Digital Waveguide Architectures for Virtual Musical Instruments

Julius O. Smith III

Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, Stanford, California 94305, `http://ccrma.stanford.edu/~jos/`

Summary. This chapter summarizes some efficient signal processing structures used for virtual musical instruments based on physical models. Instruments in the string and wind families are considered.

1 Introduction

Digital sound synthesis has become a standard staple of modern music studios, videogames, and personal computers. As processing power has increased over the years, sound synthesis implementations have evolved from dedicated chip sets, to single-chip solutions, and ultimately to software implementations within processors used primarily for other tasks (such as graphics or general purpose computing). With the cost of implementation dropping closer and closer to zero, there is increasing room for higher quality algorithms. A particularly fertile source of natural sound synthesis algorithms is the mathematical models of musical instruments developed within the science of musical acoustics [20, 25, 51]. To realize practical instrument voices from these models, it is helpful to develop robust and efficient signal processing algorithms which retain the audible physical behavior while minimizing computational cost [72].

In this article, a number of cost-effective synthesis models will be summarized for various musical instrument families, including strings, and winds. Emphasis is placed on techniques adapted from the field of digital signal processing [35, 42]. Notably absent is any discussion of percussion instruments, which are normally handled via sample-based methods [36, 26, 41], but some model-based methods have been proposed based on the digital waveguide mesh [21].

2 Vibrating Strings

In a stringed musical instrument, most of the sound energy is stored in the vibrating string at any given time. The main determinant of the sound of a stringed instrument is the interaction of the string and player. The body of the instrument functions as a passive resonator which is well modeled, in principle, by a linear, time-invariant filter [62, 69].

The musical acoustics literature on stringed musical instruments is quite rich. See, for example, [20, 38, 39, 40, 44, 45, 46, 47, 55, 81, 80]. Digital computational models of stringed instruments have been under active development since at least the 1960s [6, 8, 9, 17, 19, 18, 23, 24, 30, 29, 50, 52, 59, 58, 62, 63, 64, 66, 71, 75, 76, 78, 77, 79].

2.1 Wave Equation

The starting point for a stringed instrument model is typically a *wave equation* for transverse vibrations of the vibrating string [10, 6, 39, 80]. For example, a recently proposed [6] Partial Differential Equation (PDE) governing motion of a *piano string* is given by

$$f(t,x) = \epsilon\ddot{y} - Ky'' + EIy'''' + R_0\dot{y} + R_2\ddot{y}' \tag{1}$$

where

$$\begin{aligned}
y = y(t,x) &= \text{string displacement at position } x \text{ and time } t\\
\dot{y} = \frac{\partial}{\partial t}y(t,x), \qquad y' &= \frac{\partial}{\partial x}y(t,x), \qquad \text{(etc.)}\\
f(t,x) &= \text{driving force density (N/m)}\\
\epsilon &= \text{mass density (kg/m)}\\
K &= \text{tension force along the string axis (N)}\\
E &= \text{Young's modulus (N/m}^2\text{)}\\
I &= \text{radius of gyration of the string cross-section (m)}
\end{aligned}$$

The basic *lossless* wave equation $\epsilon\ddot{y} = Ky''$ is derived in most textbooks on acoustics, *e.g.*, [39].[1] The term $\epsilon\ddot{y}$ represents the mass per unit length times the transverse *acceleration*, and Ky'' equals the transverse *restoring force* due to the string tension K. The more elaborate wave equation for piano string includes frequency-dependent *losses* and *dispersion*. Frequency-dependent losses are critical for obtaining the correct decay time as a function of frequency, *i.e.*, for each partial overtone. Dispersion (frequency-dependent propagation speed) is required to obtain the correct *tuning* of the partial overtones.

The term EIy'''' in Eq. (1) is the transverse restoring force exerted by a *stiff string* in response to being bent. In an ideal string, with zero diameter, this force is zero. Stiffness is normally neglected in models for guitars and violins, but included in instruments with larger-diameter strings, such as the piano and cello. The test for whether stiffness is needed in the model for

[1] For an online derivation, see, *e.g.*, `http://ccrma.stanford.edu/˜jos/pasp/String_Wave_Equation.html`.

plucked or struck strings (any freely vibrating string) is whether the ear can hear the "stretching" of the partial overtones due to stiffness [27]; for bowed strings, the dispersion due to stiffness can effect the *bow-string dynamics* [37]. In the context of a digital waveguide string model (described in §2.3 below), the dispersion associated with stiff strings is modeled indirectly by designing an *allpass filter* for the string model. It is possible to correctly tune the first several tens of partials for any natural piano string with a total allpass order of 20 or less [50]. Additionally, minimization of the L^∞ norm [33] has been used to calibrate a series of allpass-filter sections [5, 56].

The final two terms of Eq. (1) provide *damping*, which is required in any string practical model. The damping associated with R_0 is frequency-independent, while the damping due to the R_2 term increases with frequency [6]. For digitally simulated piano strings of the highest quality, more than these two terms are needed in the PDE, to yield more finely tuned decay times versus frequency. Instead of introducing such terms into the wave equation based on physical considerations, these terms are normally determined implicitly by *digital filter design* techniques [43, 62]. For this application, the error minimized by the filter-design software should be formulated in terms of the audibility of the error in partial overtone decay rates and tuning [62, pp. 182–184]. For example, in [7], the damping in real piano strings was modeled using a length 17 linear-phase FIR filter for the lowest strings, and a length 9 linear-phase FIR filter for the remaining strings.

2.2 Finite Difference Models

The original approach to digitally modeling vibrating strings was by means of *Finite Difference Schemes (FDS)* [22, 52, 24, 10, 73]. Such models are also called Finite Difference Time Domain (FDTD) methods [28, 29]. In these models, partial derivatives are replaced by finite differences, *e.g.*,

$$\ddot{y}(t,x) \approx \frac{y(t+T,x) - 2y(t,x) + y(t-T,x)}{T^2} \tag{2}$$

$$y''(t,x) \approx \frac{y(t,x+X) - 2y(t,x) + y(t,x-X)}{X^2} \tag{3}$$

2.3 Digital Waveguide Models

More recently, the *Digital Waveguide (DW)* approach has been developed for modeling vibrating strings [64, 65, 71]. It can be viewed as a descendent of the *Kelly-Lochbaum model* for voice synthesis [12, 31, 34, 35, 67]. The DW approach is compared quantitatively with the FDS approach in [6]. For strings used in typical musical instruments, the digital waveguide method generally provides a more efficient simulation for a given sound quality level. A combination of digital waveguides and finite differences may be preferred, however, for nonlinear string simulation [32, 29, 45].

The digital waveguide formulation can be derived by simply *sampling* the *traveling-wave* solution to the ideal wave equation

$$Ky'' = \epsilon\ddot{y}.$$

It is easily checked that the lossless 1D wave equation is solved by any string shape y which travels to the left or right with speed $c = \sqrt{K/\epsilon}$ [16]. Denote *right-going* traveling waves in general by $y_r(t - x/c)$ and *left-going* traveling waves by $y_l(t+x/c)$, where y_r and y_l are assumed twice-differentiable. Then, as is well known, the general class of solutions to the lossless, one-dimensional, second-order wave equation can be expressed as

$$y(t,x) = y_r\left(t - \frac{x}{c}\right) + y_l\left(t + \frac{x}{c}\right). \tag{4}$$

Sampling these traveling-wave solutions yields

$$\begin{aligned} y(nT, mX) &= y_r(nT - mX/c) + y_l(nT + mX/c) \\ &= y_r[(n-m)T] + y_l[(n+m)T] \\ &= y^+(n-m) + y^-(n+m) \end{aligned} \tag{5}$$

where a "+" superscript denotes a "right-going" traveling-wave component, and "−" denotes propagation to the "left". This notation is similar to that used for acoustic-tube modeling of speech [35].

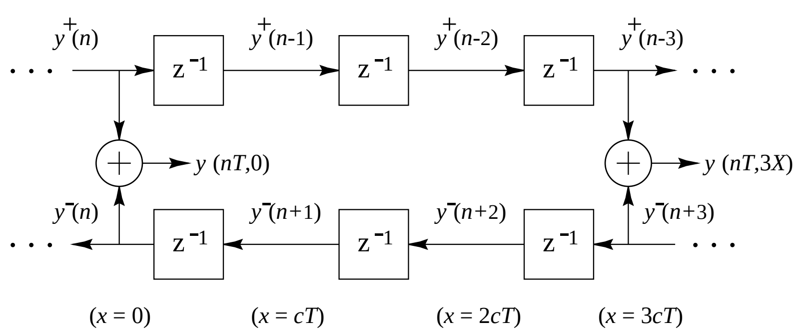

Fig. 1. Digital simulation of the ideal, lossless waveguide with observation points at $x = 0$ and $x = 3X = 3cT$. (The symbol "z^{-1}" denotes a one-sample delay.) [Reprinted with permission from [71]]

Figure 1 shows a signal flow diagram for the computational model of Eq. (5), which is often called a digital waveguide model (for the ideal string in this case) [65, 71]. Note that, by the sampling theorem [68, Appendix

G],[2] it is an exact model so long as the initial conditions and any ongoing additive excitations are bandlimited to less than half the temporal sampling rate $f_s = 1/T$.

Note also that the position along the string, $x_m = mX = mcT$ meters, is laid out from left to right in the diagram, giving a physical interpretation to the horizontal direction in the diagram, even though spatial samples have been eliminated in the translation of physical variables to traveling-wave components. In Fig. 1, "transverse displacement outputs" have been arbitrarily placed at $x = 0$ and $x = 3X$. The diagram is similar to that of well known ladder and lattice digital filter structures [35], except for the delays along the upper rail, the absence of scattering junctions, and the direct physical interpretation.

2.4 FDTD and DW Equivalence

In [67, 70], it is shown that the FDS and DW recursions for the ideal vibrating string are *equivalent*. That is, a one-to-one linear transformation exists which translates the state space of one to the other, and the time updates perform the same state-space transition in each case. As a result, the methods only differ in low-level computational details such as numerical sensitivity, cost efficiency, and the implementations of excitations and boundary conditions. In one dimension, the DW method is much more efficient in most applications. In higher dimensions, however, in which membranes and acoustic spaces are modeled using a grid of intersecting digital waveguides—the so-called *digital waveguide mesh*—the FDS approach is generally more efficient than the DW method. (See [3] for quantitative comparisons).

2.5 Bowed Strings

An example DW model for a bowed-string instrument is shown in Fig. 2 [63, 71]. The main control is bow velocity, but bow force and position also have an effect on the tone produced. The digital waveguide simulates traveling velocity-wave components. The left- and right-going traveling-wave components on the left of the bow are denoted $v^+_{s,l}(n)$ and $v^-_{s,l}(n)$, respectively, where n denotes time in samples. To the right of the bow, the components are $v^+_{s,r}(n)$ and $v^-_{s,r}(n)$. The (abstract) "incoming string velocity" is defined as

$$v^+_s(n) = v^+_{s,l}(n) + v^+_{s,r}(n) \tag{6}$$

and the "incoming differential velocity" is defined as

$$v^+_\Delta(n) = v_b(n) - v^+_s(n), \tag{7}$$

where $v_b(n)$ denotes the bow velocity at sample-time n. The incoming differential velocity v^+_Δ can be interpreted physically as the physical differential

[2] http://ccrma.stanford.edu/~jos/mdft/Sampling_Theorem.html

velocity (bow minus string) that would occur if the bow-string friction were zero (ideal, frictionless "slipping" of the bow along the string). A table-lookup (or other nonlinear function implementation) gives the *reflection coefficient* of the bow-string contact point, as seen by traveling waves on the string. This coefficient is then applied to v_Δ^+ and added to the left- and right-going traveling-wave paths. The bow table is derived from the bow-string friction-curve characteristic, such as the one shown in Fig. 3. The details of this derivation may be found in [71].[3]

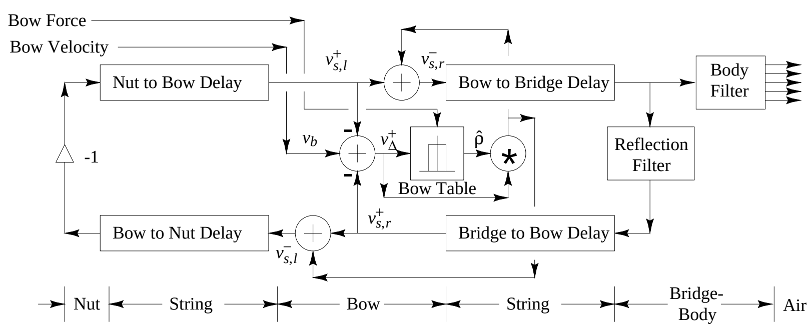

Fig. 2. Digital waveguide bowed-string model. [From [71]]

The delay lines are drawn in "physical canonical form" for ease of physical interpretation. We see that the string is modeled using two ideal (lossless) digital waveguides, one to the left and one to the right of the bowing point. (A 1D *digital waveguide* is defined as a pair of delay lines flowing in opposite directions—a *bidirectional delay line*.) In practice, only two delay lines are generally implemented, one on each side of the bowing point.

Note that delay lines require $\mathcal{O}(1)$ operations per sample, *i.e.*, the number of operations per sample does not increase as the delay-line length is increased.[4] This is the heart of the reason digital waveguide models are more efficient than finite difference models. At present, there is no known $\mathcal{O}(1)$ FDS (or FDTD) model for vibrating strings.

The reflection filter in Fig. 2 implements *all losses in one period of oscillation* due to the yielding bridge, absorption by the bow and finger, string losses, etc. Since the string model is linear and time invariant, *i.e.*, Eq. (1) is linear with constant coefficients, superposition applies, and loss/dispersion

[3] Available online at
`http://ccrma.stanford.edu/~jos/pasp/Bow_String_Scattering_Junction.html`.

[4] The notation $\mathcal{O}(K)$ denotes "computational complexity of order K". This means that the computational complexity is bounded by cN^K for some constant c, as $N \to \infty$, where N is the size of the problem (delay-line length in this case).

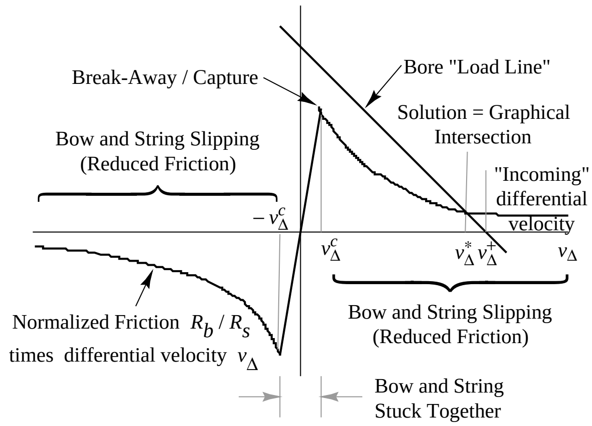

Fig. 3. Overlay of normalized bow-string friction curve $R_b(v_\Delta)/R_s$ with the string "load line" $v_\Delta^+ - v_\Delta$. The "capture" and "break-away" differential velocity is denoted v_Δ^c. Note that increasing the bow force increases v_Δ^c as well as enlarging the maximum force applied (at the peaks of the curve). [From [71]]

filtering within the string may be *commuted* to concentrated points. In principle, such filters should appear on either side of the bow, and prior to each output signal extracted. However, because the difference is perceptually moot, normally only one loss/dispersion filter is employed per string loop. For multiple coupled strings, all loss/dispersion filtering may be implemented within the *bridge* at which they share a common termination [66, 71].[5]

The bow-string junction is typically implemented as a *memoryless* lookup table (or segmented polynomial). Preferably, however, a *thermodynamic model* should be employed for bow friction, since the bow rosin is known to have a time-varying viscosity due to temperature variations within a period of sound [82]. In [61], thermal models of dynamic friction in bowed strings are discussed, and such models have been incorporated into more recent synthesis models [57, 60, 1].

A real-time software implementation of a bowed-string model similar to that shown in Fig. 2 is available in the Synthesis Tool Kit (STK) distribution [14, 11], as `Bowed.cpp`. This prototype can serve as a starting framework for more elaborate models.

2.6 Electric Guitars

While most musical vibrating strings are well approximated as linear, time-invariant systems, there are special cases in which *nonlinear* behavior is desired. One example is the distorted electric guitar.

[5] `http://ccrma.stanford.edu/~jos/pasp/Two_Ideal_Strings_Coupled.html`

A *soft clipper* is similar to a hard clipper (saturation on overflow), but with the "corners" smoothed. A common choice of soft-clipper is the *cubic nonlinearity*, *e.g.* [74],

$$f(x) = \begin{cases} -\frac{2}{3}, & x \leq -1 \\ x - \frac{x^3}{3}, & -1 \leq x \leq 1 \\ \frac{2}{3}, & x \geq 1 \end{cases} \tag{8}$$

This particular soft-clipping characteristic is diagrammed in Fig. 4. An analysis of its spectral characteristics, along with some discussion regarding how to avoid the aliasing it can cause, is given in [71].[6] An input gain may be used to set the desired degree of distortion.

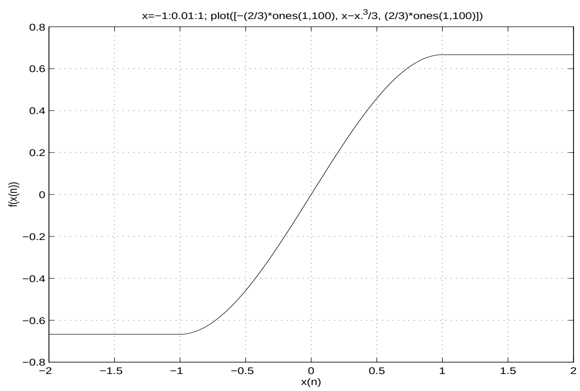

Fig. 4. Soft-clipper defined by Eq. (8). [From [71]]

A cubic nonlinearity, as well as *any* odd distortion law,[7] generates only odd-numbered harmonics (like in a square wave). For best results, and in particular for *tube distortion* simulation [2, 53], it can be argued that some amount of even-numbered harmonics should also be present. Breaking the odd symmetry in any way will add even-numbered harmonics to the output as well. One simple way to accomplish this is to add an *offset* to the input signal, obtaining

$$y(n) = f[x(n) + c], \tag{9}$$

where c is some small constant. (Signals $x(n)$ in practice are typically constrained to be zero mean by one means or another.)

Another method for breaking the odd symmetry is to add some square-law nonlinearity to obtain

[6] `http://ccrma.stanford.edu/~jos/pasp/Nonlinear_Elements.html`

[7] A function $f(x)$ is said to be *odd* if $f(-x) = -f(x)$.

$$f(x) = \alpha x^3 + \beta x^2 + \gamma x + \delta \tag{10}$$

where β controls the amount of square-law distortion. This is then a more general third-order polynomial. A square-law is the gentlest nonlinear distortion, as can be seen by considering the Taylor series expansion of a general nonlinearity transfer characteristic $f(x)$. The constant δ can be chosen to zero the mean, on average; if the input signal $x(n)$ is zero-mean with variance is 1, then $\delta = -\beta$ compensates the nonzero mean introduced by the squaring term. The term γ can be modified to adjust the "effect mix".

2.7 Amplifier Feedback

A nonlinear feedback effect used with distorted electric guitars is *amplifier feedback*. In this case, the amplified guitar signal couples back into the strings with some gain and delay, as depicted schematically in Fig. 5 [74]. The feedback delay can be adjusted to cause different partial overtones to be amplified relative to others.

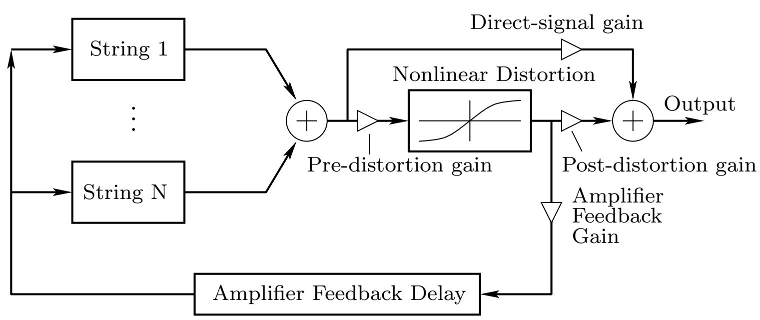

Fig. 5. Simulation of a basic distorted electric guitar with amplifier feedback. [From [71]]

2.8 Commuted Synthesis

Figure 6 depicts a diagram of *commuted synthesis* for an acoustic guitar [66, 71, 78]. The string and body resonator have been *commuted*—an operation valid for all linear, time-invariant systems. Thus, instead of plucking the string and filtering the string output with a digital filter of extremely high order (to capture the many resonances in the range of human hearing), the "pluck response" of the guitar body (a filtered impulse response) can be fed to the string instead, as shown in Fig. 7. In a typical implementation, the guitar-body impulse response (or some filtering of it), is stored in table, just as in sampling synthesis, and a low-order filter is applied to the table playback in order to impart details of the plucking excitation. This simplification exchanges an expensive body filter for an inexpensive “pluck filter”. In addition to body resonances, the excitation table may include characteristics of the listening space as well. Commuted synthesis of the piano has been developed to a high degree of quality by Bensa [5].

Fig. 6. Schematic diagram of commuted synthesis of plucked/struck stringed instruments. [From [71]]

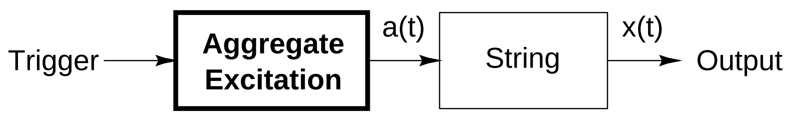

Fig. 7. Use of an aggregate excitation given by the convolution of original excitation with the resonator impulse response. [From [71]]

3 Wind Instruments

A basic DW model for a single-reed woodwind instrument, such as a clarinet, is shown in Fig. 8 [63, 54, 71].

When the bore is cylindrical (plane waves) or conical (spherical waves), it can be modeled quite simply using a bidirectional delay line [54]. Because the main control variable for the instrument is air pressure in the mouth at the reed, it is convenient to choose pressure wave variables. Thus, the delay-lines carry left-going and right-going *pressure* samples p_b^+ and p_b^- (respectively) which represent the traveling pressure-wave components within the bore.

To first order, the bell passes high frequencies and reflects low frequencies, where “high” and “low” frequencies are divided by the wavelength which equals the bell’s diameter. Thus, the bell can be regarded as a simple “cross-over” network, as is used to split signal energy between a woofer and tweeter in a loudspeaker cabinet. For a clarinet bore, the nominal “cross-over frequency” is around 1500 Hz [4].

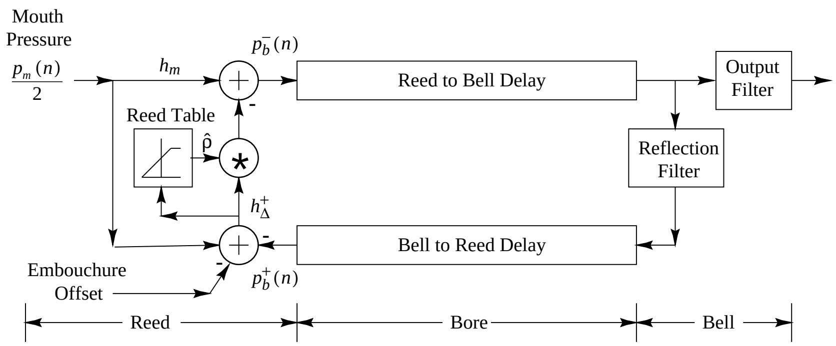

Fig. 8. Waveguide model of a single-reed, cylindrical-bore woodwind, such as a clarinet. [From [71]]

The reflection filter at the right of the figure implements the bell or tone-hole losses as well as the round-trip attenuation losses from traveling back and forth in the bore. The bell output filter is highpass, and *power complementary* with respect to the bell reflection filter. Power complementarity follows from the assumption that the bell itself does not vibrate or otherwise absorb sound. The bell is also *amplitude complementary* [71].

The reed is modeled as a signal- and embouchure-dependent *nonlinear reflection coefficient* terminating the bore. Such a model is possible because the reed mass is neglected. The player's embouchure controls damping of the reed, reed aperture width, and other parameters, and these can be implemented as parameters on the contents of the lookup table or nonlinear function.

Equation (11) below shows a simple function that can be sampled and loaded into a reed table. The controlling mouth pressure is denoted p_m. The *reflection-coefficient* of the reed is denoted $\rho(h_\Delta^+)$, where $h_\Delta^+ \triangleq p_b^-/2 - p_b^+$ ("incoming half-pressure-drop"). A simple choice of *embouchure control* is a simple additive offset in the reed-table address. Since the main feature of the reed table is the pressure-drop where the reed begins to open, such a simple offset can implement the effect of biting harder or softer on the reed, or changing the reed stiffness.

In the field of computer music, it is customary to use simple piecewise linear functions for functions other than signals at the audio sampling rate, *e.g.*, for amplitude envelopes, FM-index functions, and so on [49, 48]. Along these lines, good initial results were obtained [63] using the simplified *qualitatively* chosen table

$$\hat{\rho}(h_\Delta^+) = \begin{cases} 1 - m(h_\Delta^c - h_\Delta^+), & -1 \le h_\Delta^+ < h_\Delta^c \\ 1, & h_\Delta^c \le h_\Delta^+ \le 1 \end{cases} \tag{11}$$

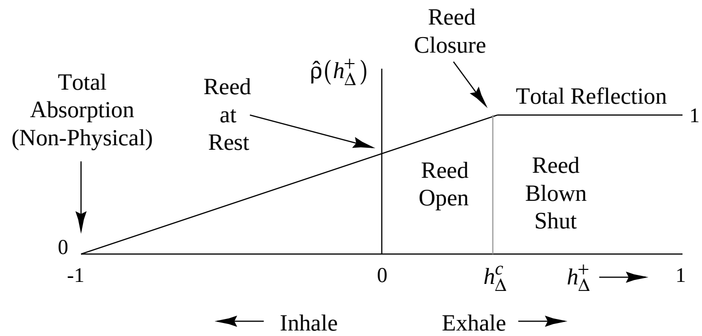

Fig. 9. Simple, qualitatively chosen reed table for the digital waveguide clarinet. [From [71]]

depicted in Fig. 9 for $m = 1/(h_\Delta^c + 1)$. The corner point h_Δ^c is the smallest pressure difference giving reed closure.[8] Embouchure and reed stiffness correspond to the choice of offset h_Δ^c and slope m. Brighter tones are obtained by increasing the curvature of the function as the reed begins to open; for example, one can use $\hat{\rho}^k(h_\Delta^+)$ for increasing $k \geq 1$.

Another variation is to replace the table-lookup contents by a piecewise polynomial approximation. While less general, good results have been obtained in practice [13, 14, 15].

An intermediate approach between table lookups and polynomial approximations is to use interpolated table lookups. Typically, linear interpolation is used, but higher order polynomial interpolation can also be considered [71].[9]

STK software [11] implementing a model as in Fig. 8 can be found in the file `Clarinet.cpp`.

4 Conclusion

In this section, a number of signal processing architectures were summarized that have been found suitable for computational modeling of acoustic musical instruments. These algorithms generally provide a high degree of sound quality and expressive response at a small fraction of the computational cost associated with more general-purpose computational modeling techniques.

[8]For operation in fixed-point DSP chips, the independent variable $h_\Delta^+ \triangleq p_m/2 - p_b^+$ is generally confined to the interval $[-1, 1)$. Having the table go all the way to zero at the maximum negative pressure $h_\Delta^+ = -1$ is not physically reasonable (0.8 would be more reasonable), but has the practical benefit that when the lookup-table input signal is about to clip, the reflection coefficient goes to zero, thereby opening the feedback loop.

[9]`http://ccrma.stanford.edu/~jos/pasp/Delay_Line_Interpolation_I.html`

References

1. F. Avanzini, S. Serafin, and D. Rocchesso, "Modeling interactions between rubbed dry surfaces using an elasto-plastic friction model," in *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx-02), Hamburg, Germany*, pp. 111–116, September 2002, `http: //www.dafx.de/`.
2. E. Barbour, "The cool sound of tubes," *IEEE Spectrum*, pp. 24–35, August 1998.
3. M. J. Beeson and D. T. Murphy, "RoomWeaver: A digital waveguide mesh based room acoustics research tool," in *Proceedings of the Conference on Digital Audio Effects (DAFx-04), Naples, Italy*, Oct. 2004.
4. A. H. Benade, *Fundamentals of Musical Acoustics*, New York: Dover, 1990.
5. J. Bensa, *Analysis and Synthesis of Piano Sounds using Physical and Signal Models*, PhD thesis, Université de la Méditérranée, Marseille, France, 2003, `http: //www.lma.cnrs-mrs.fr/~bensa/`.
6. J. Bensa, S. Bilbao, R. Kronland-Martinet, and J. O. Smith, "The simulation of piano string vibration: from physical models to finite difference schemes and digital waveguides," *Journal of the Acoustical Society of America*, vol. 114(2), pp. 1095–1107, 2003.
7. J. Bensa, S. Bilbao, R. Kronland-Martinet, and J. Smith, "Computational modeling of stiff piano strings using digital waveguides and finite differences," *EURASIP Journal on Applied Signal Processing*, vol. 4, 2004, (special issue on string modeling), in review.
8. A. Chaigne, "On the use of finite differences for musical synthesis. application to plucked stringed instruments," *Journal d'Acoustique*, vol. 5, no. 2, pp. 181–211, 1992.
9. A. Chaigne and A. Askenfelt, "Numerical simulations of piano strings. I. a physical model for a struck string using finite difference methods," *Journal of the Acoustical Society of America*, vol. 95, no. 2, pp. 1112–1118, 1994.
10. A. Chaigne and A. Askenfelt, "Numerical simulations of piano strings, parts I and II," *Journal of the Acoustical Society of America*, vol. 95, pp. 1112–1118, 1631–1640, Feb.–March 1994.
11. P. Cook and G. Scavone, *Synthesis ToolKit in C++, Version 4.0*, `http: //-ccrma.stanford.edu/CCRMA/Software/STK/`, March 2002.
12. P. R. Cook, *Identification of Control Parameters in an Articulatory Vocal Tract Model, with Applications to the Synthesis of Singing*, PhD thesis, Elec. Engineering Dept., Stanford University (CCRMA), Dec. 1990, `http: //www.cs.princeton.edu/~prc/`.
13. P. R. Cook, "A meta-wind-instrument physical model, and a meta-controller for real time performance control," in *Proceedings of the 1992 International Computer Music Conference, San Jose*, pp. 273–276, Computer Music Association, 1992.
14. P. R. Cook, "Synthesis toolkit in C++, version 1.0," in *SIGGRAPH Proceedings*, Assoc. Comp. Mach., May 1996, see `http: //-www.cs.princeton.edu/~prc/NewWork.html` for a copy of this paper. The Synthesis Tool Kit (STK) software itself is distributed by CCRMA: `http: //ccrma.stanford.edu/CCRMA/Software/STK/`.
15. P. R. Cook, *Real Sound Synthesis for Interactive Applications*, A. K. Peters, L.T.D., 2002.

16. J. l. R. d'Alembert, "Investigation of the curve formed by a vibrating string, 1747," in *Acoustics: Historical and Philosophical Development* (R. B. Lindsay, ed.), pp. 119–123, Stroudsburg: Dowden, Hutchinson & Ross, 1973.
17. G. Derveaux, A. Chaigne, P. Joly, and E. Bécache, "Time-domain simulation of a guitar: Model and method," *Journal of the Acoustical Society of America*, vol. 114, pp. 3368–3383, Dec. 2003.
18. C. Erkut and M. Karjalainen, "Finite Difference Method vs. Digital Waveguide Method in String Instrument Modeling and Synthesis," in *International Symposium on Musical Acoustics*, (Mexico City, Mexico), December 2002.
19. C. Erkut, M. Karjalainen, P. Huang, and V. Välimäki, "Acoustical analysis and model-based sound synthesis of the kantele," *Journal of the Acoustical Society of America*, vol. 112, pp. 1681–1691, October 2002.
20. N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments, Second Edition*, New York: Springer Verlag, 1998.
21. F. Fontana and D. Rocchesso, "Physical modeling of membranes for percussion instruments," *Acta Acustica*, vol. 84, pp. 529–542, May/June 1998.
22. B. Fornberg, *A Practical Guide to Pseudo-Spectral Methods*, Cambridge University Press, 1998.
23. A. Härmä, M. Karjalainen, L. Savioja, V. Välimäki, U. K. Laine, and J. Huopaniemi, "Frequency-warped signal processing for audio applications," *Journal of the Audio Engineering Society*, vol. 48, pp. 1011–1031, November 2000.
24. L. Hiller and P. Ruiz, "Synthesizing musical sounds by solving the wave equation for vibrating objects. Part I.," *Journal of the Audio Engineering Society*, vol. 19, pp. 462–470, June 1971, Part II: vol. 19, no. 7, pp. 542–551, July/Aug. 1971.
25. A. Hirschberg, "Aero-acoustics of wind instruments," in *Mechanics of Musical Instruments* (A. Hirschberg, J. Kergomard, and G. Weinreich, eds.), pp. 291–361, Berlin: Springer-Verlag, 1995.
26. A. Horner, J. Beauchamp, and L. Haken, "Methods for multiple wavetable synthesis of musical instrument tones," *Journal of the Audio Engineering Society*, vol. 41, pp. 336–356, May 1993.
27. H. Järveläinen, V. Välimäki, and M. Karjalainen, "Audibility of inharmonicity in string instrument sounds, and implications to digital sound synthesis," in *Proceedings of the 1999 International Computer Music Conference, Beijing*, pp. 359–362, Oct. 22-27, 1999, `http://www.acoustics.hut.fi/~hjarvela/publications/`.
28. M. Karjalainen, "Mixed physical modeling: DWG + FDTD + WDF," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY*, (New York), pp. 225–228, IEEE Press, Oct. 2003.
29. M. Karjalainen and C. Erkut, "Digital waveguides vs. finite difference schemes: Equivalance and mixed modeling," *EURASIP Journal on Applied Signal Processing*, vol. 2004, pp. 978–989, June 15, 2004.
30. M. Karjalainen, J. Backman, and J. Pölkki, "Analysis, modeling, and realtime sound synthesis of the kantele, a traditional Finnish string instrument," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Minneapolis*, (New York), pp. 229–232, IEEE Press, 1993.

31. J. L. Kelly and C. C. Lochbaum, "Speech synthesis," *Proceedings of the Fourth International Congress on Acoustics, Copenhagen*, pp. 1–4, September 1962, Paper G42. Reprinted in [?, pp. 127–130].
32. A. Krishnaswamy and J. O. Smith, "Methods for simulating string collisions with rigid spatial objects," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY*, (New York), IEEE Press, Oct. 2003.
33. M. Lang, "Allpass filter design and applications," *IEEE Transactions on Signal Processing*, vol. 46, no. 9, pp. 2505–2514, 1998.
34. H.-L. Lu, *Toward a High-Quality Singing Synthesizer with Vocal Texture Control*, PhD thesis, Elec. Engineering Dept., Stanford University (CCRMA), July 2002, `http://ccrma.stanford.edu/~vickylu/thesis/`.
35. J. D. Markel and A. H. Gray, *Linear Prediction of Speech*, New York: Springer Verlag, 1976.
36. D. Massie, "Wavetable sampling synthesis," in *Applications of DSP to Audio & Acoustics* (M. Kahrs and K. Brandenburg, eds.), pp. 311–341, Boston/Dordrecht/London: Kluwer Academic Publishers, 1998.
37. M. E. McIntyre and J. Woodhouse, "On the fundamentals of bowed string dynamics," *Acustica*, vol. 43, pp. 93–108, Sept. 1979.
38. M. E. McIntyre, R. T. Schumacher, and J. Woodhouse, "Aperiodicity in bowed-string motion," *Acustica*, vol. 49, pp. 13–32, Sept. 1981.
39. P. M. Morse, *Vibration and Sound*, `http: //asa.aip.org/publications.html`: American Institute of Physics, for the Acoustical Society of America, 1948, 1st edition 1936, last author's edition 1948, ASA edition 1981.
40. P. M. Morse and K. U. Ingard, *Theoretical Acoustics*, New York: McGraw-Hill, 1968.
41. A. Ng and A. Horner, "Iterative combinatorial basis spectra in wavetable matching," *Journal of the Audio Engineering Society*, vol. 50, no. 12, pp. 1054–1063, 2002.
42. A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete Time Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1999.
43. T. W. Parks and C. S. Burrus, *Digital Filter Design*, New York: John Wiley and Sons, Inc., June 1987, contains FORTRAN software listings.
44. A. D. Pierce, *Acoustics*, American Institute of Physics, for the Acoustical Society of America, 1989, `http: //asa.aip.org/publications.html`.
45. R. Pitteroff and J. Woodhouse, "Mechanics of the contact area between a violin bow and a string, part ii: Simulating the bowed string," *Acta Acustica*, vol. 84, pp. 744–757, 1998.
46. R. Pitteroff and J. Woodhouse, "Mechanics of the contact area between a violin bow and a string, part iii: Parameter dependence," *Acta Acustica*, vol. 84, pp. 929–946, 1998.
47. C. V. Raman, "On the mechanical theory of vibrations of bowed strings, etc," *Indian Assoc. Cult. Sci. Bull.*, vol. 15, pp. 1–158, 1918.
48. C. Roads, ed., *The Music Machine*, Cambridge, MA: MIT Press, 1989.
49. C. Roads and J. Strawn, eds., *Foundations of Computer Music*, Cambridge, MA: MIT Press, 1985.
50. D. Rocchesso and F. Scalcon, "Accurate dispersion simulation for piano strings," in *Proc. Nordic Acoustical Meeting (NAM'96)*, (Helsinki, Finland), p. 9 pages, June 12-14 1996.

51. T. D. Rossing, F. R. Moore, and P. A. Wheeler, *The Science of Sound (3rd Edition)*, Reading MA: Addison-Wesley, 2003.
52. P. M. Ruiz, *A Technique for Simulating the Vibrations of Strings with a Digital Computer*, PhD thesis, Music Master Diss., Univ. Ill., Urbana, 1969.
53. B. Santo, "Volume cranked up in amp debate," *Electronic Engineering Times*, pp. 24–35, October 3, 1994, http: //www.trueaudio.com/at_eetjlm.htm.
54. G. P. Scavone, *An Acoustic Analysis of Single-Reed Woodwind Instruments with an Emphasis on Design and Performance Issues and Digital Waveguide Modeling Techniques*, PhD thesis, CCRMA, Music Dept., Stanford University, March 1997, http://ccrma.stanford.edu/~gary/.
55. J. C. Schelleng, "The bowed string and the player," *Journal of the Acoustical Society of America*, vol. 53, pp. 26–41, Jan. 1973.
56. S. Serafin and J. O. Smith, "Impact of string stiffness on digital waveguide models of bowed strings," *Catgut Acoustical Society Journal*, vol. 4, no. 4, pp. 49–52, 2001.
57. S. Serafin, *The sound of friction: real-time models, playability and musical applications*, PhD thesis, Music Department, Stanford University, 2004.
58. S. Serafin and D. Young, "Bowed string physical model validation through use of a bow controller and examination of bow strokes," in *Proceedings of the Stockholm Musical Acoustics Conference (SMAC-03), http: //www.speech.kth.se/smac03/*, 2003.
59. 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, New Paltz, NY*, (New York), IEEE Press, Oct. 1999.
60. S. Serafin, P. Huang, S. Ystad, C. Chafe, and J. O. Smith, "Analysis and synthesis of unusual friction-driven musical instruments," in *Proceedings of the 2002 International Computer Music Conference, Sweden*, 2002.
61. J. H. Smith and J. Woodhouse, "The tribology of rosin," *Journal of Mechanics and Physics of Solids*, vol. 48, pp. 1633–1681, Aug. 2000.
62. J. O. Smith, *Techniques for Digital Filter Design and System Identification with Application to the Violin*, PhD thesis, Elec. Engineering Dept., Stanford University (CCRMA), June 1983, CCRMA Technical Report STAN-M-14, http: //ccrma.stanford.edu/STANM/stanm14/.
63. J. O. Smith, "Efficient simulation of the reed-bore and bow-string mechanisms," in *Proceedings of the 1986 International Computer Music Conference, The Hague*, pp. 275–280, Computer Music Association, 1986, also available in [64].
64. J. O. Smith, "Music applications of digital waveguides," Tech. Rep. STAN-M-39, CCRMA, Music Department, Stanford University, 1987, CCRMA Technical Report STAN-M-39, http: //-ccrma.stanford.edu/STANM/stanm39/.
65. J. O. Smith, "Physical modeling using digital waveguides," *Computer Music Journal*, vol. 16, pp. 74–91, Winter 1992, special issue: Physical Modeling of Musical Instruments, Part I. http: //ccrma.stanford.edu/~jos/pmudw/.
66. J. O. Smith, "Efficient synthesis of stringed musical instruments," in *Proceedings of the 1993 International Computer Music Conference, Tokyo*, pp. 64–71, Computer Music Association, 1993, incorporated into [71].

67. J. O. Smith, "Principles of digital waveguide models of musical instruments," in *Applications of Digital Signal Processing to Audio and Acoustics* (M. Kahrs and K. Brandenburg, eds.), pp. 417–466, Boston/Dordrecht/London: Kluwer Academic Publishers, 1998.
68. J. O. Smith, *Mathematics of the Discrete Fourier Transform (DFT)*, `http://w3k.org/books/`: W3K Publishing, 2003, `http://ccrma.stanford.edu/~jos/mdft/`.
69. J. O. Smith, *Introduction to Digital Filters*, `http://ccrma.stanford.edu/~jos/filters/`, May 2004.
70. J. O. Smith, "On the equivalence of digital waveguide and finite difference time domain schemes," July 21, 2004, `http://arxiv.org/abs/physics/0407032/`.
71. J. O. Smith, *Physical Audio Signal Processing: Digital Waveguide Modeling of Musical Instruments and Audio Effects*, `http://ccrma.stanford.edu/~jos/pasp/`, August 2004.
72. J. O. Smith, "Virtual acoustic musical instruments: Review and update," *Journal of New Music Research*, vol. 33, no. 3, pp. 283–304, 2004.
73. J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Pacific Grove, CA: Wadsworth and Brooks, 1989.
74. C. R. Sullivan, "Extending the Karplus-Strong algorithm to synthesize electric guitar timbres with distortion and feedback," *Computer Music Journal*, vol. 14, pp. 26–37, Fall 1990.
75. T. Tolonen, V. Välimäki, and M. Karjalainen, "Modeling of tension modulation nonlinearity in plucked strings," *IEEE Transactions on Speech and Audio Processing*, vol. SAP-8, pp. 300–310, May 2000.
76. C. Traube and J. Smith, "Estimating the fingering and the plucking points on a guitar string from a recording," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY*, (New York), IEEE Press, Oct. 2001.
77. V. Välimäki, M. Laurson, and C. Erkut, "Commuted waveguide synthesis of the clavichord," *Computer Music Journal*, vol. 27, pp. 71–82, Spring 2003.
78. V. Välimäki, J. Huopaniemi, M. Karjalainen, and Z. Jánosy, "Physical modeling of plucked string instruments with application to real-time sound synthesis," *Journal of the Audio Engineering Society*, vol. 44, pp. 331–353, May 1996.
79. V. Välimäki, H. Penttinen, J. Knif, M. Laurson, and C. Erkut, "Sound synthesis of the harpsichord using a computationally efficient physical model," *EURASIP Journal on Applied Signal Processing*, vol. 7, pp. 934–948, 2004, special issue on Model-Based Sound Synthesis.
80. C. Vallette, "The mechanics of vibrating strings," in *Mechanics of Musical Instruments* (A. Hirschberg, J. Kergomard, and G. Weinreich, eds.), pp. 115–183, Berlin: Springer-Verlag, 1995.
81. G. Weinreich, "Coupled piano strings," *Journal of the Acoustical Society of America*, vol. 62, pp. 1474–1484, Dec 1977, see also [?] and *Scientific American*, vol. 240, p. 94, 1979.
82. J. Woodhouse, "Bowed string simulation using a thermal friction model," *Acustica – Acta Acustica*, vol. 89, pp. 355–368, 2003.

Index