Elementary Digital Waveguide Models for Vibrating Strings

Julius Smith and Nelson Lee

RealSimple Project*
Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
Stanford, California 94305

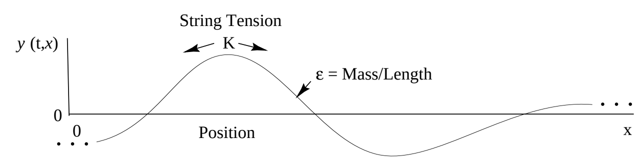
June 5, 2008

Outline

- Ideal vibrating string
- Sampled traveling waves
- Terminated string
- Plucked and struck string
- Damping and dispersion
- String Loop Identification
- Nonlinear "overdrive" distortion

*Work supported by the Wallenberg Global Learning Network

Ideal Vibrating String



Wave Equation

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

$$\begin{array}{llll} K & \stackrel{\Delta}{=} \text{ string tension} & y & \stackrel{\Delta}{=} y(t,x) \\ \epsilon & \stackrel{\Delta}{=} \text{ linear mass density} & \dot{y} & \stackrel{\Delta}{=} \frac{\partial}{\partial t}y(t,x) \\ y & \stackrel{\Delta}{=} \text{ string displacement} & y' & \stackrel{\Delta}{=} \frac{\partial}{\partial x}y(t,x) \end{array}$$

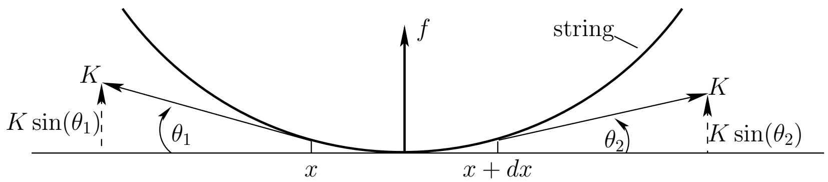
Newton's second law

$$\boxed{\text{Force} = \text{Mass} \times \text{Acceleration}}$$

Assumptions

- Lossless
- Linear
- Flexible (no "Stiffness")
- Slope $y'(t,x) \ll 1$

String Wave Equation Derivation



Force diagram for length dx string element

Total upward force on length dx string element:

$$\begin{aligned}
f(x + dx/2) &= K\sin(\theta_1) + K\sin(\theta_2) \\
&\approx K\left[\tan(\theta_1) + \tan(\theta_2)\right] \\
&= K\left[-y'(x) + y'(x + dx)\right] \\
&\approx K\left[-y'(x) + y'(x) + y''(x)dx\right] \\
&= Ky''(x)dx
\end{aligned}$$

Mass of length dx string segment: $m = \epsilon\, dx$.

By Newton's law, $f = ma = m\ddot{y}$, we have

$$Ky''(t,x)dx = (\epsilon\, dx)\ddot{y}(t,x)$$

or

$$\boxed{Ky''(t,x) = \epsilon\ddot{y}(t,x)}$$

Traveling-Wave Solution

One-dimensional lossless wave equation:

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

Plug in *traveling wave to the right*:

$$\begin{aligned}
y(t,x) &= y_r(t - x/c) \\
\Rightarrow \quad y'(t,x) &= -\frac{1}{c}\dot{y}(t,x) \\
y''(t,x) &= \frac{1}{c^2}\ddot{y}(t,x)
\end{aligned}$$

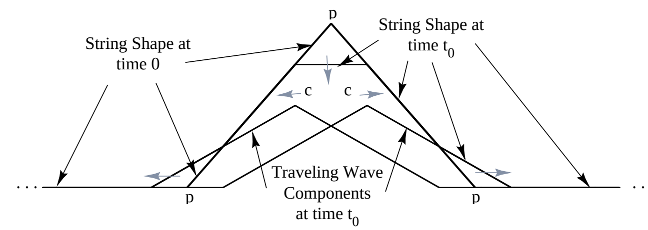
- Given $c \stackrel{\Delta}{=} \sqrt{K/\epsilon}$, the wave equation is satisfied for *any shape traveling to the right at speed* c (but remember slope $\ll 1$)
- Similarly, any *left-going* traveling wave at speed c, $y_l(t + x/c)$, statisfies the wave equation (show)

- General solution to lossless, 1D, second-order wave equation:

$$y(t,x) = y_r(t - x/c) + y_l(t + x/c)$$

- $y_l(\cdot)$ and $y_r(\cdot)$ are arbitrary twice-differentiable functions (slope $\ll 1$)
- **Important point:** Function of two variables $y(t,x)$ is replaced by two functions of a single (time) variable $\Rightarrow$ *reduced computational complexity*.
- Published by d'Alembert in 1747
(wave equation itself introduced in same paper)

Infinitely long string plucked simultaneously at three points marked 'p'



- Initial displacement = sum of two identical triangular pulses
- At time t_0, traveling waves centers are separated by $2ct_0$ meters
- String is not moving where the traveling waves overlap at same slope.
- Animation[1]

[1]http://ccrma.stanford.edu/ jos/rsadmin/TravellingWaveApp.swf

Sampled Traveling Waves in a String

For discrete-time simulation, we must *sample* the traveling waves

- Sampling interval $\stackrel{\Delta}{=} T$ seconds
- Sampling rate $\stackrel{\Delta}{=} f_s$ Hz $= 1/T$
- Spatial sampling interval $\stackrel{\Delta}{=} X$ m/s $\stackrel{\Delta}{=} cT$
  $\Rightarrow$ *systolic grid*

For a vibrating string with length L and fundamental frequency f_0,

$$c = f_0 \cdot 2L \qquad \left(\frac{\text{meters}}{\text{sec}} = \frac{\text{periods}}{\text{sec}} \cdot \frac{\text{meters}}{\text{period}}\right)$$

so that

$$X = cT = (f_0 2L)/f_s = L[f_0/(f_s/2)]$$

Thus, the number of *spatial samples* along the string is

$$L/X = (f_s/2)/f_0$$

or

Number of spatial samples = Number of string harmonics

Examples:

- Spatial sampling interval for (1/2) CD-quality digital model of Les Paul electric guitar (strings $\approx$ 26 inches)
  - $X = Lf_0/(f_s/2) = L82.4/22050 \approx 2.5$ mm for low E string
  - $X \approx 10$ mm for high E string (two octaves higher and the same length)
  - Low E string: $(f_s/2)/f_0 = 22050/82.4 = 268$ harmonics (spatial samples)
  - High E string: 67 harmonics (spatial samples)
- Number of harmonics = number of oscillators required in *additive synthesis*
- Number of harmonics = number of two-pole filters required in *subtractive, modal*, or *source-filter decomposition synthesis*
- Digital waveguide model needs only *one delay line* (length $2L$)

Examples (continued):

- Sound propagation in *air*:
  - Speed of sound $c \approx 331$ meters per second
  - $X = 331/44100 = 7.5$ mm
  - Spatial sampling rate $= \nu_s = 1/X = 133$ samples/m
  - Sound speed in air is *comparable* to that of transverse waves on a guitar string (faster than some strings, slower than others)
  - Sound travels much faster in most solids than in air
  - Longitudinal waves in strings travel faster than transverse waves
    - typically an order of magnitude faster

Sampled Traveling Waves in any Digital Waveguide

$$\begin{array}{rcl} x & \rightarrow & x_m = mX \\ t & \rightarrow & t_n = nT \end{array}$$

$\Rightarrow$

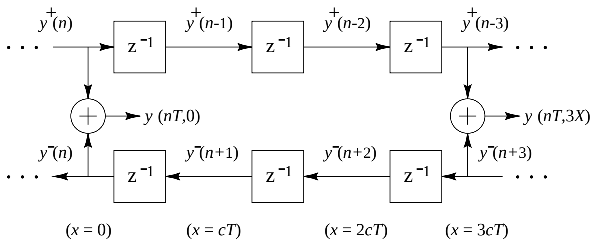
$$\begin{aligned} y(t_n, x_m) &= y_r(t_n - x_m/c) + y_l(t_n + x_m/c) \\ &= y_r(nT - mX/c) + y_l(nT + mX/c) \\ &= y_r\left[(n-m)T\right] + y_l\left[(n+m)T\right] \\ &= y^+(n-m) + y^-(n+m) \end{aligned}$$

where we defined

$$y^+(n) \triangleq y_r(nT) \qquad\qquad y^-(n) \triangleq y_l(nT)$$

- "$+$" superscript $\Longrightarrow$ *right-going*
- "$-$" superscript $\Longrightarrow$ *left-going*
- $y_r\left[(n-m)T\right] = y^+(n-m) =$ output of m-sample delay line with input $y^+(n)$
- $y_l\left[(n+m)T\right] \triangleq y^-(n+m) =$ *input* to an m-sample delay line whose *output* is $y^-(n)$

Lossless digital waveguide with observation points at $x = 0$ and $x = 3X = 3cT$



- Recall:

$$\begin{aligned} y(t,x) &= y^+\left(\frac{t - x/c}{T}\right) + y^-\left(\frac{t + x/c}{T}\right) \\ &\downarrow \\ y(nT, mX) &= y^+(n-m) + y^-(n+m) \end{aligned}$$

- *Position* $x_m = mX = mcT$ is *eliminated* from the simulation
- Position x_m remains laid out from left to right
- Left- and right-going traveling waves must be *summed* to produce a *physical* output

$$y(t_n, x_m) = y^+(n-m) + y^-(n+m)$$

- Similar to *ladder* and *lattice digital filters*

Important point: Discrete time simulation is *exact* at the sampling instants, to within the numerical precision of the samples themselves.

To avoid *aliasing* associated with sampling:

- Require all initial waveshapes be *bandlimited* to $(-f_s/2, f_s/2)$
- Require all external driving signals be similarly bandlimited
- Avoid nonlinearities or keep them "weak"
- Avoid time variation or keep it slow
- Use plenty of lowpass filtering with rapid high-frequency roll-off in severely nonlinear and/or time-varying cases
- Prefer "feed-forward" over "feed-back" around nonlinearities and/or modulations when possible

Interactive Java simulation of a vibrating string:
`http://www.colorado.edu/physics/phet/simulations/stringwave/-stringWave.swf`

Other Wave Variables

Velocity Waves:

$$v^+(n) \triangleq \dot{y}^+(n)$$
$$v^-(n) \triangleq \dot{y}^-(n)$$

Wave Impedance (we'll derive later):

$$\boxed{R = \sqrt{K\epsilon} = \frac{K}{c} = \epsilon c}$$

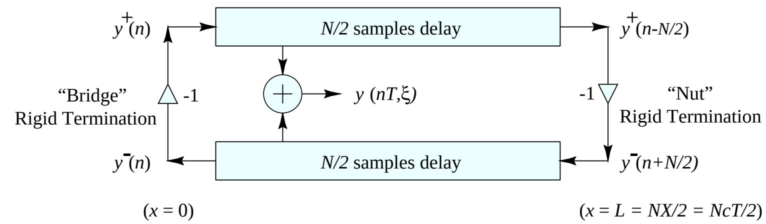
Force Waves:

$$f^+(n) \triangleq R\,v^+(n)$$
$$f^-(n) \triangleq -R\,v^-(n)$$

Ohm's Law for Traveling Waves:

$$\boxed{\begin{aligned} f^+(n) &= \quad R\,v^+(n) \\ f^-(n) &= -\,R\,v^-(n) \end{aligned}}$$

Rigidly Terminated Ideal String



- Reflection *inverts* for displacement, velocity, or acceleration waves (proof below)
- Reflection *non-inverting* for slope or force waves

Boundary conditions:

$$y(t,0) \equiv 0 \qquad y(t,L) \equiv 0 \qquad (L = \text{string length})$$

Expand into Traveling-Wave Components:

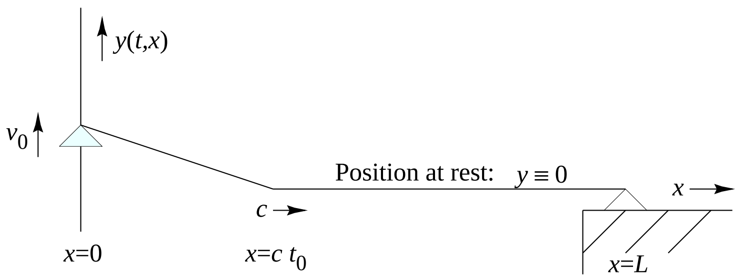
$$\begin{aligned} y(t,0) &= y_r(t) + y_l(t) = y^+(t/T) + y^-(t/T) \\ y(t,L) &= y_r(t - L/c) + y_l(t + L/c) \end{aligned}$$

Solving for outgoing waves gives

$$\begin{aligned} y^+(n) &= -y^-(n) \\ y^-(n + N/2) &= -y^+(n - N/2) \end{aligned}$$

$N \triangleq 2L/X =$ *round-trip* propagation time in samples

Moving Termination: Ideal String



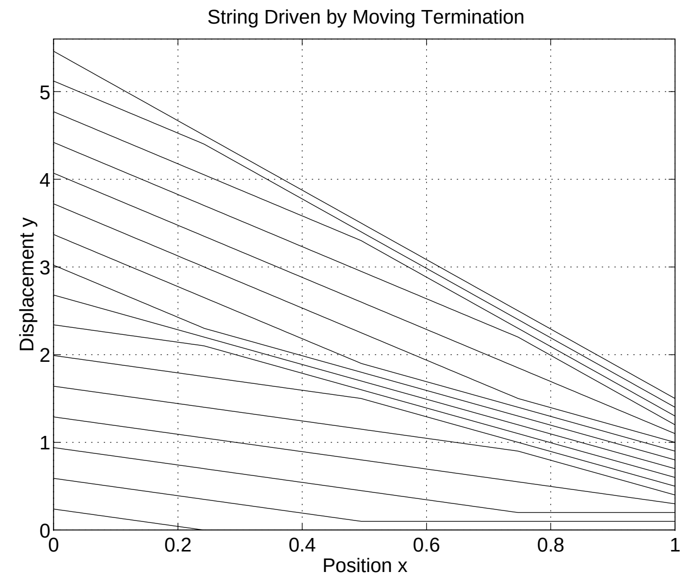
Uniformly moving rigid termination for an ideal string (tension K, mass density ϵ) at time $0 < t_0 < L/c$.

Driving-Point Impedance:

$$y'(t,0) \;=\; -\frac{v_0 t_0}{c t_0} = -\frac{v_0}{c} = -\frac{v_0}{\sqrt{K/\epsilon}}$$

$$\Rightarrow \quad f_0 \;=\; -K\sin(\theta) \approx -K y'(t,0) = \sqrt{K\epsilon}\, v_0 \stackrel{\Delta}{=} R v_0$$

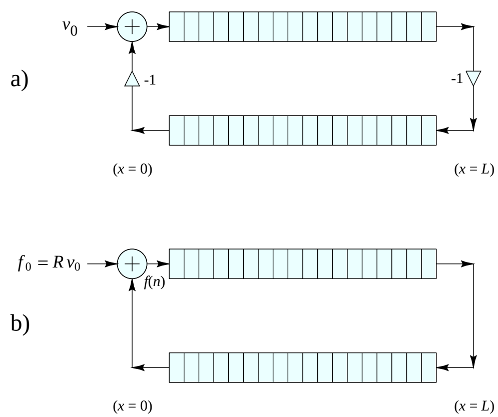
- If the left endpoint moves with constant velocity v_0 then the external applied force is $f_0 = R v_0$
- $R \stackrel{\Delta}{=} \sqrt{K\epsilon} \stackrel{\Delta}{=}$ *wave impedance* (for transverse waves)
- Equivalent circuit is a *resistor* (dashpot) $R > 0$
- We have the simple relation $f_0 = R v_0$ only in the *absence of return waves*, i.e., until time $t_0 = 2L/c$.



- Successive snapshots of the ideal string with a uniformly moving rigid termination
- Each plot is offset slightly higher for clarity
- GIF89A animation at

  `http://ccrma.stanford.edu/~jos/swgt/movet.html`

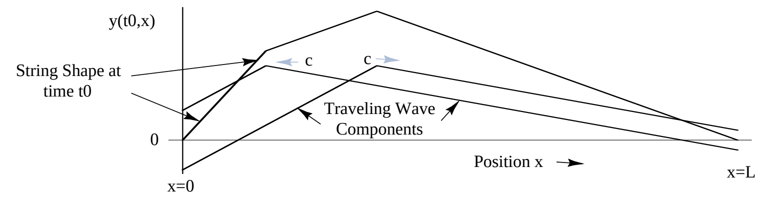
Waveguide "Equivalent Circuits" for the Uniformly Moving Rigid String Termination



a) Velocity waves   b) Force waves

- String moves with speed v_0 or 0 only
- String is always one or two straight segments
- "Helmholtz corner" (slope discontinuity) shuttles back and forth at speed c
- String slope increases without bound
- Applied force at termination steps up to infinity
  - Physical string force is labeled $f(n)$
  - $f_0 = Rv_0 =$ *incremental* force per period

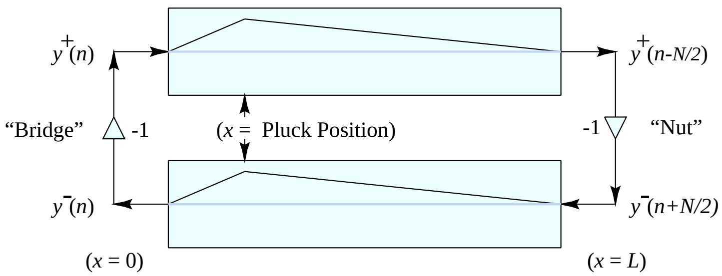
Doubly Terminated Ideal Plucked String



A doubly terminated string, "plucked" at 1/4 its length.

- Shown short time after pluck event.
- Traveling-wave components and physical string-shape shown.
- Note traveling-wave components sum to zero at terminations. (Use image method.)

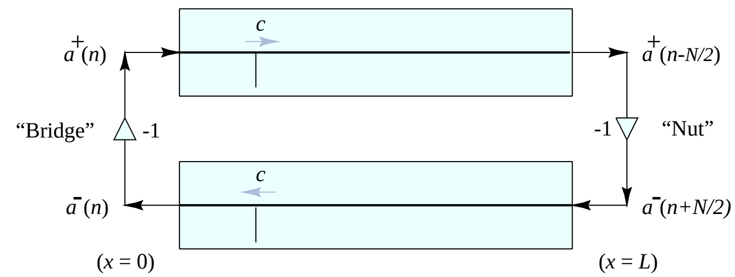
Digital Waveguide Plucked-String Model Using Initial Conditions



Initial conditions for the ideal plucked string.

- Amplitude of each traveling-wave $= 1/2$ initial string displacement.
- Sum of the upper and lower delay lines $=$ initial string displacement.

Acceleration-Wave Simulation



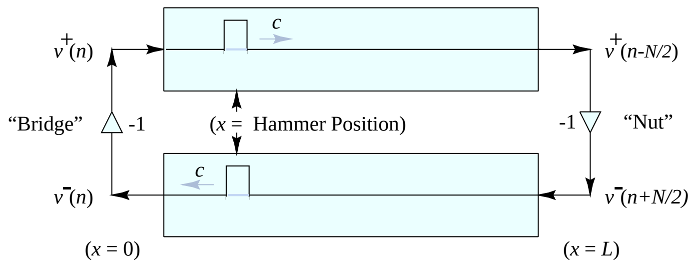
Initial conditions for the ideal plucked string: acceleration or curvature waves.

Recall:

$$\boxed{y'' = \frac{1}{c^2}\ddot{y}}$$

Acceleration waves are proportional to “curvature” waves.

Ideal Struck-String Velocity-Wave Simulation

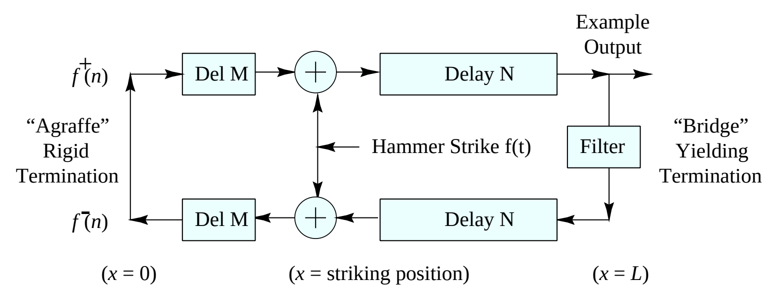


Initial conditions for the ideal struck string in a *velocity wave* simulation.

Hammer strike = *momentum transfer* = velocity step:

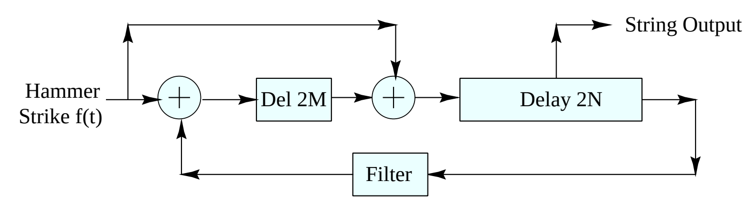
$$m_h v_h(0-) = (m_h + m_s) v_s(0+)$$

External String Excitation at a Point



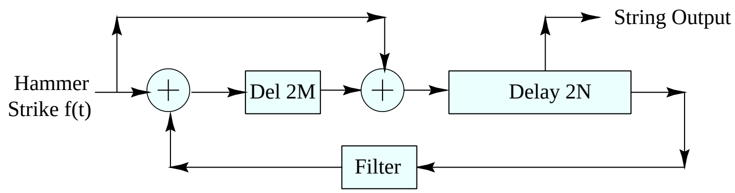
"Waveguide Canonical Form"

Equivalent System: Delay Consolidation

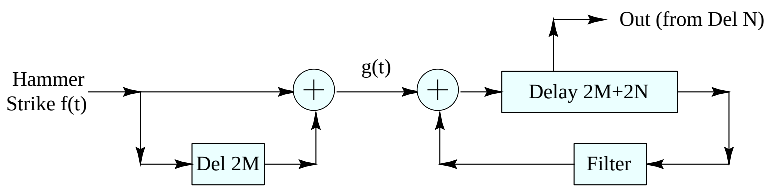


Finally, we "pull out" the comb-filter component:

Delay Consolidated System (Repeated):

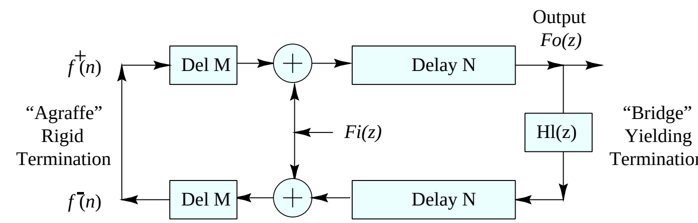


Equivalent System: FFCF Factored Out:



- Extra memory needed.
- Output "tap" can be moved to delay-line output.

Algebraic Derivation

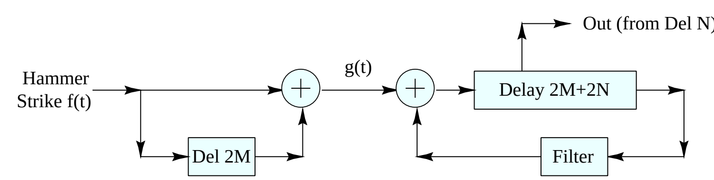


By inspection:

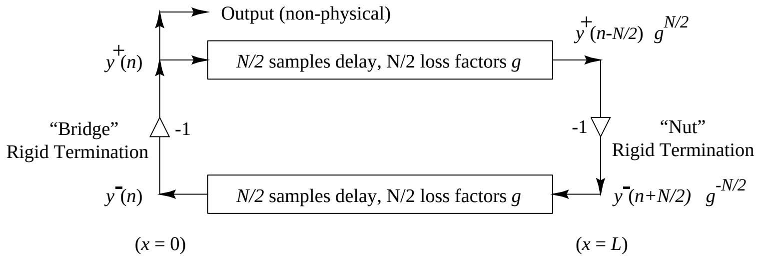
$$F_o(z) = z^{-N}\left\{F_i(z) + z^{-2M}\left[F_i(z) + z^{-N}H_l(z)F_o(z)\right]\right\}$$

$$\Rightarrow \quad H(z) \triangleq \frac{F_o(z)}{F_i(z)} = z^{-N}\frac{1+z^{-2M}}{1-z^{-(2M+2N)}}$$

$$= \left(1+z^{-2M}\right)\frac{z^{-N}}{1-z^{-(2M+2N)}}$$



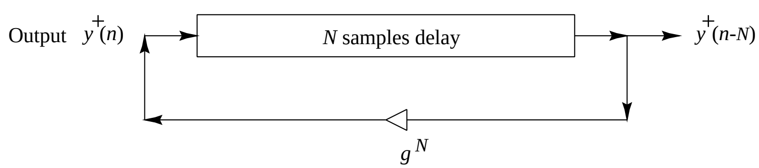
Damped Plucked String



Rigidly terminated string with distributed resistive losses.

- N loss factors g are embedded between the delay-line elements.

Equivalent System: Gain Elements Commuted



All N loss factors g have been "pushed" through delay elements and combined at a *single* point.

Computational Savings

- $f_s = 50$kHz, $f_1 = 100Hz \Rightarrow$ delay $= 500$
- Multiplies reduced by two orders of magnitude
- Input-output transfer function unchanged
- Round-off errors reduced

Frequency-Dependent Damping

- Loss factors g should really be digital filters
- Gains in nature typically decrease with frequency
- Loop gain may not exceed 1 (for stability)
- Gain filters commute with delay elements (LTI)
- Typically only *one* gain filter used per loop

Simplest Frequency-Dependent Loop Filter

$$\boxed{H_l(z) = b_0 + b_1 z^{-1}}$$

- Linear phase $\Rightarrow b_0 = b_1$ ($\Rightarrow$ delay = 1/2 sample)
- Zero damping at dc $\Rightarrow b_0 + b_1 = 1$
  $\Rightarrow b_0 = b_1 = 1/2$
  $\Rightarrow$
  $$\boxed{H_l(e^{j\omega T}) = \cos(\omega T/2), \quad |\omega| \leq \pi f_s}$$

Next Simplest Case: Length 3 FIR Loop Filter

$$\boxed{H_l(z) = b_0 + b_1 z^{-1} + b_2 z^{-2}}$$

- Linear phase $\Rightarrow b_0 = b_2$ ($\Rightarrow$ delay = 1 sample)
- Unity dc gain $\Rightarrow b_0 + b_1 + b_2 = 2b_0 + b_1 = 1 \Rightarrow$
  $$H_l(e^{j\omega T}) = e^{-j\omega T}\left[(1-2b_0) + 2b_0\cos(\omega T)\right]$$
- Remaining degree of freedom = *damping control*

Length 3 FIR Loop Filter with Variable DC Gain

Have two degrees of freedom for *brightness* & *sustain*:

$$
\begin{aligned}
g_0 &\stackrel{\Delta}{=} e^{-6.91P/S} \\
b_0 &= g_0(1-B)/4 = b_2 \\
b_1 &= g_0(1+B)/2
\end{aligned}
$$

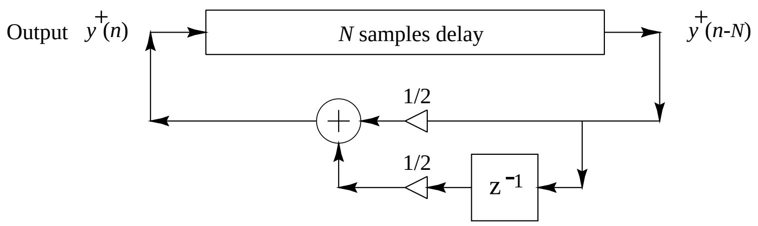
where

$$
\begin{aligned}
P &= \text{period in seconds (total loop delay)} \\
S &= \text{desired sustain time in seconds} \\
B &= \text{brightness parameter in the interval } [0,1]
\end{aligned}
$$

Sustain time S is defined here as the time to decay 60 dB (or 6.91 time-constants) when brightness B is maximum ($B=1$). At minimum brightness ($B=0$), we have

$$
|H_l(e^{j\omega T})| = g_0 \frac{1+\cos(\omega T)}{2} = g_0 \cos^2(\omega T)
$$

Karplus-Strong Algorithm



- To play a note, the delay line is initialized with random numbers ("white noise")

Interpretations of the Karplus-Strong Algorithm

The Karplus-Strong structure can be *interpreted* as a

- *pitch prediction filter* from the Codebook-Excited Linear Prediction (CELP) standard (*periodic LPC* synthesis)
- *feedback comb filter* with *lowpassed feedback* used earlier by James A. Moorer for recursively modeling *wall-to-wall echoes* ("About This Reverberation Business")
- simplified digital waveguide model

KS Physical Interpretation

- Rigidly terminated ideal string with the simplest damping filter
- Damping consolidated at one point and replaced by a one-zero filter approximation
- String *shape* $=$ *sum* of upper and lower delay lines
- String *velocity* $=$ spatial integral of the *difference* of upper and lower delay lines:

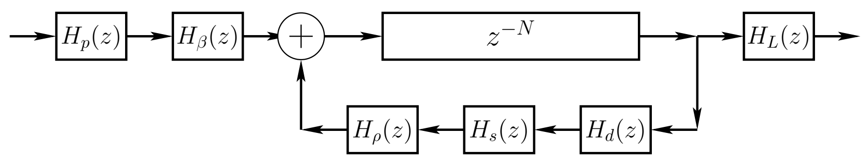
$$\begin{aligned} s \triangleq y' &= \frac{1}{c}(v_l - v_r) \\ \Rightarrow \quad y(t,x) &= \frac{1}{c}\int_0^x \left[v_l\left(t + \frac{\xi}{c}\right) - v_r\left(t - \frac{\xi}{c}\right)\right] d\xi \end{aligned}$$

- Karplus-Strong string is both "plucked" and "struck" by random amounts along entire length of string!

KS Sound Examples

- "Vintage" 8-bit sound examples:
  - Original Plucked String: (AIFF) (MP3)
  - Drum: (AIFF) (MP3)
  - Stretched Drum: (AIFF) (MP3)
- STK Plucked String: (WAV) (MP3)
  - Plucked String 1: (WAV) (MP3)
  - Plucked String 2: (WAV) (MP3)
  - Plucked String 3: (WAV) (MP3)
  - Plucked String 4: (WAV) (MP3)

Extended Karplus-Strong (EKS) Algorithm



$$\begin{aligned}
N &= \text{pitch period } (2\times \text{ string length) in samples} \\
H_p(z) &= \frac{1-p}{1-p\,z^{-1}} = \text{pick-direction lowpass filter} \\
H_\beta(z) &= 1 - z^{-\beta N} = \text{pick-position comb filter, } \beta \in (0,1) \\
H_d(z) &= \text{string-damping filter (one/two poles/zeros typical)} \\
H_s(z) &= \text{string-stiffness allpass filter (several poles and zeros)} \\
H_\rho(z) &= \frac{\rho(N) - z^{-1}}{1-\rho(N)\,z^{-1}} = \text{first-order string-tuning allpass filter} \\
H_L(z) &= \frac{1-R_L}{1-R_L\,z^{-1}} = \text{dynamic-level lowpass filter}
\end{aligned}$$

EKS Sound Example

Bach A-Minor Concerto—Orchestra Part: (WAV) (MP3)

- Executes in real time on one Motorola DSP56001 (20 MHz clock, 128K SRAM)
- Developed for the NeXT Computer introduction at Davies Symphony Hall, San Francisco, 1989
- Solo violin part was played live by Dan Kobialka of the San Francisco Symphony

Loop Filter Identification

For loop-filter design, we wish to minimize the error in

- partial decay time (set by amplitude response)
- partial overtone tuning (set by phase response)

Simple and effective method:

- Estimate pitch (elaborated next page)
- Set Hamming FFT-window length to four periods
- Compute the short-time Fourier transform (STFT)
- Detect peaks in each spectral frame
- Connect peaks through time (amplitude envelopes)
- Amplitude envelopes must decay *exponentially*
- On a dB scale, exponential decay is a *straight line*
- Slope of straight line determines decay time-constant
- Can use 1st-order `polyfit` in Matlab or Octave
- For beating decay, connect amplitude envelope peaks
- Decay rates determine ideal amplitude response
- Partial tuning determines ideal phase response

Plucked/Struck String Pitch Estimation

- Take FFT of middle third of plucked string tone
- Detect spectral peaks
- Form histogram of peak spacing Δf_i
- Pitch estimate $\hat{f}_0 \stackrel{\Delta}{=}$ most common spacing Δf_i
- Refine $\hat{f}_0$ with gradient search using harmonic comb:

$$\begin{aligned} \hat{f}_0 &\stackrel{\Delta}{=} \arg\max_{\hat{f}_0} \sum_{i=1}^{K} \log \left| X(k_i \hat{f}_0) \right| \\ &= \arg\max_{\hat{f}_0} \prod_{i=1}^{K} \left| X(k_i \hat{f}_0) \right| \end{aligned}$$

where

K = number of peaks, and
k_i = estimated harmonic number of ith peak
(valid method for non-stiff strings)

Must skip over any missing harmonics,
i.e., omit k_i whenever $|X(k_i \hat{f}_0)| \approx 0$.

References: For pointers to research literature, see
`http://ccrma.stanford.edu/~jos/jnmr/Model_Parameter_Estimation.html`

Nonlinear "Overdrive"

A popular type of distortion, used in *electric guitars*, is *clipping* of the guitar waveform.

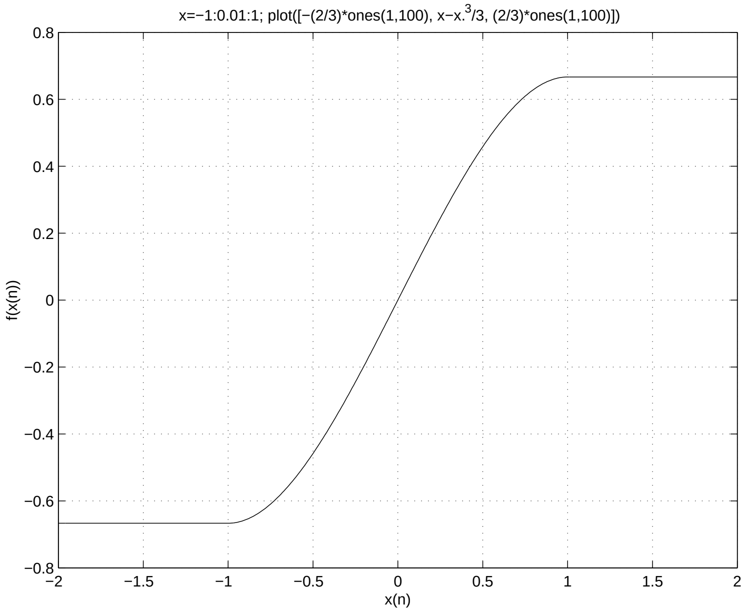
Hard Clipper

$$f(x) = \begin{cases} -1, & x \le -1 \\ x, & -1 \le x \le 1 \\ 1, & x \ge 1 \end{cases}$$

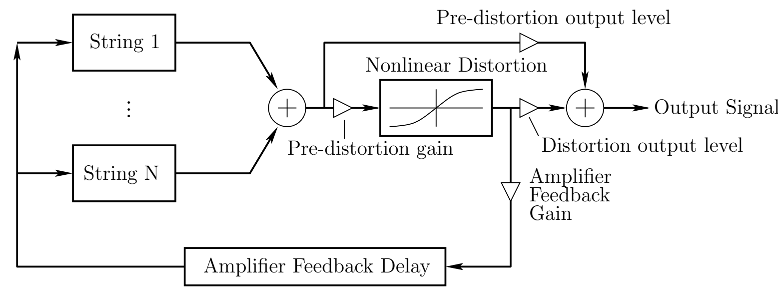
where x denotes the current input sample $x(n)$, and $f(x)$ denotes the output of the nonlinearity.

Soft Clipper

$$f(x) = \begin{cases} -\frac{2}{3}, & x \le -1 \\ x - \frac{x^3}{3}, & -1 \le x \le 1 \\ \frac{2}{3}, & x \ge 1 \end{cases}$$



Amplifier Distortion + Amplifier Feedback



Simulation of a basic distorted electric guitar with amplifier feedback.

- Distortion should be preceded and followed by *EQ*
  Simple example: integrator pre, differentiator post
- Distortion output signal often further filtered by an *amplifier cabinet filter*, representing speaker cabinet, driver responses, etc.
- In Class A tube amplifiers, there should be *duty-cycle modulation* as a function of signal level[2]
  - 50% at low levels (no duty-cycle modulation)
  - 55-65% duty cycle observed at high levels
    $\Rightarrow$ even harmonics come in
  - Example: Distortion input can *offset by a constant*
    (e.g., input RMS level times some scaling)

[2] See `http://www.trueaudio.com/at_eetjlm.htm` for further discussion.