Physical Modeling Sound Synthesis

Julius Smith
CCRMA, Stanford University

AES-2006 Masterclass

October 7, 2006
Overview

Early Voice Models
Digital Voice Models
Karplus Strong
Digital Waveguides
Single Reeds
Bowed Strings
Distortion Guitar
Committed Synthesis
Digital Waveguide Mesh
Horn Filter Design
Finite Differences
Virtual Analog
Wave Digital Filters
## Outline

**Synthesis approaches in approximate historical order, including selected research updates, for**

- Voice
- Plucked Strings
- Woodwinds
- Bowed Strings
- Piano and Harpsichord
- Membranes and Plates
- Horns
- Related Topics

**Reference:** *Physical Audio Signal Processing* (online book)
Early Voice Models
Early Talking Machines (Virtual Talking Heads)

Joseph Faber’s Euphonia
Early Brazen Heads (Wikipedia)

• ≈ 1125: First talking head description:
  William of Malmesbury’s History of the English Kings:
  ○ Pope Sylvester II said to have traveled to al-Andalus and stolen a tome of secret knowledge
  ○ Only able to escape through demonic assistance
  ○ Cast the head of a statue using his knowledge of astrology
  ○ It would not speak until spoken to, but then answered any yes/no question put to it

• Early devices were deemed *heretical* by the Church and often destroyed
• 1599: Albertus Magnus had a head with a human voice and breath:
  ○ "A certain reasoning process" bestowed by a *cacodemon* (evil demon)
  ○ Thomas Aquinas destroyed it for continually interrupting his ruminations (not everyone wants a yacking cacodemon around all the time)

• By the 18th century, talking machines became acceptable as “scientific pursuit”
Wolfgang Von Kempelin’s Speaking Machine (1791)

Replica of Wolfgang Von Kempelin’s Speaking Machine
Joseph Faber’s Euphonia (1846)

17 levers, a bellows, and a telegraphic line — sang “God Save the Queen”
Joseph Faber’s Euphonia (1846)
The Voder (Homer Dudley — 1939 Worlds Fair)

http://davidszondy.com/future/robot/voder.htm
Voder Keyboard

Voder Schematic

Fig. 8—Schematic circuit of the voder.

http://ptolemy.eecs.berkeley.edu/~eal/audio/voder.html
Voder Demos

Overview

Early Voice Models
- Virtual Heads
- Von Kempelin
- Euphonia
- Voder Keyboard
- Voder Schematic
- Voder Demos

Digital Voice Models
- Karplus Strong
- Digital Waveguides
- Single Reeds
- Bowed Strings
- Distortion Guitar
- Commuted Synthesis
- Digital Waveguide Mesh
- Horn Filter Design
- Finite Differences
- Virtual Analog
- Wave Digital Filters
Digital Voice Models

Overview
Early Voice Models
Digital Voice Models
Karplus Strong
Digital Waveguides
Single Reeds
Bowed Strings
Distortion Guitar
Committed Synthesis
Digital Waveguide Mesh
Horn Filter Design
Finite Differences
Virtual Analog
Wave Digital Filters
Kelly-Lochbaum Vocal Tract Model
(Discrete-Time Transmission-Line Model)

Overview

Early Voice Models

Digital Voice Models

- KL Voice
- "Daisy"
- "Shiela"
- Linear Prediction
- Glottal Model
- Source Estimation
- LF Glottal Model
- Phonation Variation
- Lu Sounds
- Tak
- 2D Vowels
- Pink Trombone

Karplus Strong

Digital Waveguides

- Single Reeds
- Bowed Strings
- Distortion Guitar
- Commuted Synthesis
- Digital Waveguide Mesh

Glottal Pulse Train or Noise

\( e(n) \) → \( 1 + k_1 \) → \( z^{-1/2} \) → Speech Output \( y(n) \)

\( k_1 \) → \( k_1 \) → \( R_1 \) → \( 1 - k_1 \) → \( z^{-1/2} \)

Kelly-Lochbaum Vocal Tract Model (Piecewise Cylindrical)

John L. Kelly and Carol Lochbaum (1962)
Sound Example

“Bicycle Built for Two”: (WAV) (MP3)

- Vocal part by Kelly and Lochbaum (1961)
- Musical accompaniment by Max Mathews
- Computed on an IBM 704
- Based on Russian speech-vowel data from Gunnar Fant’s book
- Probably the first digital physical-modeling synthesis sound example by any method
- Inspired Arthur C. Clarke to adapt it for “2001: A Space Odyssey” — the computer’s “first song”
Shiela” Sound Examples by Perry Cook (1990)

Overview

Digital Voice Models
- KL Voice
- “Daisy”
- “Shiela”
- Linear Prediction
- Glottal Model
- Source Estimation
- LF Glottal Model
- Phonation Variation
- Lu Sounds
- Tak
- 2D Vowels
- Pink Trombone

Karplus Strong

Digital Waveguides

Single Reeds

Bowed Strings

Distortion Guitar

Committed Synthesis

Digital Waveguide Mesh

- Diphones: (WAV) (MP3)
- Nasals: (WAV) (MP3)
- Scales: (WAV) (MP3)
- “Shiela”: (WAV) (MP3)
Linear Prediction (LP) Vocal Tract Model

- Can be interpreted as a modified Kelly-Lochbaum model
- In linear prediction, the glottal excitation must be an
  - *impulse*, or
  - *white noise*

*This prevents LP from finding a physical vocal-tract model*

- A *more realistic glottal waveform* $e(n)$ is needed before the vocal tract filter can have the “right shape”
- How to augment LPC in this direction without going to a full-blown *articulatory synthesis model*?

Jointly estimate *glottal waveform* $e(n)$ so that the vocal-tract filter converges to the “right shape”
Klatt Derivative Glottal Wave

Overview

Early Voice Models
- KL Voice
- "Daisy"
- "Shiela"
- Linear Prediction

Digital Voice Models
- Glottal Model
- Source Estimation
- LF Glottal Model
- Phonation Variation
- Lu Sounds
- Tak
- 2D Vowels
- Pink Trombone

Karplus Strong

Digital Waveguides
- Single Reeds
- Bowed Strings
- Distortion Guitar
- Commuted Synthesis
- Digital Waveguide Mesh

Digital Waveguide Mesh

Good for estimation:

- Truncated *parabola* each period
- Coefficients easily fit to *phase-aligned* inverse-filter output
Sequential Unconstrained Minimization

(Hui Ling Lu, 2002)

Klatt glottal (parabola) parameters are estimated *jointly* with vocal tract filter coefficients

- Formulation resembles that of the *equation error method for system identification* (also used in *invfreqz* in matlab)
- For *phase alignment*, we estimate
  - pitch (time varying)
  - glottal closure instant each period
- Optimization is *convex* in all but the phase-alignment dimension
  ⇒ one potentially nonlinear line search
Liljencrantz-Fant Derivative Glottal Wave Model

Overview

Early Voice Models

Digital Voice Models
- KL Voice
- “Daisy”
- “Sheila”
- Linear Prediction
- Glottal Model
- Source Estimation
- LF Glottal Model
- Phonation Variation
- Lu Sounds
- Tak
- 2D Vowels
- Pink Trombone

Digital Waveguides

Single Reeds

Bowed Strings

Distortion Guitar

Committed Synthesis

Digital Waveguide Mesh

- Better for intuitively parametrized expressive synthesis
- LF model parameters are fit to inverse filter output
- Use of Klatt model in forming filter estimate yields a “more physical” filter than LP
Parametrized Phonation Types

Overview

Early Voice Models
- KL Voice
- “Daisy”
- “Shiela”
- Linear Prediction
- Glottal Model
- Source Estimation
- LF Glottal Model
- Recognition

Digital Voice Models
- Lu Sounds
- Tak
- 2D Vowels
- Pink Trombone

Phonation Variation
- Tak
- 2D Vowels
- Pink Trombone

Karplus Strong

Digital Waveguides

Single Reeds

Bowed Strings

Distortion Guitar

Committed Synthesis

Digital Waveguide Mesh

Horn Filter Design

Julius Smith AES-2006 Masterclass – 22 / 101
Sound Examples by Hui Ling Lu

- Original: (WAV) (MP3)
- Synthesized:
  - Pressed Phonation: (WAV) (MP3)
  - Normal Phonation: (WAV) (MP3)
  - Breathy Phonation: (WAV) (MP3)

- Original: (WAV) (MP3)
- Synthesis 1: (WAV) (MP3)
- Synthesis 2: (WAV) (MP3)

where

- Synthesis 1 = Estimated Vocal Tract driven by estimated KLGLOT88 Derivative Glottal Wave (Pressed)
- Synthesis 2 = Estimated Vocal Tract driven by the fitted LF Derivative Glottal Wave (Pressed)

Google search: singing synthesis Lu
Voice Model Estimation

Overview

Early Voice Models
- KL Voice
- “Daisy”
- “Shiela”
- Linear Prediction
- Glottal Model
- Source Estimation
- LF Glottal Model
- Phonation Variation
- Lu Sounds

Digital Voice Models
- Tak
- 2D Vowels
- Pink Trombone

Digital Waveguides
- Single Reeds
- Bowed Strings
- Distortion Guitar
- Commuted Synthesis
- Digital Waveguide Mesh

(Pamornpol (Tak) Jinachitra 2006)

Noisy
A(z)

\[ v(n) \]
Noise/Error

\[ g(n) \]
Derivative glottal waveform

\[ \frac{1}{A(z)} \]
Vocal tract

\[ x(n) \]
Clean speech

\[ y(n) \]
Noisy speech

- Parametric source-filter model of voice + noise
- State-space framework with derivative glottal waveform as input and A model for dynamics
- Jointly estimate AR parameters and glottal source parameters using EM algorithm with Kalman smoothing
- Reconstruct a clean voice using Kelly-Lochbaum and estimated parameters
Online 2D Vowel Demo (2014)

Jan Schnupp, Eli Nelken, and Andrew King

http://auditoryneuroscience.com/topics/two-formant-artificial-vowels
Pink Trombone Voical Synthesis (March 2017)

Neil Thapen
Institute of Mathematics of the Academy of Sciences
Czech Republic

http://dood.al/pinktrombone/
Karplus-Strong (KS) Algorithm
Karplus-Strong (KS) Algorithm (1983)

Overview

Early Voice Models
Digital Voice Models
Karplus Strong
- Karplus Strong
Digital Waveguides
Single Reeds
Bowed Strings
Distortion Guitar
Committed Synthesis
Digital Waveguide Mesh
Horn Filter Design
Finite Differences
Virtual Analog
Wave Digital Filters

- Discovered (1978) as “self-modifying wavetable synthesis”
- Wavetable is preferably initialized with random numbers
- No physical modeling at this point
Digital Waveguide Modeling
Digital Waveguide Models (1985)

Lossless digital waveguide $\triangleleft$ bidirectional delay line at some wave impedance $R$

Useful for efficient models of

- strings
- bores
- plane waves
- conical waves
Signal scattering is caused by a change in wave impedance $R$:

$$k_1 = \frac{R_2 - R_1}{R_2 + R_1}$$

If the wave impedance changes every spatial sample, the Kelly-Lochbaum vocal-tract model results (also need reflecting terminations)
Moving Termination: Ideal String

- Left endpoint moved at velocity $v_0$
- External force $f_0 = Rv_0$
- $R = \sqrt{K/\epsilon}$ is the wave impedance (for transverse waves)
- Relevant to bowed strings (when bow pulls string)
- 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 $c = \sqrt{K/\epsilon}$
Digital Waveguide “Equivalent Circuits”

Overview

Early Voice Models

Digital Voice Models

Karplus Strong

Digital Waveguides
- Digital Waveguide
- Signal Scattering
- Moving Termination
- Waveguide Model
- Plucked String
- Struck String
- Damped String
- Commuted Gains
- Damping Filters
- Simplest Damping
- Karplus-Strong Aha
- EKS Algorithm
- Physical Excitation
- Pick Position FFCF
- PLPC Cello

Single Reeds

Bowed Strings

Distortion Guitar

Commuter Synthesis

Julius Smith

(a) Velocity waves.  (b) Force waves.

(Animation)

(Interactive Animation)
Ideal Plucked String (Displacement Waves)

Overview

Early Voice Models

Digital Voice Models

Karplus Strong

Digital Waveguides
- Digital Waveguide
- Signal Scattering
- Moving Termination
- Waveguide Model
- Plucked String
- Struck String
- Damped String
- Commuted Gains
- Damping Filters
- Simplest Damping
- Karplus-Strong Aha
- EKS Algorithm
- Physical Excitation
- Pick Position FFCF
- PLPC Cello

Single Reeds

Bowed Strings

Distortion Guitar

- Load each delay line with \textit{half} of initial string displacement
- Sum of upper and lower delay lines = string displacement
Ideal Struck String (Velocity Waves)

Overview

Early Voice Models
Digital Voice Models
Karplus Strong

Digital Waveguides
- Digital Waveguide
- Signal Scattering
- Moving Termination
- Waveguide Model
- Plucked String
- Struck String
- Damped String
- Commuted Gains
- Damping Filters
- Simplest Damping
- Karplus-Strong Aha
- EKS Algorithm
- Physical Excitation
- Pick Position FFCF
- PLPC Cello

Single Reeds
Bowed Strings
Distortion Guitar

Digital Waveguide Mesh

Hammer strike = momentum transfer = velocity step:

\[ m_h v_h(0-) = (m_h + m_s) v_s(0+) \]
Digital Waveguide Interpretation of Karplus-Strong

Begin with an ideal *damped* string model:

\[
\begin{align*}
  y^+(n) & \leftarrow N/2 \text{ samples delay, } N/2 \text{ loss factors } g \\
  y^+(n-N/2) & \leftarrow -1 \quad \text{“Bridge” Rigid Termination} \\
  y^-(n) & \rightarrow N/2 \text{ samples delay, } N/2 \text{ loss factors } g \\
  y^-(n+N/2) & \rightarrow -1 \quad \text{“Nut” Rigid Termination} \\
  y^*(n,2cT) & \leftarrow y(nT,0) \\
  y^+(n) & \leftarrow g \\
  y^-(n) & \leftarrow g \\
  y^+(n-N/2) & \leftarrow g \\
  y^-(n+N/2) & \leftarrow g \\
\end{align*}
\]
Equivalent System: Gain Elements Commuted

Output $y^+(n)$

$N$ samples delay

$g^N$

$y^+(n-N)$

All $N$ loss factors $g$ have been “pushed” through delay elements and combined at a single point.

Computational Savings

- $f_s = 50\text{kHz}, f_1 = 100\, \text{Hz} \Rightarrow \text{delay} = 500$
- 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)
- Such filters *also commute* with delay elements (LTI)
- Typically only *one* gain filter used per loop

Output $y^+(n)$

\[
y^+(n) = \sum_{k=0}^{N-1} g^k y(n-k) + y(n-N)
\]

$N$ samples delay

\[ g^N \]
Simplest Frequency-Dependent Loop Filter

\[ \hat{G}(z) = b_0 + b_1 z^{-1} \]

- **Uniform delay** \( \Rightarrow b_0 = b_1 \) (\( \Rightarrow \) delay = 1/2 sample)

- **Zero damping at dc** \( \Rightarrow b_0 + b_1 = 1 \)
  \( \Rightarrow b_0 = b_1 = 1/2 \)
  \[ \hat{G}(e^{j\omega T}) = \cos \left( \frac{\omega T}{2} \right), \quad |\omega| \leq \pi f_s \]

- **This is precisely the Karplus-Strong loop filter!**
Karplus-Strong Algorithm Revisited

Overview

Early Voice Models
Digital Voice Models
Karplus Strong
Digital Waveguides
- Digital Waveguide
- Signal Scattering
- Moving Termination
- Waveguide Model
- Plucked String
- Struck String
- Damped String
- Commuted Gains
- Damping Filters
- Simplest Damping
- Karplus-Strong Aha
- EKS Algorithm
- Physical Excitation
- Pick Position FFCF
- PLPC Cello

Single Reeds
Bowed Strings
Distortion Guitar

Physical Interpretation

- Delay line is initialized with noise (random numbers)
- Therefore, assuming a displacement-wave simulation:
  - Initial string displacement = sum of delay-line halves
  - Initial string velocity ← difference of delay-line halves
- The Karplus-Strong “string” is thus plucked and struck by random amounts along the entire length of the string!
  (“splucked string”?)
EKS Algorithm (Jaffe-Smith 1983)

\[ H_p(z) = \frac{1 - p}{1 - p z^{-1}} = \text{pick-direction lowpass filter} \]

\[ H_\beta(z) = 1 - z^{-\beta N} = \text{pick-position comb filter, } \beta \in (0, 1) \]

\[ H_d(z) = \text{string-damping filter (one/two poles/zeros typical)} \]

\[ H_s(z) = \text{string-stiffness allpass filter (several poles and zeros)} \]

\[ H_\rho(z) = \frac{\rho(N) - z^{-1}}{1 - \rho(N) z^{-1}} = \text{first-order string-tuning allpass filter} \]

\[ H_L(z) = \frac{1 - R_L}{1 - R_L z^{-1}} = \text{dynamic-level lowpass filter} \]
Karplus-Strong Sound Examples

- “Vintage” 8-bit sound examples:
  - Original Plucked String: (WAV) (MP3)
  - Drum: (WAV) (MP3)
  - Stretched Drum: (WAV) (MP3)
EKS Sound Examples

Overview

Early Voice Models

Digital Voice Models

Karplus Strong

Digital Waveguides

- Digital Waveguide
- Signal Scattering
- Moving Termination
- Waveguide Model
- Plucked String
- Struck String
- Damped String
- Commuted Gains
- Damping Filters
- Simplest Damping
- Karplus-Strong Aha
- EKS Algorithm
- Physical Excitation
- Pick Position FFCF
- PLPC Cello

Single Reeds

Bowed Strings

Distortion Guitar

Plucked String: (WAV) (MP3)

- Plucked String 1: (WAV) (MP3)
- Plucked String 2: (WAV) (MP3)
- Plucked String 3: (WAV) (MP3)

(Computed using Plucked.cpp in the C++ Synthesis Tool Kit (STK) by Perry Cook and Gary Scavone)
EKS Sound Example (1988)

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

- Executed in real time on one Motorola DSP56001 (20 MHz clock, 128K SRAM)
- Developed for the NeXT Computer introduction at Davies Symphony Hall, San Francisco, 1988
- Solo violin part was played live by Dan Kobialka of the San Francisco Symphony
Example EKS Extension

Overview

Early Voice Models

Digital Voice Models

Karplus Strong

Digital Waveguides

• Digital Waveguide
• Signal Scattering
• Moving Termination
• Waveguide Model
• Plucked String
• Struck String
• Damped String
• Commuted Gains
• Damping Filters
• Simplest Damping
• Karplus-Strong Aha
  • EKS Algorithm
  • Physical Excitation
  • Pick Position FFCF
  • PLPC Cello

Single Reeds

Bowed Strings

Distortion Guitar

Digitally Waveguide Mesh

Julius Smith

Several of the Karplus-Strong algorithm extensions were based on its *physical interpretation*.

- Originally, transfer-function methods were used (1982)
- Later, *digital waveguide* methods were used (1986)
String Excited Externally at One Point

“Waveguide Canonical Form (1986)”

Equivalent System by Delay Consolidation:

Finally, we “pull out” the comb-filter component:
EKS “Pick Position” Extension

Equivalent System: Comb Filter Factored Out

\[ H(z) = z^{-N} \frac{1 + z^{-2M}}{1 - z^{-(2M+2N)}} = (1 + z^{-2M}) \frac{z^{-N}}{1 - z^{-(2M+2N)}} \]

- **Excitation Position** controlled by left delay-line length
- **Fundamental Frequency** controlled by right delay-line length
- “Transfer function modeling” based on a physical model (1982)
PLPC Cello (1982)

- Bandlimited Impulse Train
- Comb Filter
- String Loop
- Body Filter

(bow position) (EKS) (40-pole LPC)

- Periodic LPC used to estimate string-loop filter
- Normal LPC used for body model (40 poles)
- Excitation = Bandlimited impulse train (Moorer 1975):

\[
\sum_{k=1}^{K} \cos(k\omega_0 t) = \frac{\sin[(K + 1/2)\omega_0 t]}{2 \sin(\omega_0 t/2)} - \frac{1}{2}
\]

- Bow-position simulation = variable-delay differencing comb filter (direct from physical interpretation)
- **Sound Example:**
  Moving Bow-Stroke Example: (WAV) (MP3)
  (Bowing point moves toward the “bridge”)
Single-Reed Instruments
Schematic Physical Model

- Main control variable = air pressure applied to reed
- Secondary control variable = reed embouchure
- Pressure waves = natural choice for simulation
- Radiation ≈ “Omni” at LF, more directional at HF

Bell ≈ power-complementary “cross-over” filter:

- Low frequencies reflect (inverted)
- High frequencies transmit
- Cross-over frequency ≈ 1500 Hz for clarinet
  (where wavelength ≈ bore diameter)
Single-Reed Digital Waveguide Model (1986)

- Bore = bidirectional delay line (losses lumped)
- Bore length = 1/4 wavelength in lowest register
  - Bell reflection ≈ -1 at low frequencies
  - Mouthpiece reflection ≈ +1
- Reflection filter depends on first few open toneholes
- In a simple implementation, the bore is “cut to a new length” for each pitch
Simplified Single-Reed Theory

\[ p_m \triangleq \text{mouth pressure (constant)} \]
\[ p_b \triangleq \text{bore pressure (dynamic)} \]
\[ p_{m\Delta} \triangleq p_m - p_b \triangleq \text{pressure drop across mouthpiece} \]
\[ u_m \triangleq \text{resulting flow into mouthpiece} \]
\[ R_m(p_{m\Delta}) \triangleq \text{reed-aperture impedance (measured)} \]
Toward a Computational Reed Model

Given:

\[ p_m = \text{Mouth pressure} \]
\[ p_b^+ = \text{Incoming traveling bore pressure} \]

Find:

\[ p_b^- = \text{Outgoing traveling bore pressure} \]

such that:

\[ 0 = u_m + u_b = \frac{p\Delta}{R_m(p\Delta)} + \frac{p_b^+ - p_b^-}{R_b}, \]

\[ p\Delta \triangleq p_m - p_b = p_m - (p_b^+ + p_b^-) \]

Solving for \( p_b^- \) is not immediate because \( R_m \) depends on \( p\Delta \) which depends on \( p_b^- \).
Graphically solve:

\[ G(p_\Delta) = p_\Delta^+ - p_\Delta, \quad p_\Delta^+ \triangleq p_m - 2p_b \]

where

\[ G(p_\Delta) \triangleq R_b u_m(p_\Delta) = R_b p_\Delta / R_m(p_\Delta) \]

- Analogous to finding the “operating point” of a transistor by intersecting its “operating curve” with the “load line” determined by the load resistance.

- Outgoing wave is then \( p_b^- = p_m - p_b^+ - p_\Delta (p_\Delta^+) \)
Graphical Solution Technique Illustrated

Iteratively solve:

\[ p_\Delta^+ - p_\Delta = R_b p_\Delta / R_m(p_\Delta), \quad \text{where } p_\Delta^+ \triangleq p_m - 2p_b^+ \]

Solution can be pre-computed and stored in a look-up table \( \rho(p_\Delta^+) \)
Digital Waveguide Single Reed, Cylindrical Bore Model (1986)

Overview
- Early Voice Models
- Digital Voice Models
- Karplus Strong
- Digital Waveguides

Single Reeds
- Schematic Model
- Waveguide Model
- Simplified Reed
- Reed Computation
- Graphical Solver
- Example
- Clarinet
- Wind Examples

Bowed Strings
- Distortion Guitar
- Commuted Synthesis
- Digital Waveguide Mesh
- Horn Filter Design
- Finite Differences
- Virtual Analog

Julius Smith AES-2006 Masterclass – 56 / 101

---

Control variable = mouth half-pressure
Total reed cost = two subtractions, one multiply, and one table lookup per sample
Digital Waveguide Wind Instrument Sound Examples

- STK Clarinet: (WAV) (MP3)
  Google search: *STK clarinet*
- See also Faust-STK Clarinet (new)
- Staccato Systems Slide Flute
  (based on STK flute, ca. 1995): (WAV) (MP3)
- Yamaha VL1 “Virtual Lead” synthesizer demos (1994):
  - Shakuhachi: (WAV) (MP3)
  - Oboe and Bassoon: (WAV) (MP3)
  - Tenor Saxophone: (WAV) (MP3)
Bowed Strings
A schematic model for bowed-string instruments.

- Bow divides string into two sections
- Primary control variable = bow velocity
  \( \Rightarrow \) velocity waves = natural choice of wave variable
- Bow junction = nonlinear two-port
- Must find velocity input to string (injected equally to left and right) such that friction force = string reaction force.
Bow-String Contact Physics

Overview
- Early Voice Models
- Digital Voice Models
- Karplus Strong
- Digital Waveguides
- Single Reeds
- Bowed Strings
  - Schematic Model
  - **Bow-String Contact**
  - Bow-String Junction
  - Bow-String Reflection
  - Simplified Bow Table
  - Bowed Strings
- Distortion Guitar
- Commuted Synthesis
- Digital Waveguide Mesh
- Horn Filter Design
- Finite Differences
- Virtual Analog
- Wave Digital Filters

Applied Force $= \text{Friction Curve} \times \text{Differential Velocity}$

\[\parallel\]

Reaction Force $= \text{String Wave Impedance} \times \text{Velocity Change}$
Friedlander-Keller

Overview
- Early Voice Models
- Digital Voice Models
- Karplus Strong
- Digital Waveguides
- Single Reeds
- Bowed Strings
  - Schematic Model
  - Bow-String Contact
  - Bow-String Junction
  - Bow-String Reflection
  - Simplified Bow Table
  - Bowed Strings
- Distortion Guitar
- Commuted Synthesis
- Digital Waveguide Mesh
- Horn Filter Design
- Finite Differences
- Virtual Analog
- Wave Digital Filters

---

Overlay of normalized bow-string friction curve $R_b(v_\Delta)/R_s$ with the string “load line” $v_\Delta^* - v_\Delta$. 

Break-Away / Capture

Bow and String Slipping (Reduced Friction)

Normalized Friction $R_b/R_s$ times differential velocity $v_\Delta$

String "Load Line"

Solution = Graphical Intersection

"Incoming" differential velocity

Bow and String Slipping (Reduced Friction)

Bow and String Stuck Together

Julius Smith
Bow-String Scattering Junction

Friedlander-Keller diagram is solved when

\[ R_b(v_\Delta) \cdot v_\Delta = R_s \cdot [v_\Delta^+ - v_\Delta] \]

which implies

\[ v_{s,r}^- = v_{s,l}^- + \hat{\rho}(v_\Delta^+) \cdot v_\Delta^- \]
\[ v_{s,l}^- = v_{s,r}^- + \hat{\rho}(v_\Delta^+) \cdot v_\Delta^+ \]

where

\[ v_{s,r} = \text{transverse string velocity on the right of the bow} \]
\[ v_{s,l} = \text{string velocity left of the bow } (v_{s,l} = v_{s,r}) \]
\[ v_\Delta^+ \triangleq v_b - (v_{s,r}^+ + v_{s,l}^+) = \text{“incoming differential velocity”} \]
\[ v_b = \text{bow velocity, and } \hat{\rho}(v_\Delta^+) \text{ is given by . . .} \]
Bow-String Reflection Coefficient

\[
\hat{\rho}(v_+^\Delta) = \frac{r(v_\Delta(v_+^\Delta))}{1 + r(v_\Delta(v_+^\Delta))}
\]

where

\[
r(v_\Delta) = 0.25R_b(v_\Delta)/R_s
\]

\[
v_\Delta = v_b - v_s \quad \text{bow velocity minus string velocity}
\]

\[
v_s = v_{s,l} + v_{s,l} = v_{s,r} + v_{s,r} = \text{transverse string velocity}
\]

\[
R_s = \text{wave impedance of string}
\]

\[
R_b(v_\Delta) = \text{friction coefficient for the bow against the string, i.e.,}
\]

\[
F_b(v_\Delta) = R_b(v_\Delta) \cdot v_\Delta
\]
Simplified, Piecewise Linear Bow Table

- **Overview**
- **Early Voice Models**
- **Digital Voice Models**
- **Karplus Strong**
- **Digital Waveguides**
- **Single Reeds**
- **Bowed Strings**
  - Schematic Model
  - Bow-String Contact
  - Bow-String Junction
  - Bow-String Reflection
  - Simplified Bow Table
  - Bowed Strings
- **Distortion Guitar**
- **Commuted Synthesis**
- **Digital Waveguide Mesh**
- **Horn Filter Design**
- **Finite Differences**
- **Virtual Analog**
- **Wave Digital Filters**

### Simplified Bow Table

- **Break-Away / Capture**
  - $\hat{\beta}(v^+_\Delta)$
  - Total Reflection

- **Bow and String Stuck Together**

- **Break-Away / Capture**

\[
\begin{array}{c|c|c|c}
\text{Total Transmission} & 0 & -v^c\Delta & 0 & v^c\Delta & v^+_\Delta & 1 \\
\hline
-1 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & 1 \\
\end{array}
\]

- **Flat center portion** corresponds to a fixed reflection coefficient “seen” by a traveling wave encountering the bow stuck against the string.

- **Outer sections** give a smaller reflection coefficient corresponding to the reduced bow-string interaction force while the string is slipping under the bow.

- **Hysteresis is neglected**
Digital Waveguide Bowed Strings (1986)

- Reflection filter summarizes all losses per period (due to bridge, bow, finger, etc.)
- Bow-string junction = *memoryless* lookup table (or segmented polynomial)
Bowed and Plucked Sound Examples by Esteban Maestre

- Synthetically Bowing STK’s Bowed.cpp
- Finite-Width Thermal Friction Model for Bowed String
- Waveguide Strings Coupled to Modal-Synthesis Bridge
Electric Guitar with Overdrive and Feedback
Distortion output signal often further filtered by an *amplifier cabinet filter*, representing speaker cabinet, driver responses, etc.
Distortion Guitar Sound Examples

(Stanford Sondius Project, ca. 1995)

- Distortion Guitar: (WAV) (MP3)
- Amplifier Feedback 1: (WAV) (MP3)
- Amplifier Feedback 2: (WAV) (MP3)
Commuted Waveguide Synthesis

Overview
Early Voice Models
Digital Voice Models
Karplus Strong
Digital Waveguides
Single Reeds
Bowed Strings
Distortion Guitar
Commuted Synthesis
Digital Waveguide Mesh
Horn Filter Design
Finite Differences
Virtual Analog
Wave Digital Filters
**Commuted Synthesis of Acoustic Strings (1993)**

**Overview**
- Early Voice Models
- Digital Voice Models
- Karplus Strong
- Digital Waveguides
- Single Reeds
- Bowed Strings
- Distortion Guitar
- Commuted Synthesis
  - Acoustic Strings
  - Sound Examples
  - Linearized Violin
  - Commuted Piano
  - Pulse Synthesis
  - Complete Piano
  - Sound Examples
- Digital Waveguide Mesh
- Horn Filter Design
- Finite Differences
- Virtual Analog
- Wave Digital Filters

---

**Schematic diagram of a stringed musical instrument.**

- Trigger → Excitation → String → Resonator → Output

**Equivalent diagram in the linear, time-invariant case.**

- Trigger → Excitation → Resonator → String → Output

**Use of an aggregate excitation given by the convolution of original excitation with the resonator impulse response.**

- Trigger → Aggregate Excitation → String → Output
Commuted Components

Overview
- Early Voice Models
- Digital Voice Models
- Karplus Strong
- Digital Waveguides
- Single Reeds
- Bowed Strings
- Distortion Guitar

Commuted Synthesis
- Acoustic Strings
- Sound Examples
- Linearized Violin
- Commuted Piano
- Pulse Synthesis
- Complete Piano
- Sound Examples

Digital Waveguide Mesh
- Horn Filter Design
- Finite Differences
- Virtual Analog
- Wave Digital Filters

Julius Smith

“Plucked Resonator” driving a String.

Possible components of a guitar resonator.
Sound Examples

**Electric Guitar (Pick-Ups and/or Body-Model Added)** *(Stanford Sondius Project → Staccato Systems, Inc. → ADI, ca. 1995)*

- Example 1: (WAV) (MP3)
- Example 2: (WAV) (MP3)
- Example 3: (WAV) (MP3)
- Virtual “wah-wah pedal”: (WAV) (MP3)

**STK Mandolin**

- STK Mandolin 1: (WAV) (MP3)
- STK Mandolin 2: (WAV) (MP3)
Sound Examples

More Recent Acoustic Guitar

- Bach Prelude in E Major: (WAV) (MP3)
- Bach Loure in E Major: (WAV) (MP3)
- More examples
- Yet more examples

Virtual performance by Dr. Mikael Laurson, Sibelius Institute

Virtual guitar by Helsinki Univ. of Tech., Acoustics Lab^1

^1http://www.acoustics.hut.fi/
Commuted Synthesis of Linearized Violin

- Assumptions:
  - Assumes *ideal Helmholtz motion* of string
  - Sound Examples (Stanford Sondius project, ca. 1995):
    - Bass: (WAV) (MP3)
    - Cello: (WAV) (MP3)
    - Viola 1: (WAV) (MP3)
    - Viola 2: (WAV) (MP3)
    - Violin 1: (WAV) (MP3)
    - Violin 2: (WAV) (MP3)
    - Duet: (WAV) (MP3)
Hammer-string interaction pulses (force):

![Graph showing hammer-string interaction pulses](image)
Faster collisions correspond to narrower pulses (*nonlinear filter*)

- For a *given velocity*, filter is linear time-invariant
- Piano is “linearized” for each hammer velocity
Multiple Hammer-String Interaction Pulses

Superimpose several individual pulses:

- Impulse 1
- Impulse 2
- Impulse 3

As impulse amplitude grows (faster hammer strike), output pulses become *taller and thinner*, showing less overlap.
Complete Piano Model

Natural Ordering:

\[ v_c \]

\[ \delta_1 \]

\[ \delta_2 \]

\[ \delta_3 \]

Commuted Ordering:

\[ v_c \]

- Soundboard and enclosure are *commuted*
- Only need a stored recording of their *impulse response*
- An enormous digital filter is otherwise required
Piano and Harpsichord Sound Examples

(Stanford Sondius Project, ca. 1995)

- Piano: (WAV) (MP3)
- Harpsichord 1: (WAV) (MP3)
- Harpsichord 2: (WAV) (MP3)
More Recent Harpsichord Example

- Harpsichord Soundboard Hammer-Response: (WAV) (MP3)
- Musical Commuted Harpsichord Example: (WAV) (MP3)
- More examples

References:
- “Sound Synthesis of the Harpsichord Using a Computationally Efficient Physical Model”,
- by Vesa Välimäki, Henri Penttinen, Jonte Knif, Mikael Laurson, and Cumhur Erkut, JASP-2004
- Recent dissertation by Jack Perng (Stanford, Physics/CCRMA)
Digital Waveguide Mesh
At each junction:

\[ V_J = \frac{\text{in}_1 + \text{in}_2 + \text{in}_3 + \text{in}_4}{2} \]

\[ \text{out}_k = V_J - \text{in}_k, \quad k = 1, 2, 3, 4 \]

Sound example:

- Gongs (lossless nonlinear rim by Pierce and Van Duyne JASA-1997)
Recent Mesh Topic: Virtual Quadratic Residue Diffusers

Manfred Schroeder’s Quadratic Residue Diffuser for $N = 17$:

- Thin vertical lines = rigid separators between wells.
- Well depths are $N - s_n$, where

$$s_n = n^2 \pmod{N}, \quad n \in \mathbb{Z}$$

is a quadratic residue sequence
Example QRD

For \( N = 17 \), we have

\[
s = [0, 1, 4, 9, 16, 8, 2, 15, 13, 13, 15, 2, 8, 16, 9, 4, 1; 0, 1, \ldots]
\]

Reflection magnitude *equal* in \( N \) equally spaced directions at the "design wavelength" (and "not bad" in between)
QRD Termination of a 2D Digital Waveguide Mesh

Overview
Early Voice Models
Digital Voice Models
Karplus Strong
Digital Waveguides
Single Reeds
Bowed Strings
Distortion Guitar
Committed Synthesis
Digital Waveguide Mesh
- 2D Waveguide Mesh
- QR Diffuser
- Example QRD
- QRD Mesh Boundary
- Scattering Levels
Horn Filter Design
Finite Differences
Virtual Analog
Wave Digital Filters

(Lee-Smith ICMC-2004)
Scattering levels from a Schroeder diffuser (solid) and a straight boundary (dashed)

Animations: https://ccrma.stanford.edu/~kglee/2dmesh_QRD/
Horn Filter Design
Bore Profile Reconstruction from Measured Trumpet Reflectance

- Inverse scattering applied to pulse-reflectometry data to fit piecewise-cylindrical model (like LPC model)
- Bore profile reconstruction is reasonable up to bell
- The bell is not physically equivalent to a piecewise-cylindrical acoustic tube, due to
  - complex radiation impedance,
  - conversion to higher order transverse modes
Trumpet-Bell Impulse Response Computed from Estimated Piecewise-Cylindrical Model

**Overview**

- Early Voice Models
- Digital Voice Models
- Karplus Strong
- Digital Waveguides
- Single Reeds
- Bowed Strings
- Distortion Guitar
- Commuted Synthesis
- Digital Waveguide Mesh

**Horn Filter Design**
- Bore Profile
- **Impulse Response**
- Bell Filter Design
- Bell Reflectance
- Ideal (1998)
- Four Exponentials
- Exponentials & Cubic
- Exponentials & SM
- Example
- One-Pole TIIR

**Finite Differences**

**Julius Smith AES-2006 Masterclass – 90 / 101**

- From pulse reflectometry on trumpet with no mouthpiece
- Bore profile is reconstructed, smoothed, and segmented
- Impulse response of “bell segment” = “ideal filter”
Trumpet-Bell Filter Design Problem

- At $f_s = 44.1$ kHz, impulse-response length $\approx 400 - 600$ samples
- A length 400 FIR bell filter is expensive!
- Convert to IIR? Hard because
  - Phase (resonance tunings) must be preserved
  - Magnitude (resonance Q) must be preserved
  - Rise time $\approx 150$ samples
  - Phase-sensitive IIR design methods perform poorly
Measured Trombone Bell Reflectance
Idea! (1998)

- Break up impulse response into exponential or polynomial segments
- Exponential and polynomial impulse-responses can be designed using *Truncated IIR (TIIR) Filters*
- Google search: *TIIR Horns*
Four-Exponential Fit to Estimated Trumpet-Bell Filter (Exp-4)

Overview
Early Voice Models
Digital Voice Models
Karplus Strong
Digital Waveguides
Single Reeds
Bowed Strings
Distortion Guitar
Commuted Synthesis
Digital Waveguide Mesh

Horn Filter Design
- Bore Profile
- Impulse Response
- Bell Filter Design
- Bell Reflectance
- Ideal (1998)
- **Four Exponentials**
- Exponentials & Cubic
- Exponentials & SM
- Example
- One-Pole TIIR

Exp-4 Impulse Response Fit
Two Exponentials Connected by a Cubic Spline Measured Trumpet Data (Exp2-S3)

Overview

Early Voice Models
Digital Voice Models
Karplus Strong
Digital Waveguides
Single Reeds
Bowed Strings
Distortion Guitar
Committed Synthesis

Digital Waveguide Mesh

Horn Filter Design
- Bore Profile
- Impulse Response
- Bell Filter Design
- Bell Reflectance
- Ideal (1998)
- Four Exponentials
- Exponentials & Cubic
- Exponentials & SM
- Example
- One-Pole TIIR

Finite Differences

Julius Smith AES-2006 Masterclass – 95 / 101
Two Exponentials Followed by a 6th-Order IIR Filter Designed by Steiglitz McBride Algorithm (Exp2-SM6)
Exp2-SM6 Amplitude Response Fit

Overview

Early Voice Models
Digital Voice Models
Karplus Strong
Digital Waveguides
Single Reeds
Bowed Strings
Distortion Guitar
Committed Synthesis

Digital Waveguide Mesh

Horn Filter Design
- Bore Profile
- Impulse Response
- Bell Filter Design
- Bell Reflectance
- Ideal (1998)
- Four Exponentials
- Exponentials & Cubic
- Exponentials & SM
- Example
- One-Pole TIIR

Finite Differences

Julius Smith
Virtual Analog

Magnitude Fit over Entire Nyquist Band

- Ideal Approximation

Normalized Frequency

Magnitude (dB)
Exp2-SM6 Low-Frequency Zoom

Overview

Early Voice Models
Digital Voice Models
Karplus Strong
Digital Waveguides
Single Reeds
Bowed Strings
Distortion Guitar
Commuted Synthesis
Digital Waveguide Mesh

Horn Filter Design
- Bore Profile
- Impulse Response
- Bell Filter Design
- Bell Reflectance
- Ideal (1998)
- Four Exponentials
- Exponentials & Cubic
- Exponentials & SM
- Example
- One-Pole TIIR

Finite Differences

Julius Smith
Virtual Analog

AES-2006 Masterclass – 98 / 101
Exp2-SM6 Group Delay Fit

Overview
Early Voice Models
Digital Voice Models
Karplus Strong
Digital Waveguides
Single Reeds
Bowed Strings
Distortion Guitar
Committed Synthesis

Digital Waveguide Mesh
Horn Filter Design
- Bore Profile
- Impulse Response
- Bell Filter Design
- Bell Reflectance
- Ideal (1998)
- Four Exponentials
- Exponentials & Cubic
- Exponentials & SM
- Example
- One-Pole TIIR

Finite Differences
Julius Smith
Virtual Analog

AES-2006 Masterclass – 99 / 101
Exp2-SM6 Phase Delay Fit

Overview

Early Voice Models
Digital Voice Models
Karplus Strong
Digital Waveguides
Single Reeds
Bowed Strings
Distortion Guitar
Committed Synthesis

Digital Waveguide Mesh

Horn Filter Design
- Bore Profile
- Impulse Response
- Bell Filter Design
- Bell Reflectance
- Ideal (1998)
- Four Exponentials
- Exponentials & Cubic
- Exponentials & SM
- Example
- One-Pole TIIR

Finite Differences
A One-Pole (Almost) TIIR Filter

- Generates truncated exponentials or constants
- Filter complexity on average \( \approx \) one pole
- Higher orders give truncated polynomial impulse responses
Finite Differences
Lumped Modeling Elements

- The foregoing models were *distributed-parameter systems*
- Distributed systems supporting *wave propagation*
- We also need *lumped-parameter systems*, such as for
  - Piano hammer
  - Brass-player’s lips
  - Vocal-fold models
  - Reeds with mass
  - Helmholtz resonators
  - Circuit models
- Instead of *sampled traveling waves*, we employ *finite difference schemes* to model lumped systems
Finite Difference Approximation (FDA)

Consider the simple differential equation relating velocity and force for an ideal mass:

\[
\frac{dv}{dt} = f(t) \leftrightarrow F(s) = m s V(s)
\]

- Assume \( f(t) \) = “input” signal, and \( v(t) \) = “output”.
- We need to discretize this continuous time equation.
- Input signal = \( f_n = f(nT) \), \( n = 0, 1, 2, \ldots \), where \( T \) is the sampling interval.
- The FDS will compute \( v_n \) given \( f_n \) for \( n = 0, 1, 2, \ldots \).
The *Finite Difference Approximation* replaces *differentiation* by a *finite difference*, e.g.,

\[
\frac{dv}{dt} \approx \frac{v_n - v_{n-1}}{T} \quad \text{ (“backward difference”)}
\]

\[
\approx \frac{v_{n+1} - v_{n-1}}{2T} \quad \text{ (“centered difference”)}
\]

Using the backwards-difference approximation, we obtain

\[
v_n = v_{n-1} + \frac{T}{m} f_n, \quad n = 0, 1, 2, \ldots
\]

We see that \( v_{-1} \) must be specified as an initial condition.
Starting with the driving point impedance

\[ R(s) \triangleq \frac{F(s)}{V(s)} = ms \]

the bilinear transform gives the digital impedance

\[ \frac{F_d(z)}{V_d(z)} \triangleq R_d(z) = R \left( \frac{1 - z^{-1}}{c \left( \frac{1}{1 + z^{-1}} \right)} \right) = mc \frac{1 - z^{-1}}{1 + z^{-1}} \]
Bilinear Transform of the Ideal Mass, Cont’d

Multiplying out

\[ F_d(z) + z^{-1} F_d(z) = mcV_d(z) - mc z^{-1}V_d(z) \]

and taking the inverse \( z \) transform gives

\[ f_n + f_{n-1} = mc (v_n - v_{n-1}) \]

or

\[ v_n = v_{n-1} + \frac{1}{mc} (f_n + f_{n-1}) \]

(The \( f_{n-1} \) term is new relative to the FDA.)

Equivalent to the trapezoid rule for numerical integration.
Accuracy

Recall that the backward-difference approximation is first-order accurate in $T$:

$$\omega_a = \omega_d + O(\omega_d^2 T)$$

For the trapezoid rule we get

$$\omega_a = \omega_d + O(\omega_d^3 T^2)$$

- Trapezoid rule (bilinear transform) is second-order accurate in $T$.
- Higher order accuracy obtainable using more neighboring grid points.
- How should these extra grid points be brought in?
Observations

- The FDA gave us

\[ v_n = v_{n-1} + \frac{T}{m} f_n, \quad n = 0, 1, 2, \ldots \]

which is a one-pole digital filter having a pole at \( z = 1 \) and a zero at \( z = 0 \).

- Similarly, the BLT gave us

\[ v_n = v_{n-1} + \frac{T}{2m} (f_n + f_{n-1}) \]

which has the same pole, but a zero at \( z = -1 \).

- **Question:** *How do we use more poles and zeros to obtain a more accurate FDS?*
The driving-point impedance $R(s) = ms$ of an ideal mass is an *ideal differentiator* (scaled by $m$):

$$R(j\omega) = mj\omega.$$ 

It is therefore natural to define the ideal *digital* differentiator as

$$H(e^{j\omega T}) = j\omega, \quad \omega T \in [-\pi, \pi)$$
An exact match is not possible with a finite order digital filter (note frequency-response discontinuity at $z = -1$).

In practice, we minimize $\| H(e^{j\omega T}) - \hat{H}(e^{j\omega T}) \|$ where $\hat{H}$ is a digital filter frequency response.

We need some oversampling in order to have a guard band (e.g., from 20 kHz to 22 kHz).

Desired response is unconstrained in the guard band.
Virtual Analog Synthesis
Most “Virtual Analog” synthesizers try to emulate some version of the MiniMoog or MemoryMoog synthesizers, because of their popularity. These classic synths were designed by the analog-synth pioneer Robert Moog.
Moog VCF Ladder (1966)
Moog VCF Structure

**Structure:** four identical series one-poles in a feedback loop:

\[ x(t) \rightarrow \sum \rightarrow G_1(s) \rightarrow G_1(s) \rightarrow G_1(s) \rightarrow G_1(s) \rightarrow y(t) \]

This implements a voltage-controllable four-pole filter:

**VCF Amplitude Response**
Moog VCF Controls

\[ G_1(s) = \frac{1}{1 + \frac{s}{\omega_c}} \]

Controls

- Pole location \( s = -\omega_c \): controls cut-off frequency
- Feedback gain \( k \): controls resonance
Digitizing the Moog VCF

(Stilson-Smith ICMC-96)

Need to digitize $G_1(s) \rightarrow \hat{G}_1(z)$

- **Backward Difference:** $s = \frac{1 - z^{-1}}{T}$
- **Bilinear Transform:** $s = \frac{2(1 - z^{-1})}{T(1 + z^{-1})}$

**Problem:** Delay-free loops

**Solution:** “Separation table” $k_Q(\omega_c)$

**Reference:** Tim Stilson’s PhD/EE thesis (June 2006, Stanford/CCRMA)
Bilinear transform with separation table:
Backwards difference, no separation table:
Discovery: BLT/BackDiff “Hybrid”

Note that

- backwards-difference yields zeros at $z = 0$
- bilinear transform yields zeros at $z = -1$

Try other locations... such as . . .

Zeros at $z = 0.3$ (Stilson 1996)

Reference: Tim Stilson’s June 2006 CCRMA thesis
Zeros at $z = 0.3$ — Bode Plot

Overview
- Early Voice Models
- Digital Voice Models
- Karplus Strong
- Digital Waveguides
- Single Reeds
- Bowed Strings
- Distortion Guitar
- Commuted Synthesis
- Digital Waveguide Mesh
- Horn Filter Design
- Finite Differences
- Virtual Analog
  - Virtual Analog
  - Moog VCF Ladder
  - VCF Structure
  - Moog VCF Controls
  - VCF Digitization
  - BLT Bode
  - BackDiff Bode
  - BLT/BackDiff Hybrid
- Bode Plot
  - Wave Digital Filters
Julius Smith AES-2006 Masterclass – 121 / 101
Wave Digital Filters
Wave Digital Filters

A Wave digital filter (WDF) is a particular kind of digital filter (or finite difference scheme) based on physical modeling principles.

- Developed to digitize lumped electrical circuit elements:
  - inductors
  - capacitors
  - resistors
  - gyrators, circulators, etc., (classical circuit theory)
- Each element is digitized by the bilinear transform
- Wave variables are used in place of physical variables (new), yielding superior numerical properties.
- Element connections involve wave scattering
Wave Digital Filter Construction

Wave digital elements may be derived as follows:

1. Express forces and velocities as *sums of traveling-wave components* ("wave variables"):

\[
\begin{align*}
  f(t) &= f^+(t) + f^-(t) \\
  v(t) &= v^+(t) + v^-(t)
\end{align*}
\]

The actual "travel time" is always zero.

2. Digitize via the *bilinear transform* (trapezoid rule)

3. Use *scattering junctions* ("adaptors") to connect elements together in series and/or parallel.
Physical Construction of Traveling-Wave Element Interfaces

Overview

Early Voice Models
Digital Voice Models
Karplus Strong
Digital Waveguides
Single Reeds
Bowed Strings
Distortion Guitar
Committed Synthesis
Digital Waveguide Mesh
Horn Filter Design
Finite Differences
Virtual Analog
Wave Digital Filters
  - Wave Digital Filters
  - Wave Digital Filter Construction
  - Physical Construction of Traveling-Wave Element Interfaces
  - Points to Note
  - Element Reflectance
  - Reflectance of an Ideal Mass
  - Simplifying the

\[ F^+(s) + V(s) \]

\[ R_0 \]

\[ F^-(s) \]

\[ R(s) \]

\[ F^+(s) \]

\[ F^-(s) \]

\[ R_0 \]

\[ T(s) \]

\[ K(s) \]

\[ K_R(s) \]

\[ F^R_+(s) \]

\[ F^R_-(s) \]
Points to Note

- The inserted waveguide impedance $R_0$ is \textit{arbitrary} because it was \textit{physically introduced}.
- The element now interfaces to other elements by abutting its waveguide (transmission line) to that of other element(s).
- Such junctions involve \textit{lossless wave scattering}:

\[
\begin{align*}
F_R^+(s) &= T(s)F_R^+(s) + K_R(s)F_R^-(s) \\
F_R^-(s) &= T_R(s)F_R^-(s) + K(s)F_R^+(s)
\end{align*}
\]
Element Reflectance

Imposing *physical continuity constraints* across the junction:

\[ F(s) = F_R(s) \]
\[ 0 = V(s) + V_R(s) \]

with

\[ F(s) = F^+(s) + F^-(s) \]
\[ F_R(s) = F^+_R(s) + F^-_R(s) \]
\[ V(s) = V^+(s) + V^-(s) = \frac{F^+(s)}{R_0} - \frac{F^-(s)}{R_0} \]
\[ V_R(s) = V^+_R(s) + V^-_R(s) = \left[ \frac{F^+_R(s)}{R(s)} - \frac{F^-_R(s)}{R(s)} \right] \]

we obtain the *reflection transfer function* ("reflectance") of the element with impedance \( R(s) \):

\[ \hat{\rho}_R(s) \triangleq \frac{F^-(s)}{F^+(s)} = \frac{R(s) - R_0}{R(s) + R_0} \]  \hspace{1cm} (1)
Reflectance of an Ideal Mass

For a mass of $m$ kg, we get

$$ R_m(s) = ms $$

$$ \Rightarrow \hat{\rho}_m(s) = \frac{ms - R_0}{ms + R_0} $$

The mass reflectance is a stable first-order allpass filter (as expected).
Simplifying the Reflectance

Since the impedance $R_0$ of the inserted waveguide is arbitrary, we may set it to

$$ R_0 = m $$

to simplify the reflectance of the mass seen from the waveguide:

$$ \hat{\rho}_m(s) = \frac{ms - m}{ms + m} = \frac{s - 1}{s + 1} \quad \text{(when } R_0 = m) $$

A similar derivation for the ideal spring $k$ gives

$$ \hat{\rho}_k(s) = \frac{1 - s}{1 + s} = -\hat{\rho}_m(s) \quad \text{(when } R_0 = k) $$

For a dashpot $\mu$, setting $R_0 = \mu$ gives

$$ \hat{\rho}_\mu(s) = 0 \quad \text{(when } R_0 = \mu) $$
Bilinear Transformation

To digitize via the bilinear transform, we make the substitution

\[
s = \frac{c (1 - z^{-1})}{1 + z^{-1}}
\]

where \( c > 0 \) is arbitrary (usually \( c = 2/T \)).

Solving for \( z^{-1} \) gives the inverse bilinear transform:

\[
z^{-1} = \frac{1 - s/c}{1 + s/c}
\]

Setting \( c = 1 \) gives a particularly simple formula:

\[
z^{-1} = \frac{1 - s}{1 + s} = \hat{\rho}_k(s) = -\hat{\rho}_m(s)
\]
Resulting Wave Digital Elements

- Thus, under this choice of bilinear transform, we get extremely simple elementary reflectances:

  \[
  \text{ideal spring} \rightarrow \text{unit delay} \\
  \text{ideal mass} \rightarrow \text{unit delay and sign inversion} \\
  \text{ideal dashpot} \rightarrow 0
  \]

- The element values remain only in the waveguide-interface impedances \( R_0 = k, m, \mu \).

- Note that there is no delay free path through the digitized elements (important for modularity)
Wave Digital Elements

- “Wave digital mass” (interface impedance $m$)

\[ \hat{\rho}_m(z) = -z^{-1} \]

- “Wave digital spring” (interface impedance $k$)

\[ \hat{\rho}_k(z) = z^{-1} \]

- “Wave digital dashpot (interface impedance $\mu$)

\[ \hat{\rho}(z) = 0 \]

where

\[ \hat{\rho}_x(z) \triangleq \hat{\rho}_x \left( \frac{1 - z^{-1}}{1 + z^{-1}} \right) \]

These are the digital reflectances of the basic “circuit elements.”
Wave Digital Mass Derivation

For an ideal mass $m$, we have the *driving point impedance*

$$R(s) = ms$$

which, when used to terminate a waveguide of impedance $R_0$, gives the reflectance

$$\hat{\rho}_m(s) = \frac{ms - R_0}{ms + R_0}$$

(continuous time, Laplace domain). Setting $R_0 = m$ gives

$$\hat{\rho}_m(s) = \frac{s - 1}{s + 1}$$
Digitizing the Mass Reflectance

Digitizing using the bilinear transform gives the digital reflectance

\[ \hat{\rho}_m(z) \triangleq \hat{\rho}_m \left( \frac{1 - z^{-1}}{1 + z^{-1}} \right) = -z^{-1} \]

The corresponding *difference equation* is then simply

\[ f^-(n) = -f^+(n - 1) \]  

(wave digital mass).
Elementary Wave Flow Diagrams

Overview

Early Voice Models
Digital Voice Models
Karplus Strong
Digital Waveguides
Single Reeds
Bowed Strings
Distortion Guitar
Committed Synthesis
Digital Waveguide Mesh
Horn Filter Design
Finite Differences
Virtual Analog
Wave Digital Filters
- Wave Digital Filters
- Wave Digital Filter Construction
- Physical Construction of Traveling-Wave Element Interfaces
Points to Note
Element Reflectance
Reflectance of an Ideal Mass
Simplifying the

Wave digital mass

Wave digital spring

Wave digital dashpot
Example: “Piano hammer in flight”

Mass $m$ at constant velocity:

(a) \[ f^+(n) \]

(b) \[ z^{-1} \]

\[ x(n) \]

- State variable is in units of force \( x(n) \overset{\Delta}{=} f^+(n - 1) \)
- Physical force is \( f(n) = f^+(n) + f^-(n) \equiv 0 \)
- Nonzero state variable \( \Rightarrow \) nonzero mass velocity
Mass Velocity

Force-wave simulations easily provide velocity outputs:

\[ v(n) = v^+(n) + v^-(n) = \frac{f^+(n)}{m} - \frac{f^-(n)}{m} \]

\[ = \frac{x(n)}{m} + \frac{x(n)}{m} = \frac{2}{m} x(n) \]

Thus, the mass velocity is simply the state variable \( x(n) \) scaled by \( 2/m \).
Spring and Free Mass

\[ f(t) \rightarrow k \rightarrow v_m(t) \]

\[ f_k(t) = f_m(t) = f(t) \]

Physical Diagram

Electrical Equivalent Circuit

Digitization (BLT)

Three parallel branches \(\Rightarrow\) *three-port parallel adaptor* needed
By flipping an element reference direction, we could realize also as a **series** connection.
Expanded Wave Digital Mass-Spring Oscillator

Wave variables to Physical Variables:

\[ f_k(n) = f_k^+(n) + f_k^-(n) \]  \hspace{1cm} \text{(Spring Force)}

\[ f_m(n) = f_m^+(n) + f_m^-(n) \]  \hspace{1cm} \text{(Mass Force)}

Reflection coefficient:

\[ \rho = \frac{m - k}{k + m} \]
Normalized Wave Digital Piano Hammer
Wave Digital Mass-Spring Oscillator

\[ f_k^-(n) \]
\[ x_1(n) \]
\[ z^{-1} \]
\[ f_k^+(n) \]
\[ 1 + \rho \]
\[ -\rho \]
\[ 1 - \rho \]
\[ f_m^-(n) \]
\[ z^{-1} \]
\[ x_2(n) \]
Normalized Wave Scattering

Convert force waves to \textit{root-power waves}:

\[
\tilde{f}_i^+ \triangleq \frac{f_i^+}{\sqrt{R_i}} \\
\tilde{v}_i^+ \triangleq v_i^+ \cdot \sqrt{R_i}
\]

where $R_i$ = wave impedance in waveguide $i$. 
Normalized Wave Digital Mass-Spring Oscillator

- Stored energy *invariant* with respect to *time varying* and/or *nonlinear* changes in mass or spring constants
- For the piano hammer, the felt spring constant may vary with force $f_k(n)$ without altering stored energy
- Only the reflection coefficient $\rho(n)$ varies as the spring is compressed
- Note delay-free interdependence of $\rho(n)$ and $f_k(n)$
- See Bensa & Bilbao et al. (SMAC-03) for an update on the *lossy* normalized wave digital piano hammer
- *Four multiplies, two additions* required in this form
- Using transformer normalization, we can obtain *three-multiply, three-add* variations (see Mathews-Smith SMAC-03 paper on...
Transformer Normalized Wave Scattering

\[ g_i(t) = \sqrt{\frac{1 - k_i(t)}{1 + k_i(t)}} \]

\[ f_{i-1}^+(t-T) \rightarrow \frac{1}{g_i(t)} \rightarrow f_{i-1}^-(t+T) \rightarrow f_i^+(t) \]

\[ R_i \rightarrow R_{i-1} \]

\[ f_i^-(t) \rightarrow R_i \]

General transformer-normalized scattering junction
Normalized Waveguide Oscillator/Resonator

Overview
Early Voice Models
Digital Voice Models
Karplus Strong
Digital Waveguides
Single Reeds
Bowed Strings
Distortion Guitar
Commuted Synthesis
Digital Waveguide Mesh
Horn Filter Design
Finite Differences
Virtual Analog
Wave Digital Filters
Normalized Wave
Digital Piano Hammer
  • Wave Digital
  • Mass-Spring Oscillator
  • Normalized Wave
  • Scattering
  • Normalized Wave
  • Digital Mass-Spring
Structural Losslessness - Two-Port Case

- All one-multiply scattering junctions are structurally lossless:
  - The one multiply can only affect oscillation frequency

- Not all normalized scattering junctions are structurally lossless
  - The four-multiply normalized junction has two parameters, \( s \triangleq \rho \) and \( c \triangleq \sqrt{1 - \rho^2} \), which may not satisfy \( s^2 + c^2 = 1 \) after quantization to finite wordlengths
Passive Nonlinear Piano Hammer

Overview
Early Voice Models
Digital Voice Models
Karplus Strong
Digital Waveguides
Single Reeds
Bowed Strings
Distortion Guitar
Committed Synthesis
Digital Waveguide Mesh
Horn Filter Design
Finite Differences
Virtual Analog
Wave Digital Filters
Normalized Wave
Digital Piano Hammer
- Wave Digital
  Mass-Spring Oscillator
- Normalized Wave
  Scattering
- Normalized Wave
  Digital Mass-Spring

\[ \tilde{f}_k(n) \]

\[ \sqrt{1 + \rho^2} \]

\[ \tilde{f}_m(n) \]

\[ \sqrt{1 - \rho^2} \]

normalized wave digital spring

normalized wave digital mass

\[ \rho \]

\[ -\rho \]

\[ z^{-1} \]
Passive Nonlinear Piano Hammer in Transformer Normalized Form

- Mass-spring oscillator is *structurally lossless* when damping is zero.
- *Energy invariant* in the *non-linear* and *time-varying* cases by construction.