to DSP Programming*. France: Edition Delatour, 2009.

[19] K. Kuchenbecker, J. Park, and G. Niemeyer, "Characterizing the human wrist for improved haptic interaction," in *Proceedings of the International Mechanical Engineering Congress and Exposition*, Washington, DC, USA, Nov. 2003.

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

[21] C. Carlson, E. Marschner, and H. McCurry, "The sound flinger: A haptic spatializer," in *Proc. New Interfaces for Musical Expression*, Oslo, Norway, May 30-June 1 2011, pp. 138–139.

[22] S. Gupta and J. Sadler, "PROJECT SQUEEZE: A low-cost 1-DOF exoskeletal haptic pincher," Stanford University, CS277 Term Project Report, March 11 2012.

[23] E. Berdahl, G. Niemeyer, and J. O. Smith, "Using haptic devices to interface directly with digital waveguide-based musical instruments," in *Proceedings of the Ninth International Conference on New Interfaces for Musical Expression*, Pittsburgh, PA, June 4-6 2009, pp. 183–186.

[24] E. Berdahl, "FireFader: A single degree-of-freedom force-feedback device for multimodal interaction with physical models," in *162nd Meeting of the Acoustical Society of America*, San Diego, CA, October 31-November 4 2011.

[25] A. Freed, "Application of new fiber and malleable materials for agile development of augmented instruments and controllers," in *Proceedings of the 2008 Conference on New Interfaces for Musical Expression*, Genova, Italy, 2008.

[26] A. Hollinger and M. Wanderley, "Evaluation of commercial force-sensing resistors," in *International Conference on New Interfaces for Musical Expression*, Paris, France, 2006.