Exploring the Virtual Reed Parameter Space Using Haptic Feedback

(Invited Paper)

Tamara Smyth*, Thomas N. Smyth†, and Arthur E. Kirkpatrick‡
School of Computing Science
Simon Fraser University
British Columbia, Canada
Email: *tamaras,†tsmyth,‡ted@cs.sfu.ca

***Abstract*— A high quality computer synthesis of an acoustic sound source does not necessarily yield a playable virtual musical instrument. A computer simulation of an acoustic musical instrument creates a disconnect between sound production and user input, and correspondingly, between hearing and feeling, in contrast to their interconnection in an acoustic instrument. This disconnect denies the user important haptic clues well known to help instrument control, impeding the user's ability to find, and remain inside, regions of playability. This research explores the addition of haptic feedback to a virtual reed model. In particular, we render the instrument's parameter space as a dynamic force field in order to support fine motor movements and, in turn, provide the user with cues regarding the instrument's oscillatory state and possible regions of playability. We then observe the effects that this additional feedback has on the user's ability to play the virtual instrument.**

I. Introduction

When playing an acoustic musical instrument, a musician interacts directly with the mechanism that produces that sound, modifying the physical parameters that are integral to the instrument's sound production. In so doing, the musician effectively becomes part of the vibrating system, their playing gestures significantly impacting the produced sound, while in turn, being influenced by the vibrations of the instrument.

In acoustic wind instruments, the mechanical oscillation of a reed relies on feedback from the bore pressure variations, a function of the flow through the reed (by which it is limited) and reflections from the bell. Conditions for reed oscillation—and thus playability—rely on the proper coupling of embouchure and blowing pressure, ensuring a strong resonance matching between the reed and the bore (this is particularly true of lip reed instruments).

When playing a reed instrument, the musician must therefore adapt their input, which for this research is considered to be reed tension (embouchure) and blowing pressure, to the instrument's oscillatory state. The squeaks and squeals one hears from a beginner clarinet player, or the inability to control and sustain a note on the trumpet, is partly a result of not yet having mastered a responsiveness to the instrument's feedback. With time and practice, a player is usually able to improve their control, using both audition and feel, and find regions of playability.

In physics-based computer simulations of musical instruments, the produced sound also responds to changes in control parameters, and as with their acoustic counterparts, a user must determine regions of playability by exploring the model's parameter space (though in many cases, the model will include a function limiting parameter values to ranges that ensure stability and oscillation). Depending on the user's ability to control these parameters, all the sounds and oscillatory states achievable on the acoustic model, stable or otherwise, should be reproducible in the simulation.

The virtual reed presented in [1] may be configured to operate as several different reed types (discussed in Section II), and like the acoustic models on which it is based, will oscillate periodically, chaotically, or not at all, in response to certain combinations of input parameters: reed tension (responsible for reed resonance), blowing pressure and bore frequency. Though it may certainly be desirable to produce inharmonic tones through aperiodic oscillation (as is done using "extended techniques"), any state that is either unstable or not oscillating at all, is described as being outside the playability region, since, in the former case, the reed is not responding predictably and reliably to the user's input, and in the latter case, the user is expelling energy but is getting nothing in return.

Depending on the type of reed, its physical parameters and its configuration, the conditions for oscillation and playability will vary considerably. It would be difficult (though certainly not impossible) to simply limit the model's parameter space to ensure playability, given the dependency on an overwhelming number of possible parameter combinations for each reed type. Also, though it may be desirable to inform the user they are approaching an unplayable region, it may not necessarily be desirable to restrict entry.

In order to judge the model's ability to produce a sound likeness to reed instruments, and also to improve playability of the simulation, transforming it from 'computational model' to 'virtual musical instrument', we present a method for exploring the virtual reed's parameter space using haptic feedback, provided by a Phantom device. We begin by describing the virtual reed model (see Section II), providing context for the discussion on playability and parameter mapping (see Section III), and then present (in Section IV) a method for using haptic feedback to direct the user into general regions of playability, and once there, guiding them while they explore.

II. The Virtual Reed Model

In reed instruments, as well as many vocal systems, air pressure from a source such as the lungs controls the oscillation of a valve by changing the pressure across a reed or membrane. This primary resonator, known as a pressure controlled-valve, is classified according to the effect of an additional pressure applied to the upstream or downstream side of the valve [2], [3]. If an increase in blowing pressure causes the valve to close further, and a bore pressure increase causes the valve to open further, the reed is said to be *blown closed*, the classification of most woodwind instruments. If a blowing pressure increase causes the valve to open further, and an increase in bore pressure causes the valve to close, the reed is *blown open*, the typical configuration of brass (lip reed) instruments, and the human voice. A *swinging door* or "transverse" reed, typically found in the avian syrinx, is one where a pressure increase from either side of the valve will cause it to open further [4].

The generalized reed model was first introduced in [1], providing a configurable model of a pressure controlled valve, allowing the user to design their own virtual reed, simply by setting model parameters. The parameters are continuously variable, and may be configured to produce blown closed, blown open, and the symmetric "swinging door" models (see Figure 1), as well as to set the valve geometry (see Figure 2).

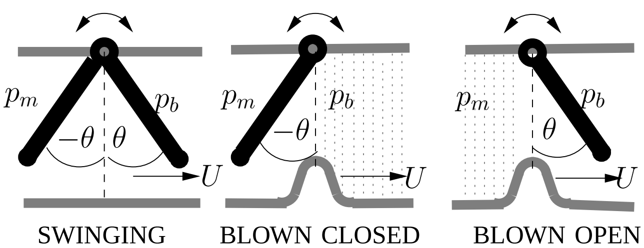

Fig. 1. *The valve types, showing four evolving model parameters: input mouth pressure p_m, bore pressure p_b, valve displacement $theta$, and volume flow U. For each valve type, the motion of the valve is constrained differently: blown open and close, or not at all (swinging).*

Figure 1 illustrates one mode of oscillation for each of three possible generalized valve configurations. The displacement of the valve is given by its angle θ from the vertical axis. The valve type is determined in part by the initial position of the valve θ_0 (its equilibrium position in the absence of flow), and in part by the use of a *stop*—a numerical limit placed at the center vertical axis that prevents the valve from swinging beyond the point $\theta = 0$ (see Figure 1 b and c).

If no stop is placed, as shown in Figure 1 a), the valve is free to swing across this center boundary and the model provides a symmetric "swinging" model, that is, an additional pressure from either side of the valve will cause it to open further. If a stop is placed in the channel, the configuration is further determined by the initial equilibrium position of the valve θ_0: an initial position to the left of the stop, at $\theta_0 < 0$, will cause the reed to *blow closed*, while an initial position to the right of the stop, $\theta_0 > 0$, will cause the reed to *blow open*. To specify the valve classification therefore, the user need only specify the equilibrium position θ_0 and whether the valve should be limited by $\theta = 0$ (for *blown open* and *closed* cases). A clarinet is implemented with $\theta < 0$ plus a stop.

The geometry of the valve may be further specified, as shown in Figure 2, by the effective length of the reed that sees the mouth pressure λ_m, the reed length that sees the bore pressure λ_b, and the reed length that sees the flow, given by μ. These variables have significant—and audible—effect on the overall driving force acting on the reed, given by F in (1), and can be seen as offering finer control of embouchure.

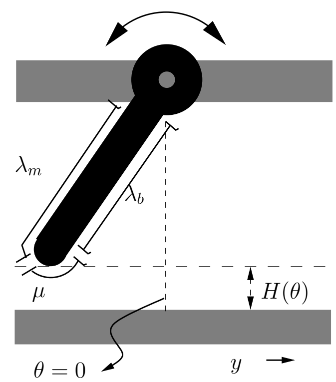

Fig. 2. *The geometric parameters of the virtual reed: λ_m is the effective length of the reed that sees the mouth pressure, λ_b is the reed length that sees the bore pressure, and the reed length that sees the flow, given by μ.*

Once the valve is set into motion, the value for θ is determined by the second order differential equation

$$m\frac{d^2\theta}{dt^2} + m2\gamma\frac{d\theta}{dt} + k(\theta - \theta_0) = F, \tag{1}$$

where m is the effective mass of the reed, γ is the damping coefficient, k is the stiffness of the reed, and F is the overall driving force acting on the reed, a function of the mouth and bore pressure, and flow in contact with the reed. The frequency of vibration for this mode is given by $\omega_v = \sqrt{k/m}$.

The differential equation governing air flow through the valve, fully derived in [5], [6], is given by

$$\frac{dU}{dt} = (p_m - p_b)\frac{A(t_0)}{\mu\rho} - \frac{U(t_0)^2}{2\mu A(t_0) + U(t_0)T}. \tag{2}$$

where p_m is mouth pressure, p_b is the bore pressure, modeled using waveguide synthesis methods [7], and $A(t)$ is the cross sectional area of the valve channel, and μ is the length of reed that sees the flow.

There are, therefore, three variables which evolve over time in response to an applied pressure

$$\begin{aligned} p_b &\triangleq \text{pressure in the bore at the mouthpiece} \\ \theta &\triangleq \text{displacement of the reed} \\ U &\triangleq \text{flow through the valve channel.} \end{aligned} \tag{3}$$

The model, implemented in Pd [8] for this research, permits access to the state of each of these variables, at any given time during performance.

III. Playability and the Parameter Space

Playability is loosely defined for the bowed string by Serafin *et al.*, as "the *volume* of the multidimensional parameter space in which *good tone* is produced" [9]. This definition refers to the produced sound of the computer simulation, and does not include a qualification of the interface to which it is connected, nor the ease, or intuition, with which parameters are controlled during a real-time performance. We adopt a similar definition of playability here, since our aim is not to evaluate the Phantom device as a controller for a wind instrument, but rather to explore, using haptic feedback, the parameters allowing for reed oscillation, while keeping within the bounds of a stable model.

The behaviour of the valve is governed by its dynamics, and the way in which upstream and downstream pressures exert force on the valve [1]. The generalized model allows for independent control of the valve dynamics, described by the effective tension of the reed and its corresponding resonant frequency f_r[1], as well as the upstream and downstream forces, functions of the blowing pressure and bore geometry, respectively. Therefore, the user has three continuously variable parameters with which to modify the produced sound:

$$\begin{aligned} p_m &\triangleq \text{blowing pressure}, \\ f_r &\triangleq \text{reed frequency}, \\ f_b &\triangleq \text{bore frequency}. \end{aligned} \tag{4}$$

Depending on the model's valve type and geometry, combinations of these parameters will yield varying results—as one would also expect of their acoustic counterparts. For example, the lip reed for brass instruments is larger, more massive, and less rigid than a clarinet reed. Because of its greater mass, the resonance of the lip reed plays a more essential role in sound production by having greater influence over the sounding pitch. In contrast, the clarinet reed, which is often simulated as being effectively masssless, influences the sounding pitch only in as much as it excites resonances of the bore. Because of the importance of the lip reed resonance (which for a two dimensional reed is inharmonic), it is more difficult to achieve oscillation, and thus playability, when it is coupled to the harmonic resonances of the clarinet's cylindrical bore.

There are, therefore, certain combinations of parameter values for which the reed simply will not oscillate. Likewise, certain parameters will cause the reed to behave unexpectedly, possibly producing a chaotic oscillation which, if not handled properly, may render the model unstable. We therefore describe a playability region as one in which combinations of reed frequency, f_r (either input directly or established through a tension input parameter), blowing pressure, p_m, and bore frequency, f_b, produce a stable oscillation, with significant amplitude. Of course, a playability region should tolerate deviations from a regular periodic oscillation, perhaps even one approaching chaos, but a stable oscillation for a given user input should be assured. To establish playability, we analyze the state of the model by looking at the reed displacement, θ (though any of the model's evolving parameters, p_b, θ, or U, could have just as easily been used).

[1]There is an option to have either the physical parameter of tension, or the mapped parameter of frequency, as input. Given the nonlinear relationship between the two, the latter is often more desirable.

A. Establishing Significant Amplitude

We begin by using an amplitude envelope detector (or *envelope follower*), to determine whether or not the reed is oscillating at significant amplitude. The amplitude envelope $y(n)$, is given by the difference equation

$$y(n) = (1 - \nu)|\theta(n)| + \nu y(n-1), \tag{5}$$

where ν determines how quickly changes in $\theta(n)$ are tracked. If ν is close to one, changes are tracked slowly; if ν is close to zero, θ has an immediate influence on y. In order to capture attacks in the signal, the value for ν is usually smaller for an increasing signal and larger for one that is decreasing. If the user is providing input, and thus energy, into the system and the reed amplitude goes below a certain threshold, the energy is wasted and that area is considered *unplayable*.

B. Establishing Stability and Chaos Tolerance

In a periodic oscillation, the duration from peak to peak, and the corresponding zero crossings, repeats regularly. In contrast, aperiodic, or chaotic, oscillations are less regular, and are characterized by more frequent zero crossings. We therefore determine whether the reed is approaching a chaotic state by tracking zero crossings, and setting up a counter that is incremented each time the signal goes from a negative to a positive value [10]. If the number of zero crossings suddenly becomes very large, it is likely because the reed oscillation is entering a region of chaotic behaviour.

Since it will be desirable to keep some of this behaviour, a tolerance threshold is established to the maximum level, beyond which the model becomes unstable. The region below this threshold of tolerance, and above the threshold of significant amplitude, is the region deemed to be *playable*.

IV. Central Differencing for Haptic Control

It is becoming increasingly popular to incorporate haptic feedback in music controllers [11], [12], [13]. The Phantom has been used to provide haptic feedback for several virtual instruments [14], particularly for physics-based plucked string models [15], [16].

A. Haptic Rendering

Here we provide a framework for a haptic rendering of a subset of the virtual reed's parameter space, that is, the input control parameters given in (4). In this discussion, we consider the virtual reed model to be defined by the function

$$f_M(I, S) \tag{6}$$

Fig. 3. A user controlling the virtual reed model using a Phantom and a MIDI keyboard.

where $I = \{p_m, f_r, f_b\}$ is the set of input parameters to the model, and $S = \{\theta, U, p_b\}$, defined by (3), is the model's current state.

The function σ linearly scales the Phantom's position (x, y, z) to match the scales of the model input parameters I. Next, we define a *playability* function f_p that computes the playability of the model, using the criteria described in Section III, given I. We then combine f_p with σ to produce a function

$$f_d(x, y, z, S) = f_p(f_M(\sigma(x, y, z), S)) \quad (7)$$

which computes the playability of the model M for a given Phantom position (x, y, z) and model state S. The force displayed by the Phantom is then determined by the gradient of f_d with respect to x, y, and z. This force tends to push the Phantom tip towards regions of greater playability. If the user insists on continuing along an axis opposing the force displayed by the Phantom, at a certain point the gradient force will flip and they will be guided into the next region of playability.

Since the gradient cannot be computed analytically, we use central differencing:

$$\frac{\partial f_d}{\partial x} \approx \frac{f_d(x + \Delta, y, z, S) - f_d(x - \Delta, y, z, S)}{2\Delta}, \quad (8)$$

and similarly for y and z, where Δ is some small perturbation constant.

B. Implementation

In our implementation we chose to map the Phantom's y and z position coordinates to the model's mouth pressure p_m, and reed frequency f_r parameters, respectively. The x coordinate of the Phantom was not mapped to bore frequency f_b as this proved too difficult to control. A midi device is used to control bore frequency instead.

The control algorithm was implemented, along with the model, in Pd [8]. A screenshot of the final Pd patch is provided in Figure 4 to serve as a reference for the following discussion.

We developed the `phantom~` object to communicate with the Phantom device. It has three inlets corresponding to the x, y, and z components of the force to be displayed by the Phantom. Similarly, it has three outlets which output the x, y, and z components of the Phantom tip's position. The output position coordinates are clamped and normalized to the range $[0, 1]$. Only x and y inlets and outlets are used in this implementation.

The focal point of the patch is the set of five `greed~` objects which implement the reed synthesis model described above. The leftmost such object, labeled 'master', produces the sound to be sent to the `dac~`, Pd's sound output object. The other four `greed~` objects, labeled 'slave', implement the gradient computation (8). One pair computes the central difference for mouth pressure, while the other computes if for reed frequency. The complete gradient computation consists of seven steps:

1) The Phantom position coordinates, y and z, are received by the `r~` objects.
2) The coordinates are perturbed by $\pm\Delta = 0.01$
3) The coordinates are converted to reed model input parameters as per our mapping σ.
4) The parameters are received by the reed model which performs its computation (f_M).
5) The output of the reed model is analyzed by the `playability~` object which computes a value for f_p.
6) Subtraction and division complete the central differencing calculation.
7) The result is sent as a force to the `phantom~` object and displayed.

Note that the same coordinate-to-parameter transformation occurs for the master object, but no perturbation is added.

V. Conclusion

We have presented a method which assists in controlling a physical model by reconnecting the user with the oscillating state of the instrument. The exploration of the model's parameter space is simplified by the multi-dimensional, continuous input from the Phantom, and the instrument's feedback to the user is greatly improved by the addition of force, guiding them while they explore the regions of playability.

Our approach is also highly modular. The set of parameters controlled, the measure of playability, and the synthesis model can all be exchanged and manipulated in the time it takes to modify a Pd patch.

In future work we hope to perform experiments with trained musicians to refine our haptic rendering, with the goal of incorporating our results in a custom haptic controller, more suited to a reed model.

Acknowledgment

We thank Steven Bergner for his advice on volume rendering and NSERC and CFI for funding.

References

[1] T. Smyth, J. Abel, and J. O. Smith, "A generalized parametric reed model for virtual musical instruments," in *Proceedings of ICMC 2005*, Barcelona, Spain, September 2005.

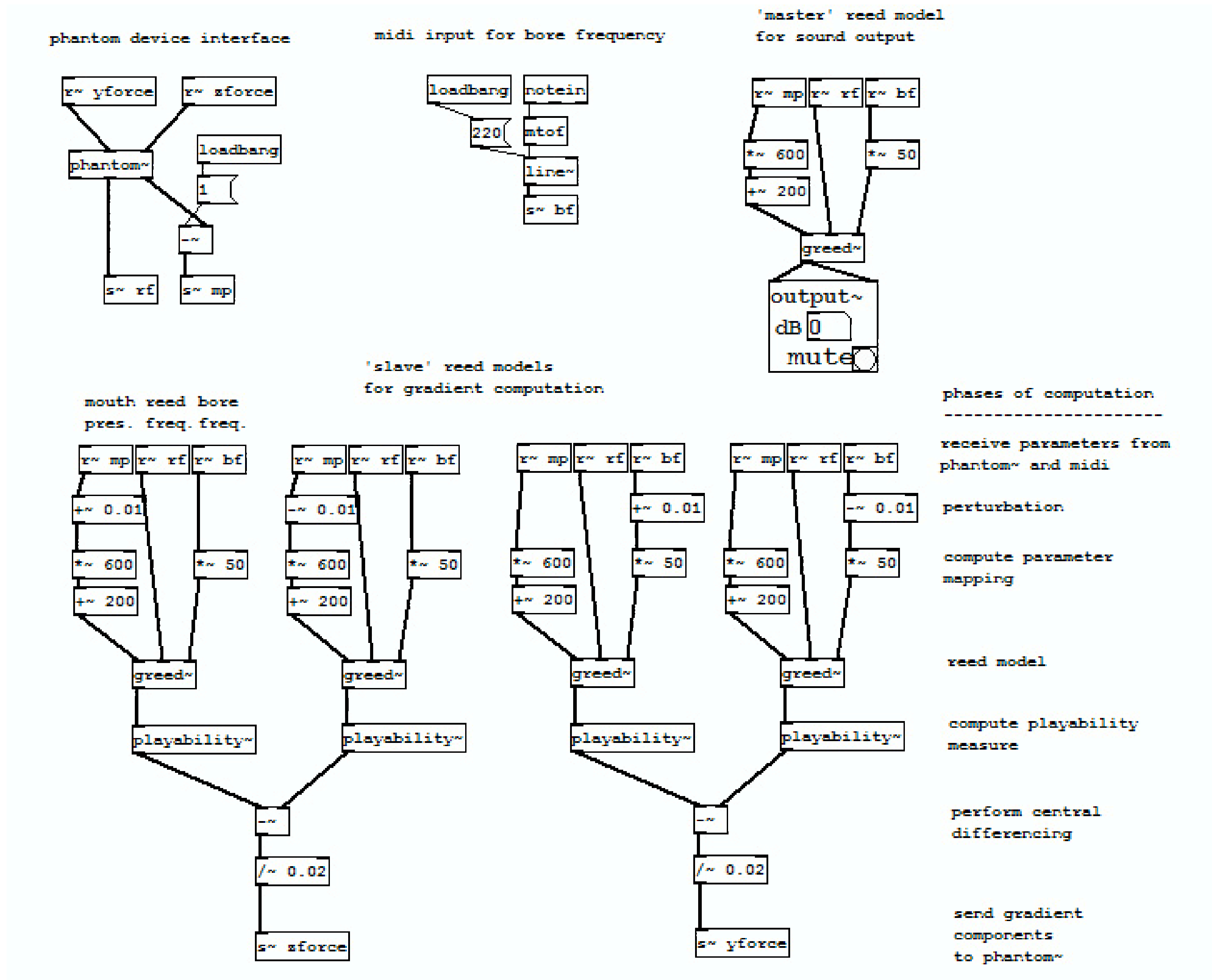

Fig. 4. A `Pd` patch implementing the haptic rendering technique. The `phantom~` object communicates with the Phantom while the other four compute the gradient of the playability function via central differencing. The stages of processing are described in the comments on the right.

[2] N. H. Fletcher, "Autonomous vibration of simple pressure-controlled valves in gas flows," *Jounal of the Acoustical Society of America*, vol. 93, no. 4, pp. 2172–2180, April 1993.

[3] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*. Springer-Verlag, 1995.

[4] T. Smyth, "Applications of bioacoustics to musical instrument technology," Ph.D. dissertation, Stanford University, April 2004.

[5] T. Smyth, J. Abel, and J. O. Smith, "Discrete-time simulation of airflow cut-off in pressure-controlled valves," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'03)*, New Paltz, New York, October 2003.

[6] ——, "The feathered clarinet reed," in *Proceedings of the International Conference on Digital Audio Effects (DAFx'04)*, Naples, Italy, October 2004.

[7] J. O. Smith, *Digital Waveguide Modeling of Musical Instruments*. ccrma.stanford.edu/˜jos/waveguide/, 2003.

[8] "Pd," http://www.pure-data.org.

[9] S. Serafin, J. O. Smith, and J. Woodhouse, "An investigation of the impact of torsion waves and friction characteristics on the playability of virtual bowed strings," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'03)*, New Paltz, New York, October 2003.

[10] J. S. Abel, "Private correspondence."

[11] C. Nichols, "The vbow: Development of a virtual violin bow haptic human-computer interface," in *Proceedings of NIME 2002*. Dublin, Ireland: Conference on New Instruments for Musical Expression, May 2002.

[12] B. Verplank, M. Gurevich, and M. Mathews, "The plank: Designing a simple haptic controller," in *Proceedings of NIME 2002*. Dublin, Ireland: Conference on New Instruments for Musical Expression, May 2002.

[13] G. Essl and S. O'Modhrain, "Scrubber: An interface for friction-induced sounds," in *Proceedings of NIME 2005*. Vancouver, Canada: Conference on New Instruments for Musical Expression, May 2005.

[14] W. P. Moss and B. Cunitz, "Haptic theremin: Developing a haptic musical controller using the sensable phantom omni," in *Proceedings of ICMC 2005*. Barcelona, Spain: International Computer Music Conference, September 2005, pp. 275–277.

[15] B. Verplank, 2000, demo at CCRMA, Stanford University.

[16] J. Feasel, "A virtual stringed instrument with haptic feedback," course project for COMP 259, University of North Carolina, http://www.cs.unc.edu/˜feasel/classes/259/final/results.html.