MUS420/EE367A Lecture 6
(last used in 2001 - superceded by SimpleStrings)
Distributed Modeling in Discrete Time

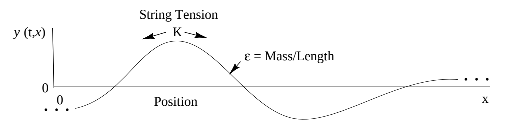
Julius O. Smith III (jos@ccrma.stanford.edu)
Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
Stanford, California 94305

March 12, 2008

Outline

- Ideal Vibrating String
- Finite Difference Approximation (FDA)
- Traveling-Wave Solution
- Ideal Plucked String
- Sampling Issues
- Lossless Digital Waveguides
- Sampled Traveling Waves versus Finite Differences
- Lossy 1D Wave Equation
- Dispersive 1D Wave Equation

Ideal Vibrating String



Wave Equation

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

$$\begin{aligned} K &\triangleq \text{string tension} & y &\triangleq y(t,x) \\ \epsilon &\triangleq \text{linear mass density} & \dot{y} &\triangleq \frac{\partial}{\partial t} y(t,x) \\ y &\triangleq \text{string displacement} & y' &\triangleq \frac{\partial}{\partial x} y(t,x) \end{aligned}$$

Newton's second law

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

Assumptions

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

Example One-Dimensional Waveguides

- Any elastic medium displaced along 1D
- Air column of a clarinet or organ pipe
  - Air-pressure deviation $p \leftrightarrow$ string displacement y
  - Longitudinal volume velocity $u \leftrightarrow$ transverse string velocity v
- Vibrating strings
  - Really need at least *three* coupled 1D waveguides:
    * Horizontally polarized transverse waves
    * Vertical polarized transverse waves
    * Longitudinal waves

    (Typically 1 or 2 WG per string used in practice)
  - Bowed strings also require *torsional waves* (Typical: one waveguide per string [plane of the bow])
  - Piano requires up to three coupled strings per key
    * Two-stage decay
    * Aftersound

    (Typical: 1 or 2 waveguides per string)

Let's first review the finite difference approximation applied to the ideal string (for comparison purposes):

Finite Difference Approximation (FDA)

$$\dot{y}(t,x) \approx \frac{y(t,x) - y(t-T,x)}{T}$$

and

$$y'(t,x) \approx \frac{y(t,x) - y(t,x-X)}{X}$$

- T = temporal sampling interval
- X = spatial sampling interval
- Exact in limit as sampling intervals $\rightarrow$ zero
- Half a sample delay at each frequency.
  Fix: $\dot{y}(t,x) \approx [y(t+T,x) - y(t-T,x)]/(2T)$

Zero-phase second-order difference:

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

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

- All odd-order derivative approximations suffer a half-sample delay error
- All even order cases can be compensated as above

FDA of 1D Wave Equation

Substituting finite difference approximation (FDA) into the wave equation $Ky'' = \epsilon\ddot{y}$ gives

$$K\frac{y(t,x+X) - 2y(t,x) + y(t,x-X)}{X^2}$$
$$= \epsilon\frac{y(t+T,x) - 2y(t,x) + y(t-T,x)}{T^2}$$

$\Rightarrow$ Time Update:

$$y(t+T,x) = \frac{KT^2}{\epsilon X^2}\left[y(t,x+X) - 2y(t,x) + y(t,x-X)\right] + 2y(t,x) - y(t-T,x)$$

Let $c \stackrel{\Delta}{=} \sqrt{K/\epsilon}$ (speed of sound along the string).
In practice, we typically normalize such that

- $T = 1 \Rightarrow t = nT = n$
- $X = cT = 1 \Rightarrow x = mX = m$, and

$$\boxed{y(n+1,m) = y(n,m+1) + y(n,m-1) - y(n-1,m)}$$

- Recursive *difference equation* in two variables (time and space)

- Time-update recursion for time $n+1$ requires *all* values of string displacement (i.e., all m), for the two preceding time steps (times n and $n-1$)
- Recursion typically started by assuming zero past displacement: $y(n,m) = 0, n = -1, 0, \forall m$.
- Higher order wave equations yield more terms of the form $y(n-l, m-k) \Leftrightarrow$ frequency-dependent *losses* and/or *dispersion* characteristics are introduced into the FDA:
- Linear differential equations with constant coefficients give rise to some linear, time-invariant discrete-time system via the FDA
  - Linear, time-invariant, "filtered waveguide" case:

$$\boxed{\sum_{k=0}^{\infty}\alpha_k\frac{\partial^k y(t,x)}{\partial t^k} = \sum_{l=0}^{\infty}\beta_l\frac{\partial^l y(t,x)}{\partial x^l}}$$

  - More general linear, time-invariant case

$$\sum_{k=0}^{\infty}\sum_{l=0}^{\infty}\alpha_{k,l}\frac{\partial^k\partial^l y(t,x)}{\partial t^k\partial x^l} = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\beta_{m,n}\frac{\partial^m\partial^n y(t,x)}{\partial t^m\partial x^n}$$

  - Nonlinear example:

$$\frac{\partial y(t,x)}{\partial t} = \left(\frac{\partial y(t,x)}{\partial x}\right)^2$$

  - Time-varying example:

$$\frac{\partial y(t,x)}{\partial t} = t^2\frac{\partial y(t,x)}{\partial x}$$

Traveling-Wave Solution

One-dimensional lossless wave equation:

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

Plug in *traveling wave to the right*:

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

- Since $c \stackrel{\Delta}{=} \sqrt{K/\epsilon}$, 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, $y_l(t + x/c)$, satisfies the wave equation

- 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 complexity*.
- Published by d'Alembert in 1747

Laplace-Domain Analysis

- e^{st} is an *eigenfunction* under differentiation
- Plug it in:
$$y(t,x) = e^{st+vx}$$
- By *differentiation theorem*
$$\begin{aligned} \dot{y} &= sy & y' &= vy \\ \ddot{y} &= s^2 y & y'' &= v^2 y \end{aligned}$$
- Wave equation becomes
$$\begin{aligned} Kv^2 y &= \epsilon s^2 y \\ \Rightarrow \frac{s^2}{v^2} &= \frac{K}{\epsilon} = c^2 \\ \Rightarrow v &= \pm \frac{s}{c} \end{aligned}$$

Thus
$$y(t,x) = e^{s(t \pm x/c)}$$
is a solution for all s.

Interpretation: left- and right-going exponentially enveloped complex sinusoids

General eigensolution:

$$y(t,x) = e^{s(t \pm x/c)}, \quad s \text{ arbitrary, complex}$$

By *superposition,*

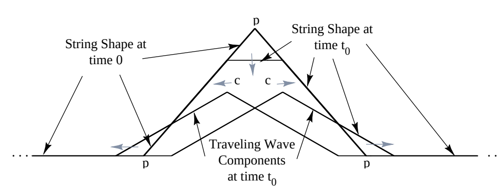
$$y(t,x) = \sum_i A^+(s_i) e^{s_i(t-x/c)} + A^-(s_i) e^{s_i(t+x/c)}$$

is also a solution for all $A^+(s_i)$ and $A^-(s_i)$.

Alternate derivation of D'Alembert's solution:

- Specialize general eigensolution to $s \stackrel{\Delta}{=} j\omega$
- Extend summation to an integral over ω $\Rightarrow$ *Inverse Fourier transform* gives
$$y(t,x) = y_r\left(t - \frac{x}{c}\right) + y_l\left(t + \frac{x}{c}\right)$$
where $y_r(\cdot)$ and $y_l(\cdot)$ are arbitrary continuous functions

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.

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 rate $\stackrel{\Delta}{=} X$ m/s $\stackrel{\Delta}{=} cT$
  $\Rightarrow$ *systolic grid*

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

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

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 long)
  - $X = Lf_0/(f_s/2) = L82.4/22050 \approx 2.5$ 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*

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

Sampled Traveling Waves in any Digital Waveguide

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

$\Rightarrow$

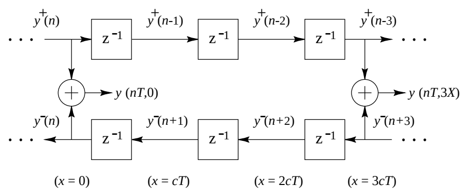
$$\begin{aligned} y(t_n, x_m) &= y_r(t_n - x_m/c) + y_l(t_n + x_m/c) \\ &= y_r(nT - mX/c) + y_l(nT + mX/c) \\ &= y_r\left[(n-m)T\right] + y_l\left[(n+m)T\right] \\ &= y^+(n-m) + y^-(n+m) \end{aligned}$$

where we defined

$$y^+(n) \stackrel{\Delta}{=} y_r(nT) \qquad\qquad y^-(n) \stackrel{\Delta}{=} y_l(nT)$$

- "+" 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 $y^+(n)$
- $y_l\left[(n+m)T\right] \stackrel{\Delta}{=} y^-(n+m)$ = *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 when possible

Relation of Sampled D'Alembert Traveling Waves to the Finite Difference Approximation

Recall FDA result [based on $\dot{x}(n) \approx x(n) - x(n-1)$]:

$$y(n+1,m) = y(n,m+1) + y(n,m-1) - y(n-1,m)$$

Traveling-wave decomposition (exact in lossless case at sampling instants):

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

Substituting into FDA gives

$$\begin{aligned} y(n+1,m) \;&=\; y(n,m+1) + y(n,m-1) - y(n-1,m) \\ &=\; y^+(n-m-1) + y^-(n+m+1) \\ &\quad\; +y^+(n-m+1) + y^-(n+m-1) \\ &\quad\; -y^+(n-m-1) - y^-(n+m-1) \\ &=\; y^-(n+m+1) + y^+(n-m+1) \\ &=\; y^+[(n+1)-m] + y^-[(n+1)+m] \\ &\stackrel{\Delta}{=}\; y(n+1,m) \end{aligned}$$

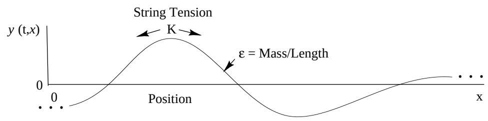
- FDA recursion is also *exact* in the lossless case (!)
- Recall that FDA introduced artificial damping in mass-spring systems

- The last identity above can be rewritten as

$$\begin{aligned} y(n+1,m) \;&\stackrel{\Delta}{=}\; y^+[(n+1)-m] + y^-[(n+1)+m] \\ &=\; y^+[n-(m-1)] + y^-[n+(m+1)] \end{aligned}$$

- Displacement at time $n+1$ and position m is the superposition of left- and right-going components from positions $m-1$ and $m+1$ at time n
- The physical wave variable can be computed for the next time step as the sum of incoming traveling wave components from the left and right
- Lossless nature of the computation is clear

The Lossy 1D Wave Equation



The ideal vibrating string.

Sources of loss in a vibrating string:

1. Yielding terminations
2. Drag due to air viscosity
3. Internal bending friction

Simplest case: Add a term proportional to velocity:

$$Ky'' = \epsilon\ddot{y} \underbrace{+\mu\dot{y}}_{\text{new}}$$

More generally,

$$Ky'' = \epsilon\ddot{y} + \sum_{\substack{m=0 \\ m \text{ odd}}}^{M-1} \mu_m \frac{\partial^m y(t,x)}{\partial t^m}$$

where μ_m may be determined *indirectly* by *measuring* linear damping versus frequency

Solution to Lossy 1D Wave Equation

$$y(t,x) = e^{-(\mu/2\epsilon)x/c} y_r\,(t - x/c) + e^{(\mu/2\epsilon)x/c} y_l\,(t + x/c)$$

Assumptions:

- Small displacements ($y' \ll 1$)
- Small losses ($\mu \ll \epsilon\omega$)
- $c \stackrel{\Delta}{=} \sqrt{K/\epsilon} =$ as before (wave velocity in lossless case)

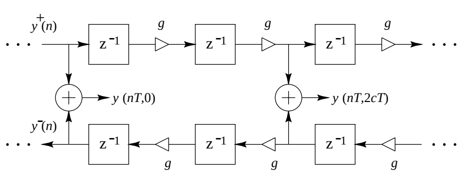
Components *decay exponentially* in direction of travel

Sampling with $t = nT$, $x = mX$, and $X = cT$ gives

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

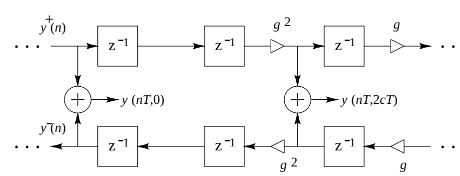
where $g \stackrel{\Delta}{=} e^{-\mu T/2\epsilon}$

Lossy Digital Waveguide



- Order ∞ distributed system reduced to finite order
- Loss factor $g = e^{-\mu T/2\epsilon}$ *summarizes* distributed loss in one sample of propagation
- Discrete-time simulation *exact* at sampling points
- Initial conditions and excitations must be *bandlimited*
- *Bandlimited interpolation* reconstructs continuous case

Loss Consolidation



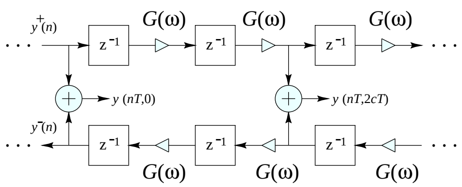
- Loss terms are simply *constant gains* $g \leq 1$
- Linear, time-invariant elements *commute*
- Applicable to *undriven* and *unobserved* string sections
- Simulation becomes *more accurate* at the outputs (fewer round-off errors)
- Number of multiplies *greatly reduced* in practice

Frequency-Dependent Losses

- Losses in nature tend to *increase* with frequency
  - Air absorption
  - Internal friction
- Simplest string wave equation giving higher damping at high frequencies

$$Ky'' = \epsilon\ddot{y} + \mu_1\dot{y} + \underbrace{+\mu_3\frac{\partial^3 y(t,x)}{\partial t^3}}_{\text{new}}$$

  - Used in Chaigne-Askenfelt piano string PDE
  - Damping asymptotically proportional to ω^2
- Waves propagate with frequency-dependent attenuation (zero-phase filtering)
- Loss consolidation remains valid (by commutativity)



The Dispersive One-Dimensional Wave Equation

Stiffness introduces a restoring force proportional to the fourth spatial derivative:

$$\epsilon\ddot{y} = Ky'' \underbrace{-\kappa y''''}_{\text{new}}$$

where

- $\kappa = \frac{Q\pi a^4}{4}$ (moment constant)
- $a =$ string radius
- $Q =$ Young's modulus (stress/strain) (spring constant for solids)
- Stiffness is a *linear* phenomenon
  - Imagine a "bundle" or "cable" of ideal string fibers
  - Stiffness is due to the *longitudinal* springiness

Limiting cases

- Reverts to *ideal flexible string* at very *low frequencies* ($Ky'' \gg \kappa y''''$)
- Becomes *ideal bar* at very *high frequencies* ($Ky'' \ll \kappa y''''$)

Effects of Stiffness

- *Phase velocity increases with frequency*

$$c(\omega) \stackrel{\Delta}{=} c_0\left(1 + \frac{\kappa\omega^2}{2Kc_0^2}\right)$$

  where $c_0 = \sqrt{K/\epsilon} =$ zero-stiffness phase velocity
- Note ideal-string (LF) and ideal-bar (HF) limits
- Traveling-wave components see a frequency-dependent sound speed
- High-frequency components "run out ahead" of low-frequency components ("HF precursors")
- Traveling waves "disperse" as they travel ("dispersive transmission line")
- String overtones are "stretched" and "inharmonic"
- Higher overtones are progressively sharper (Period$(\omega) = 2 \times$ Length / $c(\omega)$)
- *Piano strings* are *audibly stiff*

Reference: L. Cremer: **Physics of the Violin**

Digital Simulation of Stiff Strings

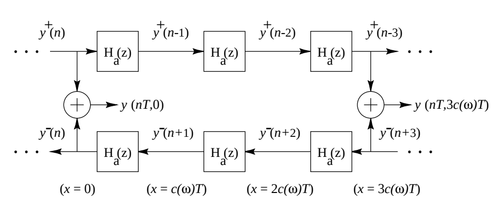
- *Allpass filters* implement a *frequency-dependent delay*
- For stiff strings, we must generalize $X = cT$ to

$$X = c(\omega)T \Rightarrow\ T(\omega) = X/c(\omega) = c_0T_0/c(\omega)$$

  where $T_0 = T(0) =$ zero-stiffness sampling interval
- Thus, replace unit delay z^{-1} by

$$z^{-1} \rightarrow z^{-c_0/c(\omega)} \stackrel{\Delta}{=} H_a(z) \quad \text{(frequency-dependent delay)}$$

- Each delay element becomes an *allpass filter*
- In general, $H_a(z)$ is *irrational*
- We approximate $H_a(z)$ in practice using some finite-order *fractional delay digital filter*



General Allpass Filters

- General, order L, allpass filter:

$$\begin{aligned} H_a(z) &\stackrel{\Delta}{=} z^{-L}\frac{A(z^{-1})}{A(z)} \\ &= \frac{a_L + a_{L-1}z^{-1} + \cdots + a_1 z^{-(L-1)} + z^{-L}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_L z^{-L}} \end{aligned}$$

- General order L, monic, minimum-phase polynomial:

$$A(z) \stackrel{\Delta}{=} 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_L z^{-L}$$

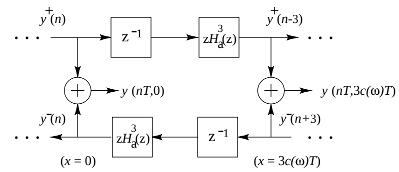
  where $A(z_i) = 0 \Rightarrow |z_i| < 1$ (roots inside unit circle)
- Numerator polynomial $=$ *reverse* of denominator
- First-order case:

$$H_a(z) \stackrel{\Delta}{=} \frac{a_1 z^{-1} + 1}{1 + a_1 z^{-1}}$$

- Each pole p_i gain-compensated by a zero at $z_i = 1/p_i$
- There are papers in the literature describing methods for designing allpass filters with a prescribed *group delay* (see reader for refs)
- For piano strings L is on the order of 10

Consolidation of Dispersion

Allpass filters are *linear and time invariant* which means they *commute* with other linear and time invariant elements



- At least one sample of pure delay must normally be "pulled out" of ideal desired allpass along each rail
- Ideal allpass design minimizes *phase-delay error* $P_c(\omega)$
- Minimizing $\| P_c(\omega) - c_0/c(\omega) \|_\infty$ approximately minimizes *tuning error* for modes of freely vibrating string (main audible effect)
- Minimizing *group delay* error optimizes *decay times*

Related Links

- Online draft of the book[1] containing this material
- Derivation of the wave equation for vibrating strings[2]

[1] http://ccrma.stanford.edu/~jos/waveguide/
[2] http://ccrma.stanford.edu/~jos/waveguide/String_Wave_Equation.html