Exploring Haptics in Digital Waveguide Instruments

Aditya Tirumala Bukkapatnam - MUMT 618 Final Project

1 Introduction

Digital Waveguide systems essentially allow the modelling of traveling waves in a musical instrument through the use of delay lines. Popular examples of this approach include the plucked string models, bowed string models, open and closed acoustic pipes, which essentially model the velocity of a traveling acoustic wave [1].

While these modelled instruments often use external controllers to vary their parameters in real-time performance, some attempts have been made to integrate special controllers that also produce a force-feedback to the performer, allowing one to feel vibrations of the instrument's sound, as though interacting with a real traditional instrument [2, 3]. This study attempts to explore some of this work, in an attempt to gain insight into some of the intricacies of this process.

Berdahl et al. [4] provides a basic outline to a haptic instrument (Figure 1):

> *A Haptic musical instrument consists of actuators that exert forces on the musician, sensors that detect the gestures [being performed], … 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.*

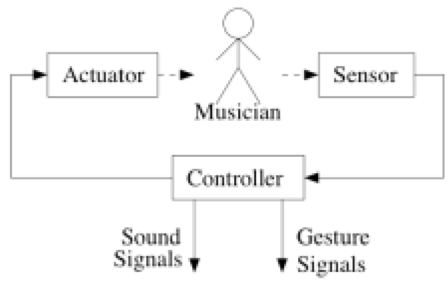

Figure 1: Haptic Musical Instrument Interaction as outlined by Berdahl et al.[4]

With that, one can begin to take a closer look at the integration of the components involved in Haptic Instruments.

2 Factors concerning Haptic Instruments

Haptic feedback in instruments is most often realized with the help of actuators, which can be a combination of Vibrotactile and Force-feedback actuators. While these components may be used as building blocks to construct a more sophisticated system, there already exist haptic systems that combine these actuators into one holistic haptic device. Such devices are often found in applications that involve tele-manipulation interactions, as discussed by Berdahl as *"Multi-DOF Haptic Devices"* [4].

However, when concerned with a system that provides force-feedback, certain factors arise which need to be accounted for.

2.1 Open and Closed Loop Systems

Vibrotactile feedback is produced by vibrating mechanisms and are commonly open ended systems, where a control signal produces a corresponding vibration output. On the other hand, force-feedback actuators function in closed-loops (except in the case of linear actuators). This means that the response of a device is also constantly being sensed as a means of correlating a control voltage to a measured output displacement. For e.g. a motorized fader often senses its own position, which functions as one of the control inputs to the servo that affects the fader's position [4].

Much of the work relevant to this study is related to the implementation of haptic feedback that involves the use of closed-loop feedback systems, with Digital Waveguide Systems.

2.2 Sampling Rate of the Control Loop

Another relevant factor here is that of system latency. Often haptic systems function in a control loop, where the information from its sensors is frequently used to calculate a result to the actuator and then provide a resultant feedback. This haptic control loop is run at its own sampling rate, which is known to vary between 1 – 3 kHz depending on extraneous factors like the type of controller and the application being used [4]. Luciani [5] (as cited in [4]) notes that setting this *"haptic sampling rate"* lower than the audio sampling rate, has perceptually noticeable results to a user, where generating feedback parameters at a higher sampling rate appears to produce results of a higher accuracy to the vibrations felt in a real instrument. However, it is understood to be a concern of efficiency, where a trade-off between a low latency (or *"system delay"*) and higher audio rate needs to be met because increasing the haptic sampling rate induces a higher delay between the controller and feedback [4]. Additionally, the deduction of haptic parameters at audio rate is also reported as being too computationally expensive to implement, without using dedicated DSP hardware [2, 4]. However, some attempts that have been made to implement this condition, which are discussed later in this study.

3 An Implementation of Haptic Feedback in Digital Waveguides

Various attempts to implement haptic interaction with physical models have been explored previously, which have been cited in references [4, 6, 7]. Amidst the various approaches to physically modelled musical instruments, hybrid digital waveguides emerged as a trend to working around the computational limitations posed by a single method. This effectively allows the instrument to be modeled through a combination of physical modelling approaches [8]. However, when interfaced with haptics, new challenges manifest due to the limitations posed by the control loop rate [6].

Here, we explore two of the proposed approaches, in an attempt to uncover what the possible challenges that each poses.

3.1 Direct Coupling

The most primitive of approaches, involves a unilateral interaction with the digital waveguide. Here, the actuators of the haptic device are driven as some non-linear function of the input sensors' values, as described in Figure 2.

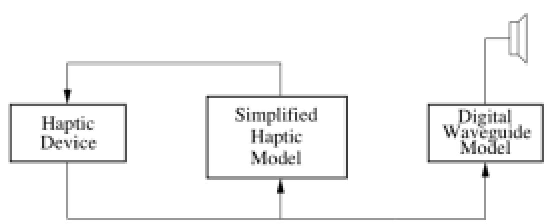

Figure 2: Simplified Haptic Model outlined by Berdahl [2]

While one might be able to feel a force or resistance and still operate the digital waveguide model, it doesn't pertain to feeling the forces of the waveguide instrument. Additionally, the haptic interaction doesn't provide any new means of influencing the velocities in the waveguide. Berdahl [2] observes this as a gross simplification of the haptic model, which intrinsically limits the possible scope of interactions.

At this stage, it is important to re-establish that the goal of implementing haptics is to be able to interact with the instrument, via the velocities being simulated within the waveguide. This would allow one to feel vibrations as though one were interacting with a real instrument. With that in mind the most basic implementation for interacting with the waveguide would be to directly couple the haptic device with the waveguide's forces or velocities. However, a haptic device comes with its own impedance characteristics that might render the parameters generated by the waveguide or haptic device inadequate to influence each other. Additionally, there exist the risks of an instability arising due to delay-free loops. To address this situation, Berdahl introduces a *"control junction element"* [3].

3.2 The Control Junction Element

In an attempt to revive the direct integration of haptic feedback with digital waveguides, Berdahl [2] proposes the use of *"control junction element"* that attempts to directly integrate the haptic interface with a digital waveguide model of a string.

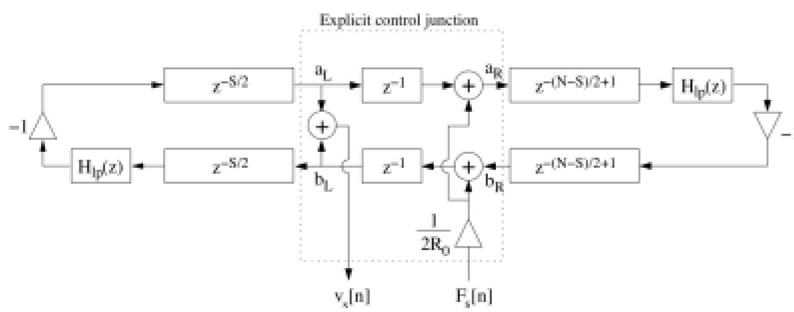

Figure 3: Berdahl's Control Junction Element [2].

Figure 3 implements a digital waveguide model, where an interaction causes a wave to travel in two directions, from the point of interaction. The control junction interfaces the haptic controller as a sensor and actuator connected one sample apart in the digital waveguide, at the point of interaction [2]. From this, the velocity $v_s[n]$ of the wave at the control point can be deduced as sum of the instantaneous velocities at that point:

$$v_s[n] = a_L + b_L$$

Additionally, If the waveguide has a wave impedance of R_0, the applied input force $F_s[n]$ is scaled by a factor of $1/2R_0$.

However, it is noted that an actuator is often limited by a maximum stiffness that it can generate. If the waveguide implements a very high stiffness value, the system with 1 sample separation can tend to instability as in the direct coupling method. To address this, the maximum stiffness needs to be limited, and a damping condition introduced to avoid instability. This is implemented by a spring-damper system illustrated in Figure 4, where the spring k binds the maximum stiffness, and a damping factor R, controls the system preventing an unstable condition.

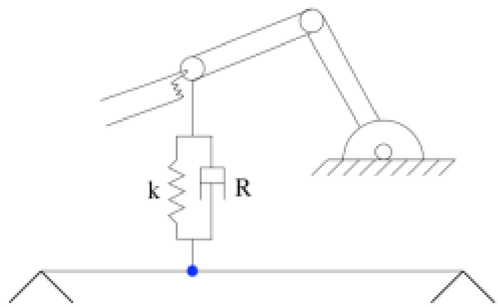

Figure 4: Spring Damper system to stabilize the haptic system [2]

3.2.1 Adapting the Mechanism

Haptic devices often sense displacement, but modelled instruments produce velocity outputs. Berdahl [2] expresses the use of displacement derivative to adapt the sensed displacement to velocity, and integrate the velocity of the waveguide into a displacement. These functions would be performed by the Velocity and Displacement Estimators in Figure 5.

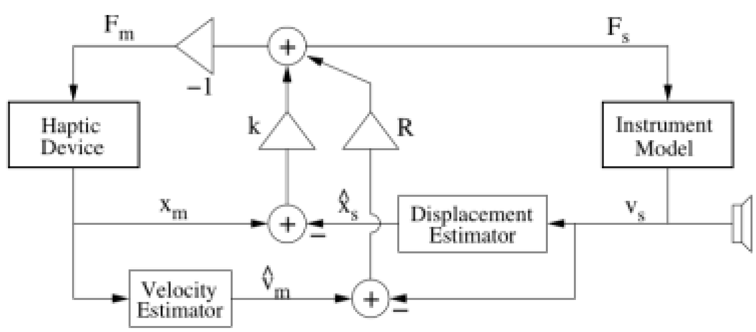

Figure 5: Adapting variables in the interfacing process [2]

From Figure 5, the resultant force can be expressed as:

$$F_s[n] = k(x_m[n] - \hat{x}_s[n]) + R(\hat{v}_m - v_s[n]) = -F_m[n]$$

Varying k and R allows haptic systems of different impedances to be adapted to the waveguide, while reproducing the feedback parameters generated within the waveguide. However, it has been noted that there is the possibility of an unstable condition arising, for high frequencies and large loop gains induced by the waveguide. To implement a plucked string, the damper is omitted, leaving a “spring that disengages at… large force levels”.

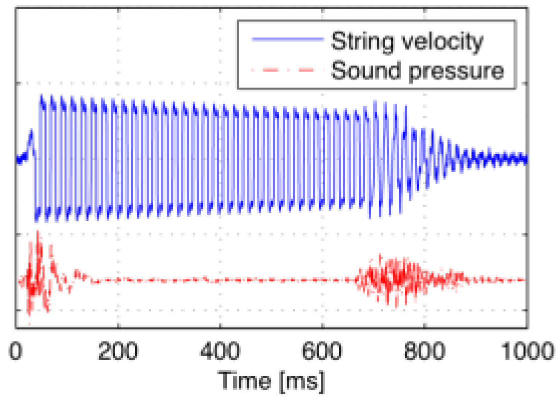

Figure 6: Haptic interaction with a Plucked String[2]

Figure 6 illustrates the interaction with the digital waveguide, implemented the explicit control junction discussed above. The string is plucked and damped using haptic feedback, where the grip of the haptic device controls the damping factor. A *“buzz”* is produced when damping occurs, due to the large string displacement at lower frequencies (~50Hz).

Berdahl [2] also extends this control junction into an *“implicit control junction”,* where the unit sample of delay can be omitted. This is done by evaluating the input string force as a function of a known string impedance;

$$F_s[n] = \frac{2R_0}{2R_0 + R + \frac{k}{f_s}}\left(k(x_m[n] - p\hat{x}_s[n-1]) + R\hat{v}_m[n] - \left(R + \frac{k}{f_s}\right)(a_L[n] + b_R[n])\right)$$

In this case, the system is run at the audio rate, and allows for an interaction with the plucked string model. However, in the observation of bowed strings, other factors come into play.

3.3 Haptic Interaction with Bowed String models

Although other implementations of haptic interactions with bowed string models have been attempted [3], a challenge in digital waveguide model approaches is the affair of frictional forces of the bow interacting with the string. The frictional interaction at the bow-string junction is a function of velocities of traveling wave components at the bow junction, which involves a more computationally inefficient ordeal of deriving velocity from position.

One workaround is to implement a bow-table, which effectively implements the stick-slip interaction in the form a look-up table that relates string velocity and bow pressure in terms of a reflection coefficient μ. One such approach is explored by Sinclair et al. [7] using the Hayward-Armstrong (H-A) friction model, which is implemented as the "bow-table" (see Figure 7) in the C++ library STK (Synthesis Tool Kit), [3, 7].

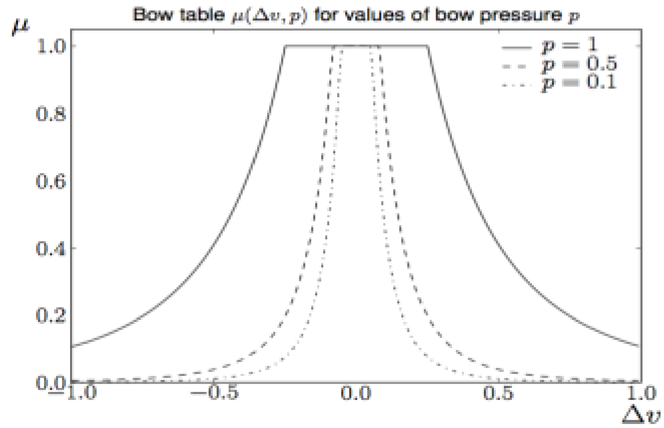

Figure 7: The "bow-table" implemented in STK [7]

The principle idea here is illustrated by Figure 8, where a point on the moving entity x, is related to a point of adhesion w, by the strain relation $z = x - w$, in the direction of motion. When the displacement force being applied causes the strain value to exceed a predefined maximum value Z_{max} called the *"breakaway distance"*, the adhesion point w relocations to constantly retain the value of z such that it doesn't exceed Z_{max}. This illustrates the objects interacting with a *"pre-sliding"* factor and a sticking behavior, which is analogous to the behavior of a mass-spring system [9]. The H-A model builds on the Dahl friction model and essentially attempts to model the bow friction, as shown in Figure 9.

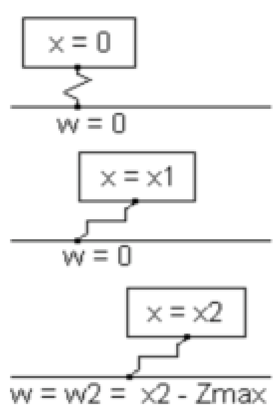

Figure 8: Dahl's friction model

In our case, the friction is interpret as a restoring force between the sensor position indicated by the haptic device x_k and an *"anchor point"* w_k, which is said to be vary in the direction of bow movement with respect to a term α such that:

$$w_k = w_{k-1} + \alpha(z_k)\,|x_k - x_{k-1}|\,z_k$$

where,

$$z_k = x_k - w_k$$

$$\alpha(z) = \begin{cases} \dfrac{1}{Z_{max}}, & |z| > Z_{max} \\ 0, & otherwise \end{cases}$$

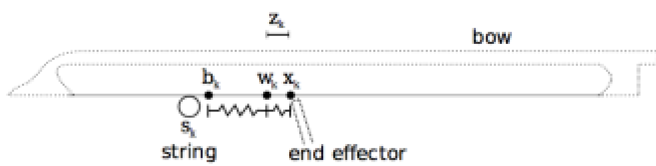

Figure 9: Using the H-A model to interpret bow friction [7]

In this effort, the value μ is used to interpret conditions of absorption and transmission of energy, or in other words, of regions stick and slip. Therefore, the value $(1 - \mu)$ used with velocity calculates the frictional force experienced, where a value $\mu = 1$, should ideally result in the complete arrest of the bow movement, indicating an infinite frictional opposition. Additionally, in order to circumvent noise challenges from the use of velocity from the sensor input, a virtual bowing point b is implemented to allow friction to be modelled as a spring between b and the effector position x. The friction that needs to be modelled is then deduced from the friction force F_f, given by [7]:

$$F_f = (x_k - b_k)\,p\,\mu_x,$$

where, μ_x depends on change in string velocities $v_x - v_s$ and downward bow pressure p, implemented in STK as:

$$\mu(\Delta v, p) = \min([|\Delta v\,(5 - 4p)| + 0.75]^{-4}, 1)$$

and the value of v_x is replaced with a virtual velocity that is estimated using a velocity estimator that derives the velocity from the H-A anchor point w_k, effectively giving the value of μ_x from μ_w, which evaluates the final force as:

$$F_f = (x_k - b_k)\, p\, \mu_w$$

$$\text{where, } \mu_w = \mu(v_w - v_s, p)$$

With this, the friction force is synthesized for feedback, by deriving from the velocities occurring within the waveguide, and a bow table that controls the stick-slip frictional motion [7, 9].

3.3.1 System Sampling Rate Considerations

A DSP assisted approach is taken towards solving the challenge pertaining to sample rates, where a higher sampling rate of 16kHz is made possible for both the audio and haptic systems. The earlier discussion on limitation of system latency is moderately overcome by the use of DSP assisted system.

However, this system is also executed on a dual sample rate system, where the haptic sample rate is set at 1kHz and run synchronously with the audio system running at 48kHz.

4 Conclusions

This study attempts to explore haptic interaction with Digital Waveguide based Virtual Instruments, where the haptic feedback parameters can be derived from the forces acting within the waveguide itself. A brief introduction to haptic systems reveals some of the considerations that need to be made in terms of sampling rates and control loops. The haptic sampling rate, often found between 1 – 3kHz, is discovered to have a significant impact on the perceptual quality of the feedback; however, involves a trade-off between quality of feedback versus system latency. Some solutions achieve higher sampling rates close to 16kHz, using the help of DSP systems, and report that even higher rates might be achievable through implementation of DSP optimized routines.

In order to deduce parameters for feedback from within the waveguide synthesis, a control junction is proposed where the velocities of the traveling wave within the waveguide are adapted to the haptic device, through matching the impedance of the waveguide with the impedance of the haptic device. This is done through a spring-damper system. While the explicit control junction is capable of extracting the haptic parameters directly from the digital waveguide, there are noted potential instability conditions that can arise in the system. Nevertheless, an interesting interaction of this interface is presented, where the user is able to damp the vibrating plucked string model, through haptic interaction.

The evaluation of parameters for a bowed string are slightly more complex due to the bow-string interactions. This is illustrated with citation of Serafin's [6] exploration with the vBow, wherein haptic feedback for the frictional and normal bow forces were attempted. Here, the audio and haptic systems operated at different sampling rates, and communicated over MIDI. An extension of this approach is presented by Sinclair [7], where the system is realized using DSP assistance, thereby achieve a higher overall sampling rate of 16kHz. This method uses a Hayward – Armstrong friction model, to model the frictional forces as a function of the velocity of traveling waves within the bowed string model, and a bow-pressure variable.

Berdahl's *"implicit control junction"* attempts to model the bowed string implementation by using the same bow-table lookup approach, to calculate a non-linear response of a damper coefficient in the spring-damper control junction. From these examples, some methods of interfacing a haptic device to a digital waveguide instrument have been investigated. While the velocity component appears to be the parameter of interest in the interfacing process, each model appears to bring its own set of challenges, into the implementation domain.

5 References

5.1 Primary Sources

1. Smith, J.O., *Physical Audio Signal Processing*. 2010.
2. Berdahl, E., G. Niemeyer, and J.O. Smith. *Using Haptic Devices to Interface Directly with Digital Waveguide-Based Musical Instruments*. in *NIME*. 2009. Citeseer.
3. Sinclair, S., *Velocity-driven audio-haptic interaction with real-time digital acoustic models*. 2012, McGill University.
4. Berdahl, E., H.-C. Steiner, and C. Oldham. *Practical Hardware and Algorithms for Creating Haptic Musical Instruments*. in *NIME*. 2008. Citeseer.
6. Serafin, S., et al. *Expressive controllers for bowed string physical models*. in *Proc. DAFX 2001, Limerick, Ireland*. 2001.
7. Sinclair, S., G. Scavone, and M.M. Wanderley, *Audio-haptic interaction with the digital waveguide bowed string*. 2009: Citeseer.

5.2 Secondary Sources

5. Luciani, A., et al. *Ergotic sounds: A new way to improve playability, believability and presence of digital musical instruments*. in *ENACTIVE/07-International Conference on Enactive Interfaces*. 2007. ACROE.
8. Bilbao, S., *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics*. 2006, Wiley.
9. Hayward, V. and B. Armstrong, *A new computational model of friction applied to haptic rendering*, in *Experimental Robotics VI*. 2000, Springer. p. 403-412.