History and Practice of Digital Sound Synthesis

Julius Smith
CCRMA, Stanford University

AES-2006 Heyser Lecture

October 6, 2006

Overview

Outline

Digital sound synthesis approaches in approximate historical order:

- Wavetable (one period)
- Subtractive
- Additive
- Frequency Modulation (FM)
- Sampling
- Spectral Modeling
- Physical Modeling

Some connections with audio coding will be noted

Emphasis:

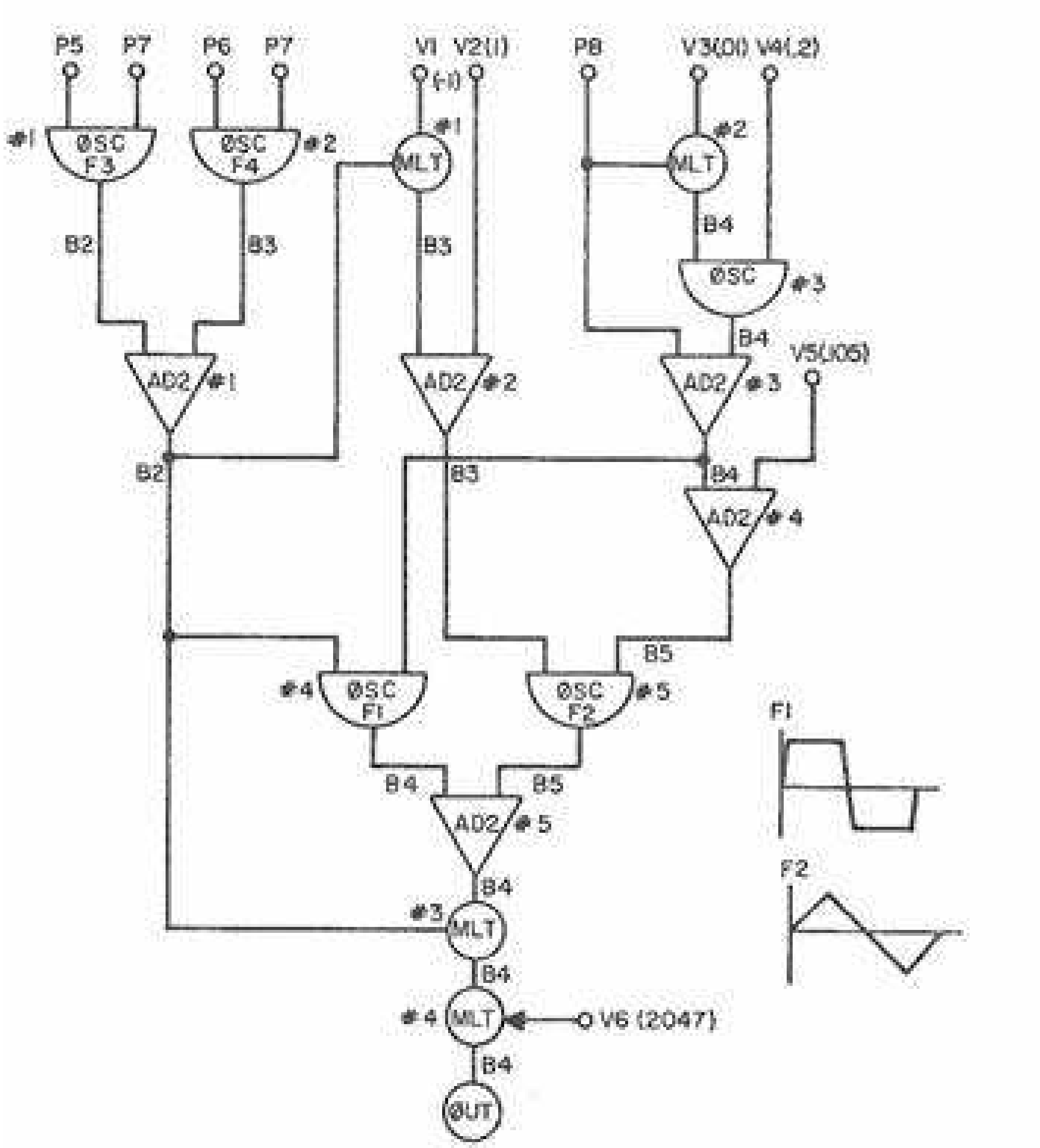
- Sound examples
- Block diagrams
- Historical notes

CCRMA Perspective

The Knoll, Stanford University

Early Digital Sound Synthesis

Wavetable Synthesis in Music I-V (1957-1969)

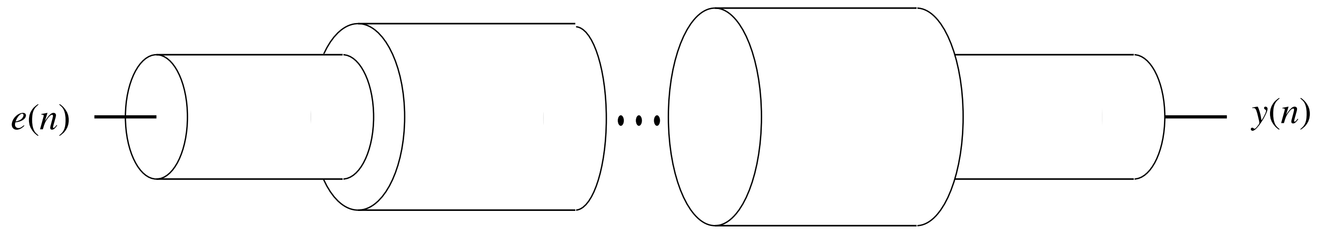


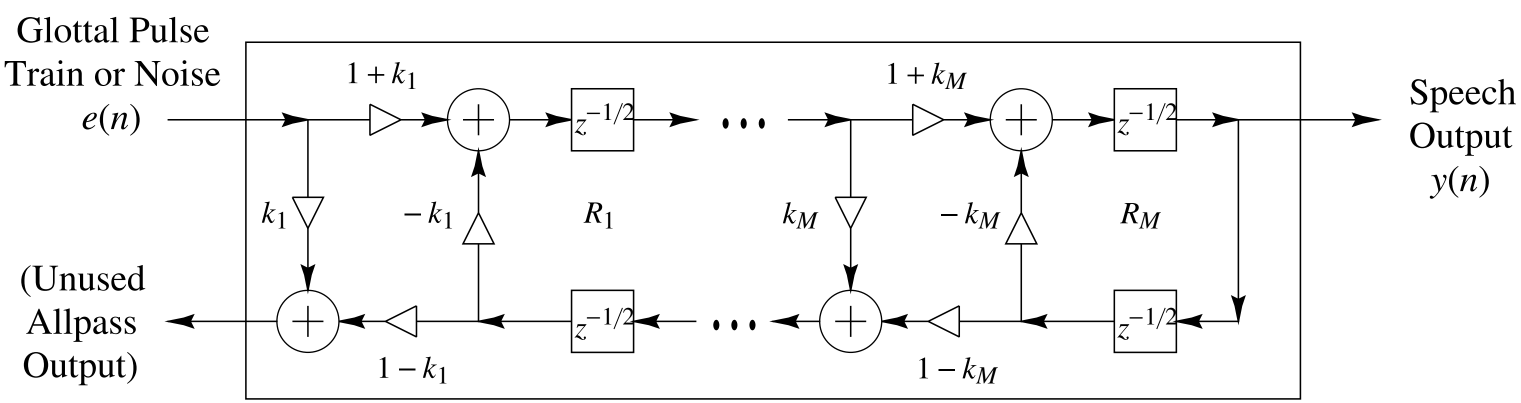
Music V Scripting Language ("Note Cards")

```
INS 0 4 ;
ØSC P5 P7 B2 F3 P30 ;
ØSC P6 P7 B3 F4 P29 ;
AD2 B2 B3 B2 ;
MLT B2 V1 B3 ;
AD2 B3 V2 B3 ;
MLT P8 V3 B4 ;
ØSC B4 V4 B4 F5 P28 ;
AD2 P8 B4 B4 ;
AD2 B4 V5 B5 ;
ØSC B3 B5 B5 F2 V7 ;
ØSC B2 B4 B4 F1 V8 ;
MLT B2 B4 B4 ;
MLT B4 V6 B4 ;
ØUT B4 B1 ;
END ;
```

- Essentially Supported in MPEG-4 Structured Audio Orchestra Language (SAOL) (Music V $\longrightarrow$ `csound` $\longrightarrow$ SAOL)
- "Encoding sounds" as "instruments" is hard, in general

Kelly-Lochbaum Vocal Tract Model





Kelly-Lochbaum Vocal Tract Model (Piecewise Cylindrical)

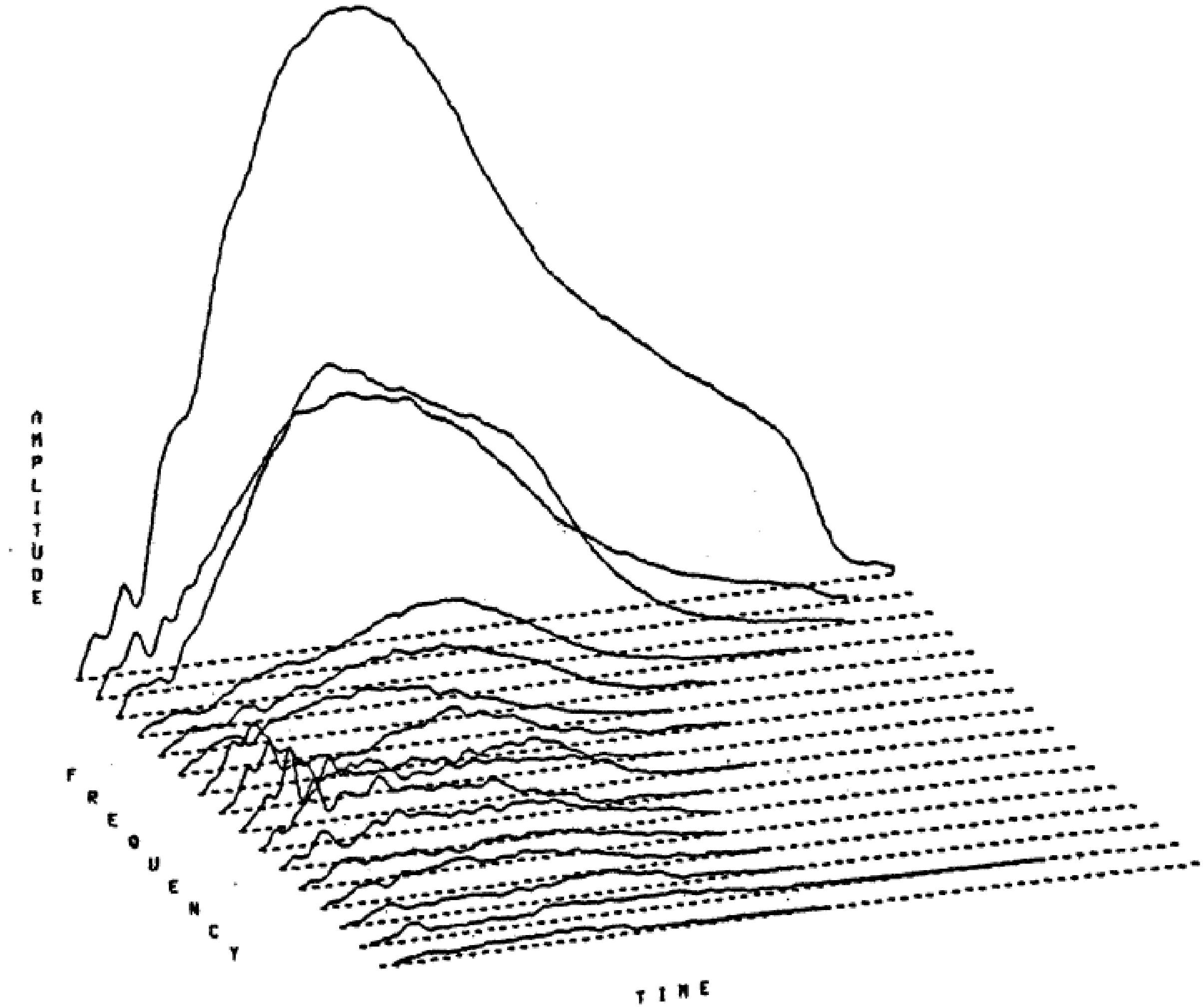
John L. Kelly and Carol Lochbaum (1962)

Sound Example

"Bicycle Built for Two": (WAV) (MP3)

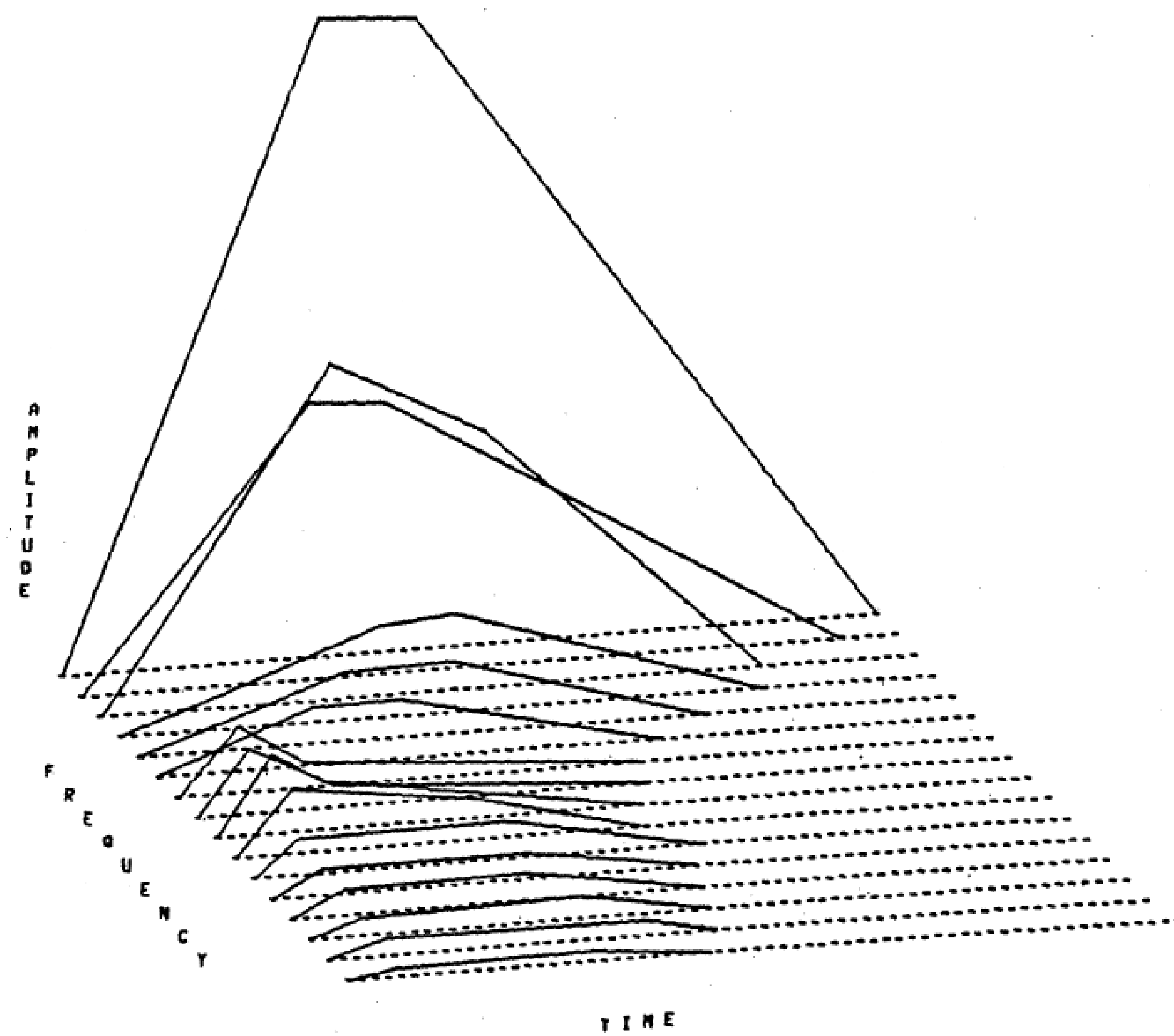
- Vocal part by Kelly and Lochbaum (1961)
- Musical accompaniment by Max Mathews
- Computed on an IBM 704
- Based on Russian speech-vowel data from Gunnar Fant's book
- Probably the first digital physical-modeling synthesis sound example by any method
- Inspired Arthur C. Clarke to adapt it for "2001: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

Classic Additive-Synthesis Analysis (Heterodyne Comb)



John Grey 1975 — CCRMA Tech. Reports 1 & 2
(CCRMA "STANM" reports — available online)

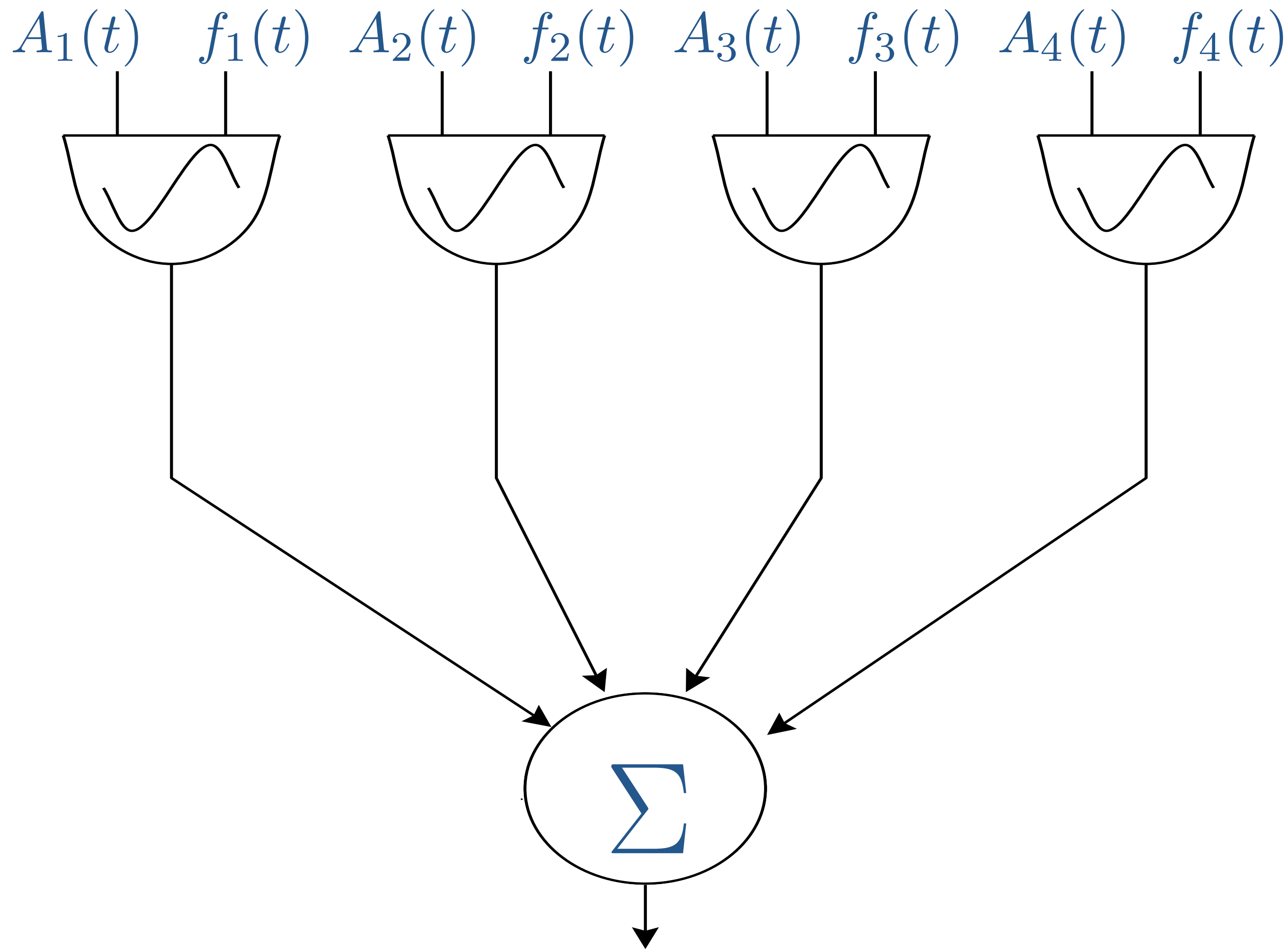
Classic Additive-Synthesis (Sinusoidal Oscillator Envelopes)



John Grey 1975 — CCRMA Tech. Reports 1 & 2

(CCRMA “STANM” reports — available online)

Classic Additive Synthesis Diagram



$$y(t) = \sum_{i=1}^{4} A_i(t) \sin\left[\int_0^t \omega_i(t)dt + \phi_i(0)\right]$$

Classic Additive-Synthesis Examples

- Bb Clarinet
- Eb Clarinet
- Oboe
- Bassoon
- Tenor Saxophone
- Trumpet
- English Horn
- French Horn
- Flute

- All of the above
- Independently synthesized set

(Synthesized from original John Grey data)

Frequency Modulation (FM) Synthesis

FM synthesis is normally used as a *spectral modeling* technique.

- Discovered and developed (1970s) by John M. Chowning (CCRMA Founding Director)
- Key paper: JAES 1973 (vol. 21, no. 7)
- Commercialized by Yamaha Corporation:
  - DX-7 synthesizer (1983)
  - OPL chipset (SoundBlaster PC sound card)
  - Cell phone ring tones

- On the physical modeling front, synthesis of vibrating-string waveforms using *finite differences* started around this time: Hiller & Ruiz, JAES 1971 (vol. 19, no. 6)

FM Formula

$$x(t) = A_c \sin[\omega_c t + \phi_c + A_m \sin(\omega_m t + \phi_m)]$$

where

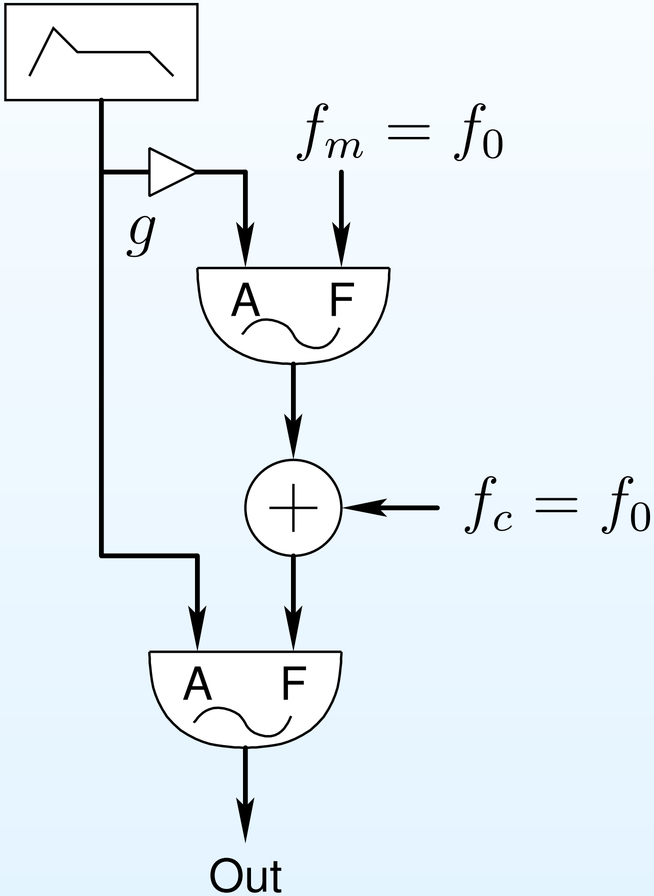
(A_c, ω_c, ϕ_c) specify the *carrier* sinusoid
(A_m, ω_m, ϕ_m) specify the *modulator* sinusoid

Can also be called *phase modulation*

Simple FM "Brass" Patch (1970–)

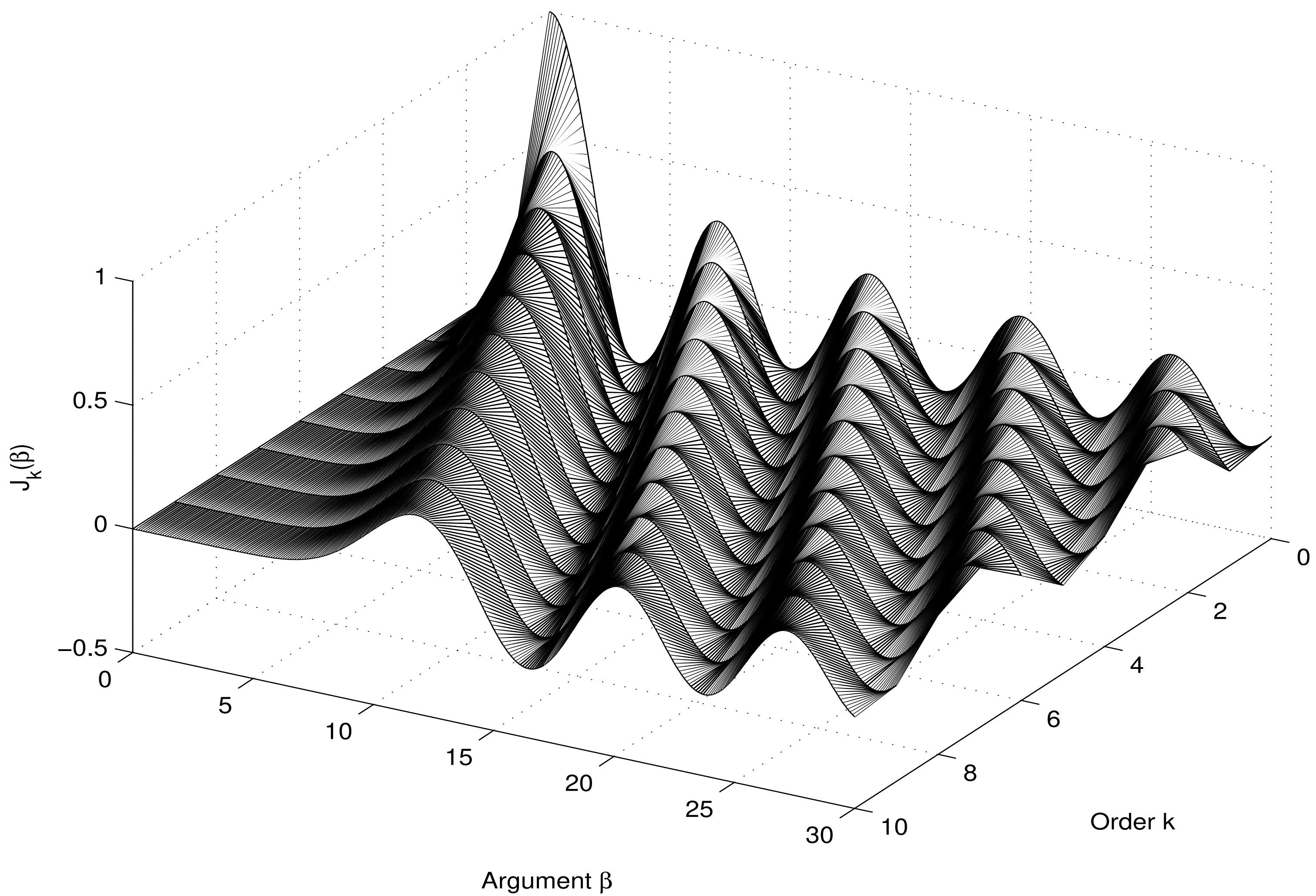
Jean-Claude Risset observation (1964–1969):

Brass bandwidth $\propto$ amplitude



FM Harmonic Amplitudes (Bessel Function of First Kind)

Harmonic number k, FM index β:



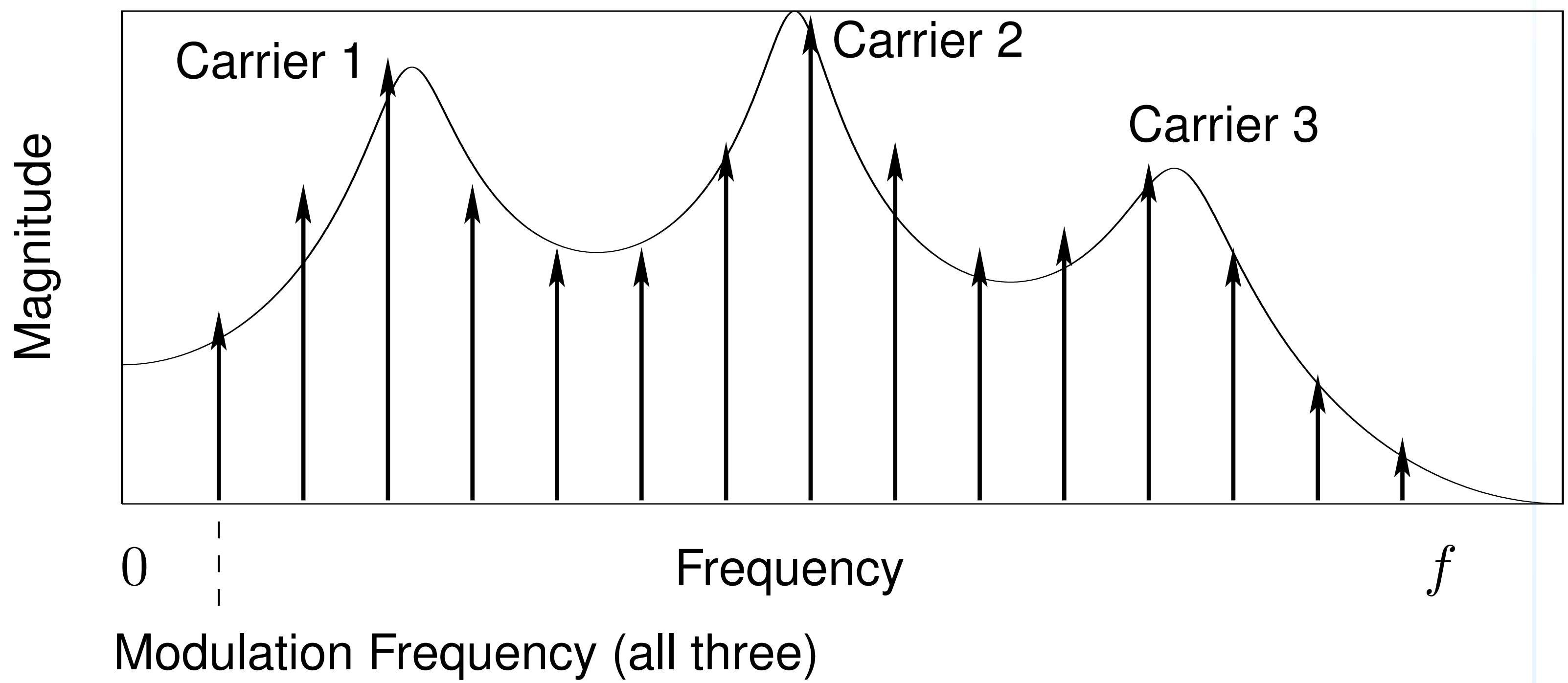
Frequency Modulation (FM) Examples

All examples by John Chowning unless otherwise noted:

- FM brass synthesis
  - Low Brass example
  - Dexter Morril's FM Trumpet
- FM singing voice (1978)
  Each formant synthesized using an FM operator pair
  (two sinusoidal oscillators)
  - Chorus
  - Voices
  - Basso Profundo
- Other early FM synthesis
  - Clicks and Drums
  - Big Bell
  - String Canon

FM Voice

FM voice synthesis can be viewed as *compressed modeling of spectral formants*



Sampling Synthesis History

- 1979 - Fairlight Computer Music Instrument - 8-bit
  - First commercial sampler
  - Eight voices, 8 bits, 64 KB (4 sec) RAM, 16 kHz (mono)
  - Editing, looping, mixing
  - One could *draw* waveforms and additive-synthesis amplitude envelopes (for each harmonic) with a light pen
  - $25,000–$36,000!
- 1981 - E-mu Systems Emulator
  - First “affordable” sampler ($10,000)
  - Eight voices, 128 K RAM, 8-bit, 80 lb.
- 1986 - Ensoniq Mirage
  - Breakthrough price-point ($1695)
  - Eight voices, 144 K RAM, 8-bit

Modern Sampled Piano

Example:[1]

- 40 Gigabytes on ten DVDs (three sampled pianos)
- Every key sampled
- 4–10 “velocity layers”
- Separate recordings with soft pedal down
- Separate “release” recordings, for multiple striking velocities

[1] Synthogy Ivory, $349 (*Electronic Musician*, October 2006)

Fundamental Problem with Sampling Synthesis

Piano timbre is determined by

- key number (1 byte)
- key velocity (2 bytes more than enough)
- pedal state (1 bit [or byte] per pedal)

Piano control is relatively low-dimensional:

- Less than six bytes of information per note played
- No continuous controls (typically)
- Ratio of total sampled data to one note of control data $\approx$ one billion

Now consider bowed strings

Control parameters:

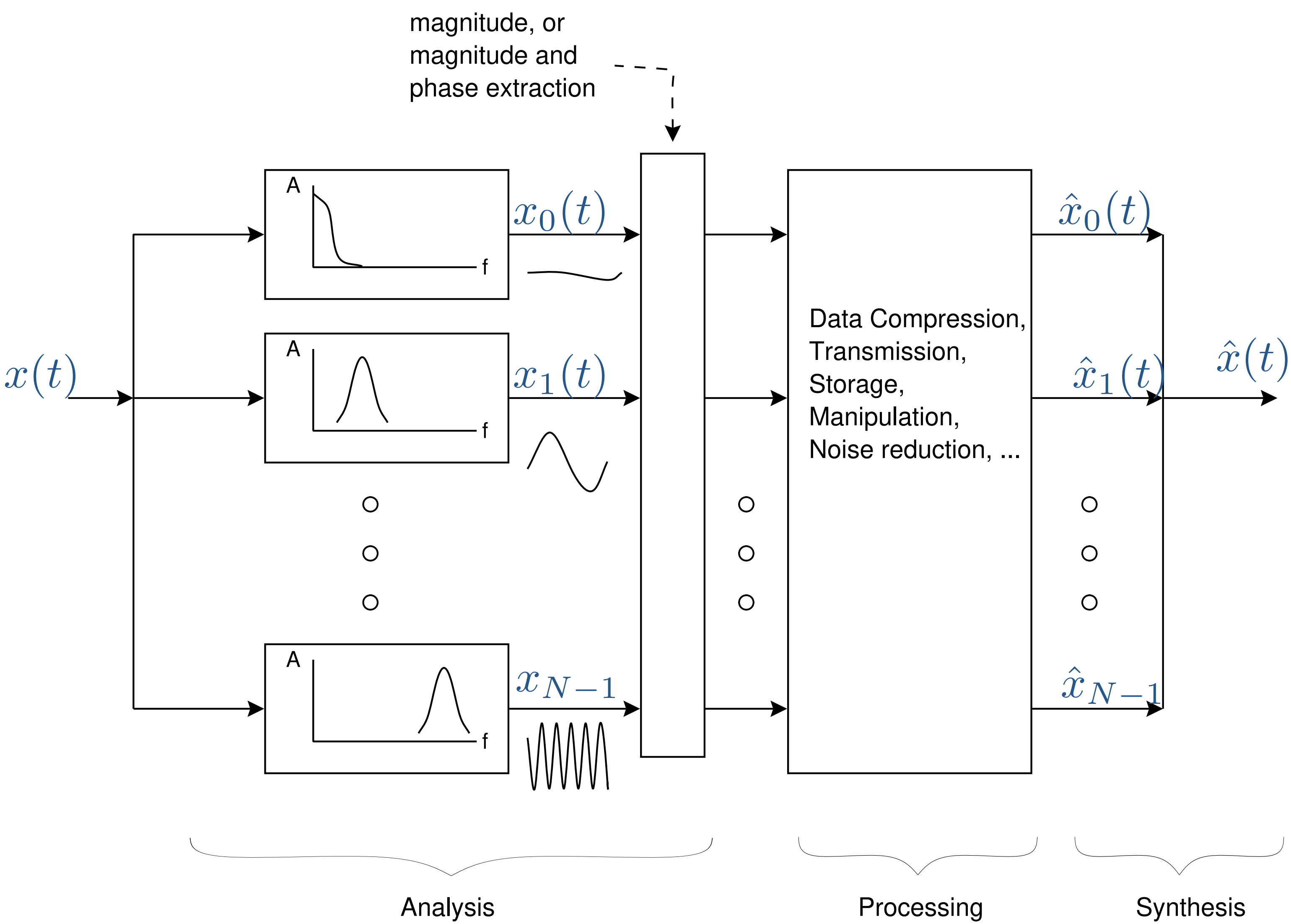
- Left-hand finger position(s)
- Left-hand vibrato
- Bow velocity
- Bow force
- Bow position
- Bow angle
- Shoulder damping
- Instrument orientation
- Player motion (within a room)

Difficulty of sampling bowed strings

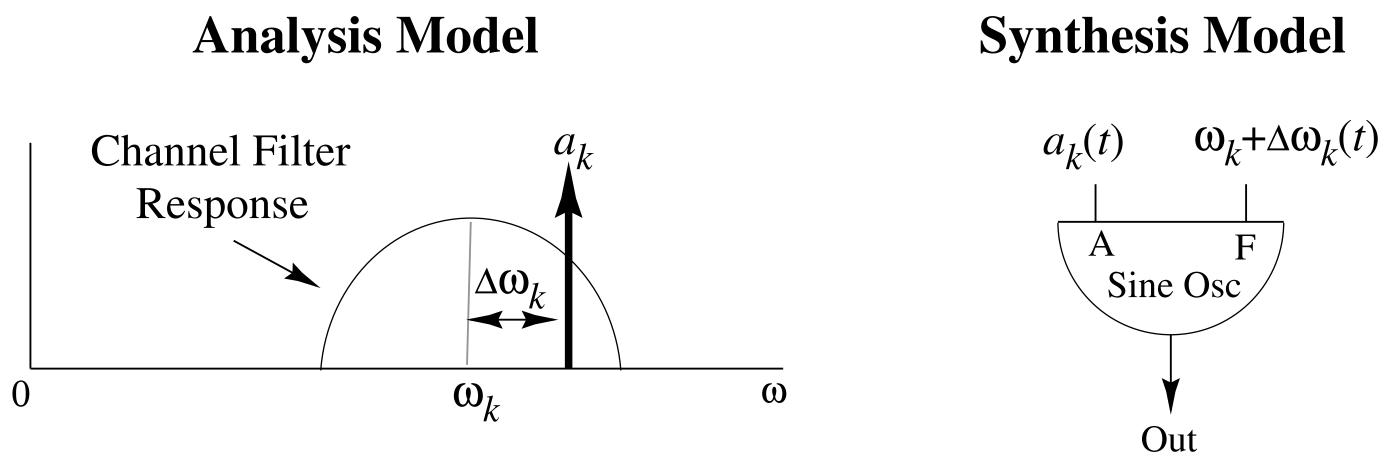
- Bowed-string control is *infinite-dimensional* in principle
- Many *time-varying functions* — “gestures” (we counted more than 10)
- Complete sampling of bowed strings on the level of pianos has apparently never been done
- Rule-driven navigation of the *most useful* recorded playing regimes has worked well (*e.g.*, *Synful Orchestra*)
- *Model*-based approaches greatly reduce data requirements:
  - Spectral models (inspired by sound *perception*)
  - Physical models (model the sound *source*)

Spectral Modeling Synthesis (Historical Summary)

Classic Vocoder Analysis & Resynthesis (Dudley 1939)

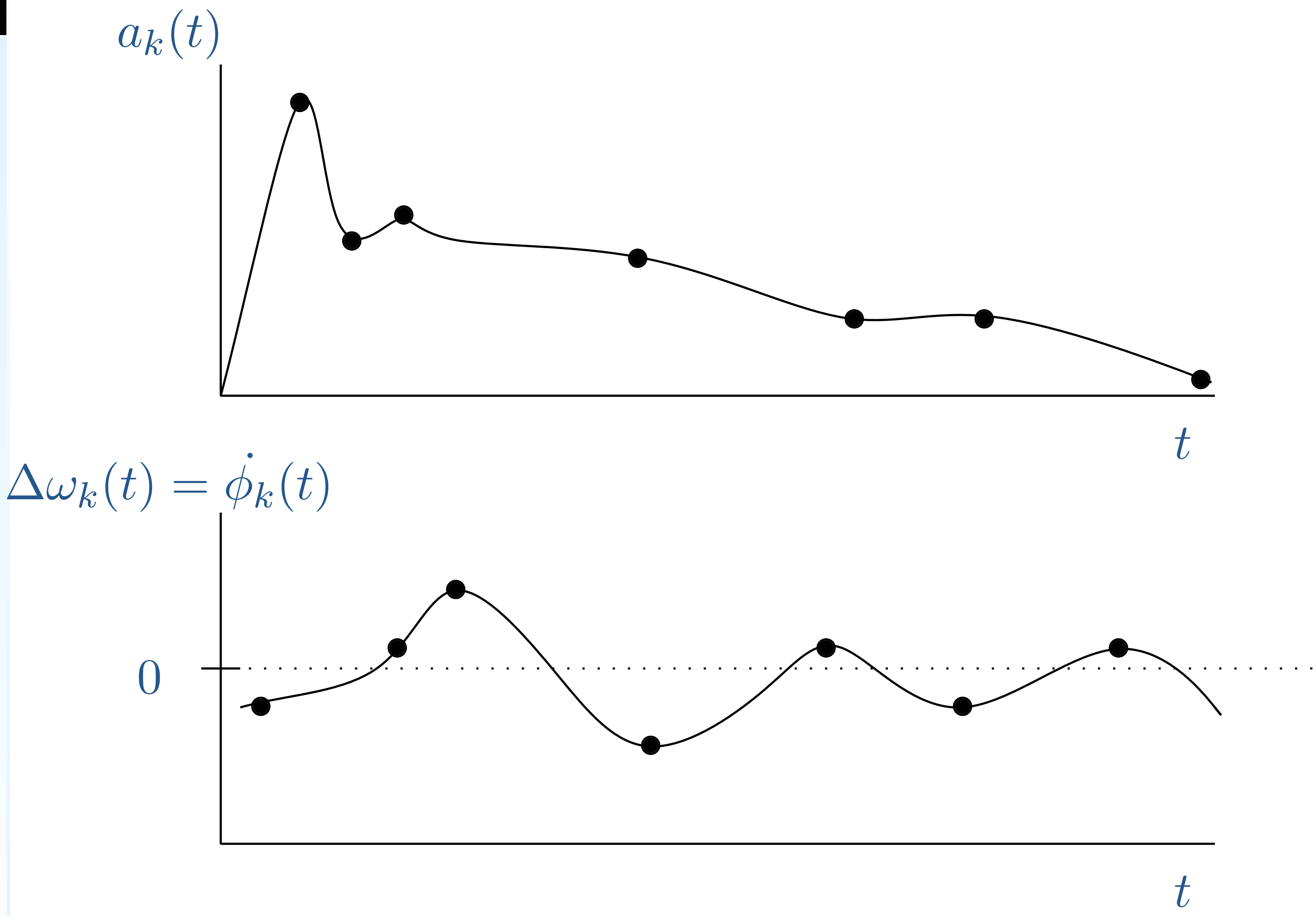


Phase Vocoder Channel Model



- Early "channel vocoder" implementations (hardware) only measured amplitude $a_k(t)$ (Dudley 1939)
- The "phase vocoder" (Flanagan and Golden 1966) added phase tracking in each channel
- Portnoff (1976) developed the FFT phase vocoder, which replaced the heterodyne comb in computer-music additive-synthesis analysis (James A. Moorer)
- Inverse FFT synthesis (Rodet and Depalle 1992) gave faster sinusoidal oscillator banks

Amplitude and Frequency Envelopes



Channel Vocoder Sound Examples

- Original

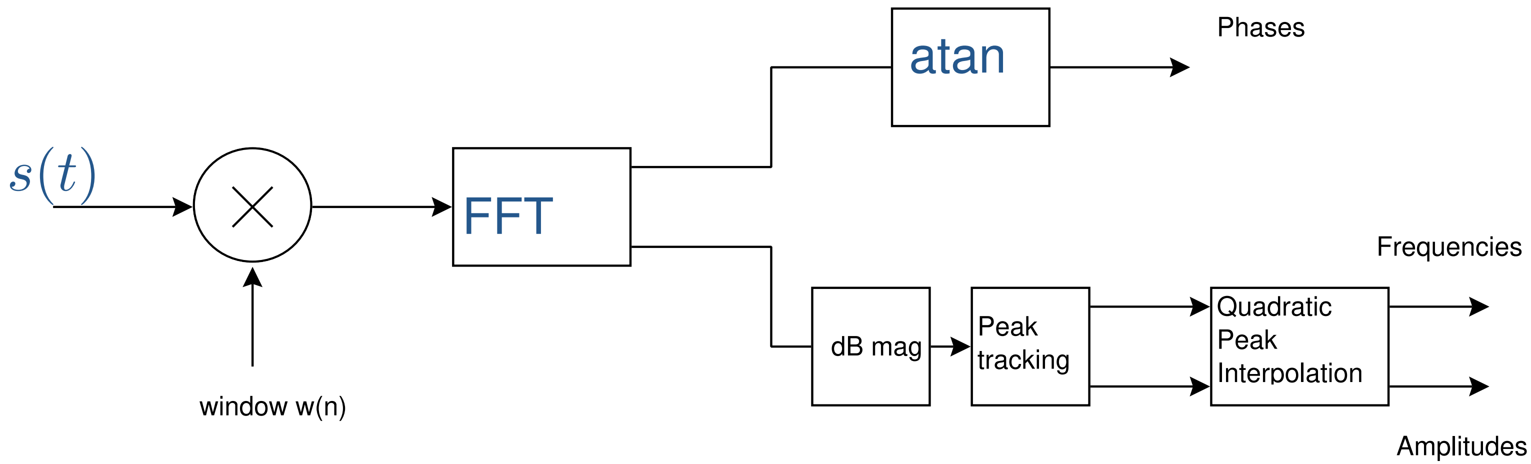
- 10 channels, sine carriers
- 10 channels, narrowband-noise carriers

- 26 channels, sine carriers
- 26 channels, narrowband-noise carriers
- 26 channels, narrowband-noise carriers, channels reversed

- ***Phase* Vocoder:** Identity system in absence of modifications

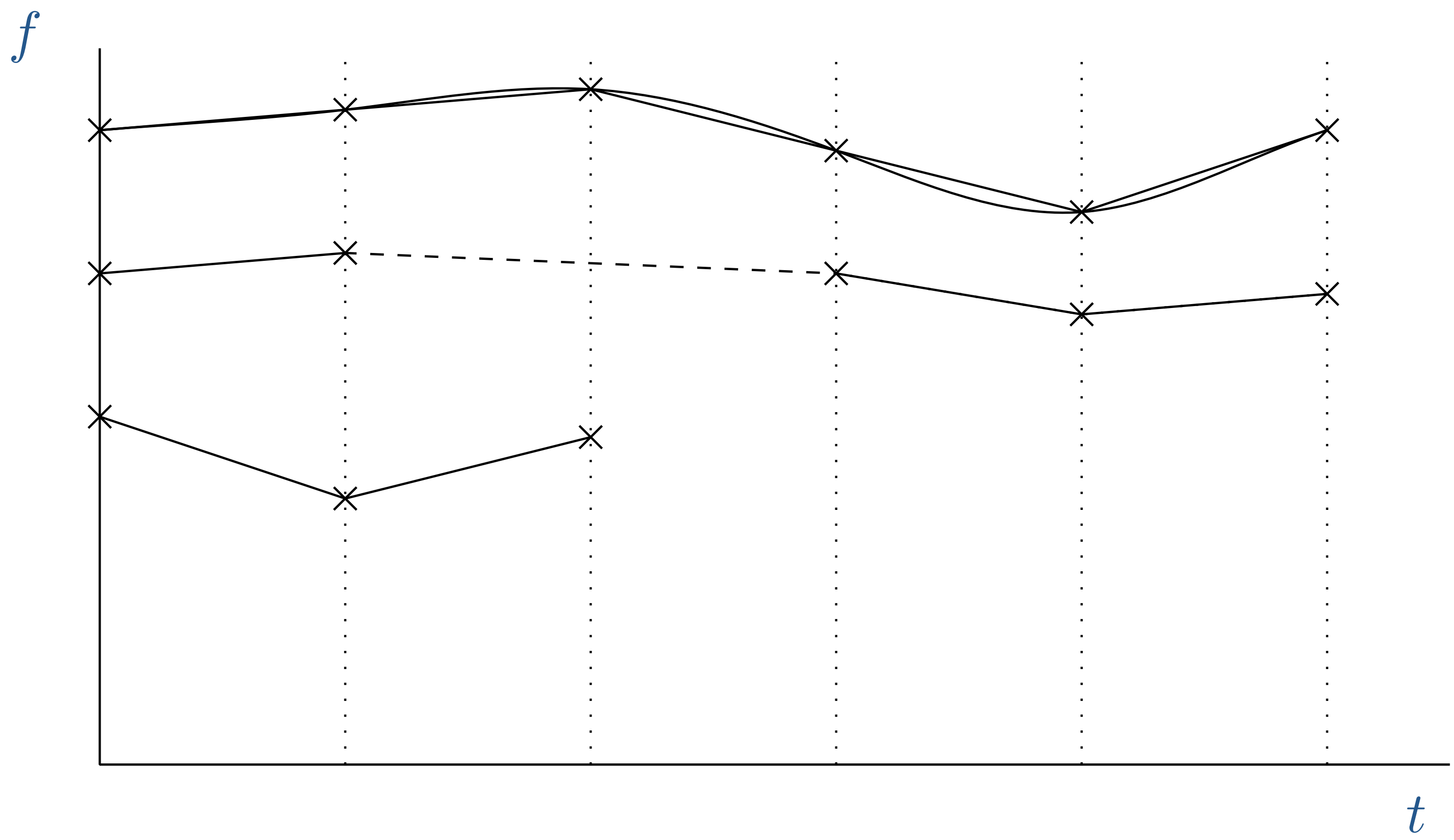
- The FFT Phase Vocoder next transitioned to the Short-Time Fourier Transform (STFT) (Allen and Rabiner 1977)

Tracking Spectral Peaks in the Short-Time Fourier Transform

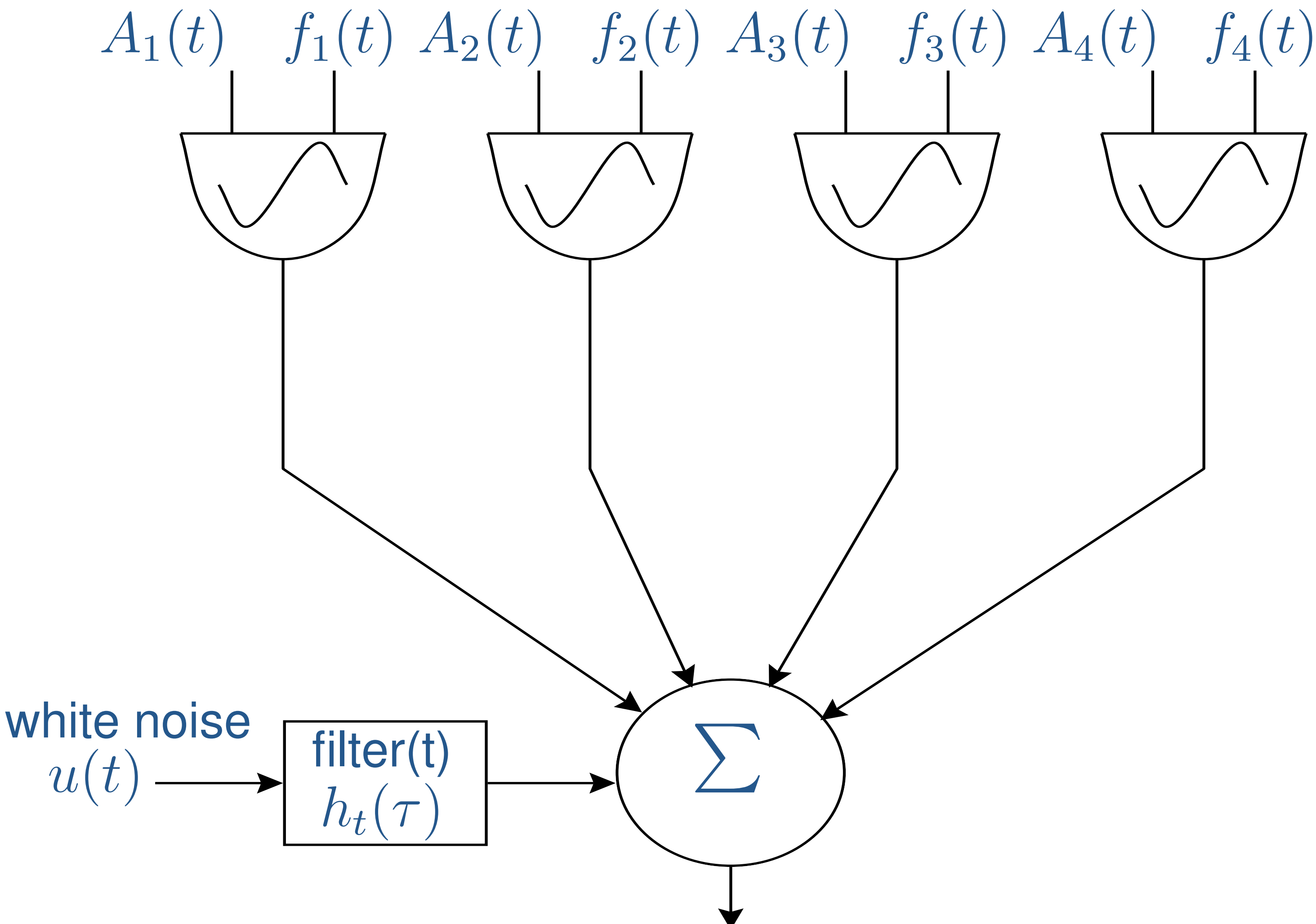


- STFT peak tracking at CCRMA: mid-1980s (PARSHL program)
- Motivated by vocoder analysis of piano tones
- Influences: STFT (Allen and Rabiner 1977), ADEC (1977), MAPLE (1979)
- Independently developed for speech coding by McAulay and Quatieri at Lincoln Labs

Example Spectral Trajectories



Sines + Noise Synthesis (1989)



$$y(t) = \sum_{i=1}^{4} A_i(t) \cos\left[\int_0^t \omega_i(t)dt + \phi_i(0)\right] + (h_t * u)(t)$$

Sines + Noise Sound Examples

Xavier Serra 1989 thesis demos (Sines + Noise signal modeling)

- Piano
  - Original
  - Sinusoids alone
  - Residual after sinusoids removed
  - Sines + noise model
- Voice
  - Original
  - Sinusoids
  - Residual
  - Synthesis

Musical Effects with Sines+Noise Models (Serra 1989)

- Piano Effects
  - Pitch downshift one octave
  - Pitch flattened
  - Varying partial stretching
- Voice Effects
  - Frequency-scale by 0.6
  - Frequency-scale by 0.4 and stretch partials
  - Variable time-scaling, deterministic to stochastic

Cross-Synthesis with Sines+Noise Models (Serra 1989)

- Voice “modulator”

- Creaking ship’s mast “carrier”
- Voice-modulated creaking mast
- Same with modified spectral envelopes

Sines + Transients Sound Examples (Serra 1989)

In this technique, the sinusoidal sum is phase-matched at the cross-over point only (with no cross-fade).

- Marimba
  - Original
  - Sinusoidal model
  - Original attack, followed by sinusoidal model
- Piano
  - Original
  - Sinusoidal model
  - Original attack, followed by sinusoidal model

Multiresolution Sines + Noise + Transients (Levine 1998)

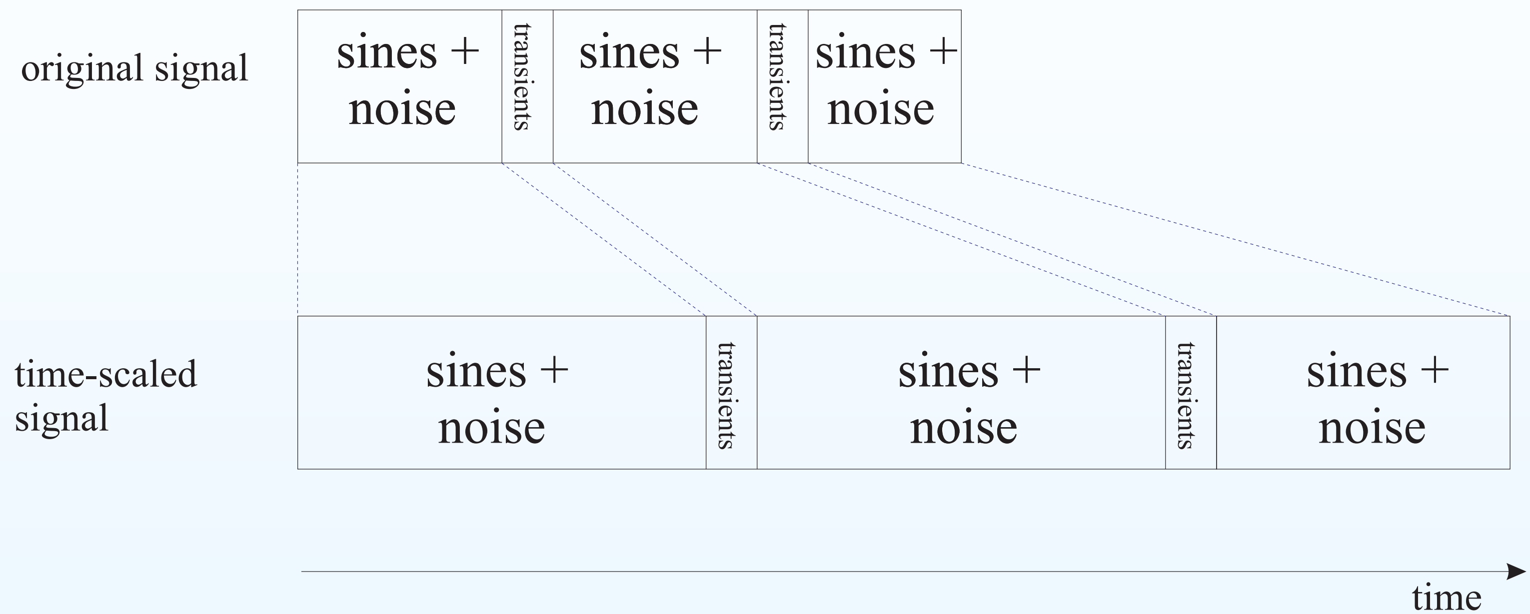
Why Model Transients Separately?

- Sinusoids efficiently model spectral *peaks* over time
- Filtered noise efficiently models spectral *residual* vs. t
- Neither is good for *abrupt transients* in the waveform
- Phase-matched oscillators are expensive
- More efficient to switch to a *transient model* during transients
- Need sinusoidal *phase matching* at the switching times

Transient models:

- Original waveform slice (1988)
- Wavelet expansion (Ali 1996)
- MPEG-2 AAC (with short window) (Levine 1998)
- Frequency-domain LPC
  (time-domain amplitude envelope) (Verma 2000)

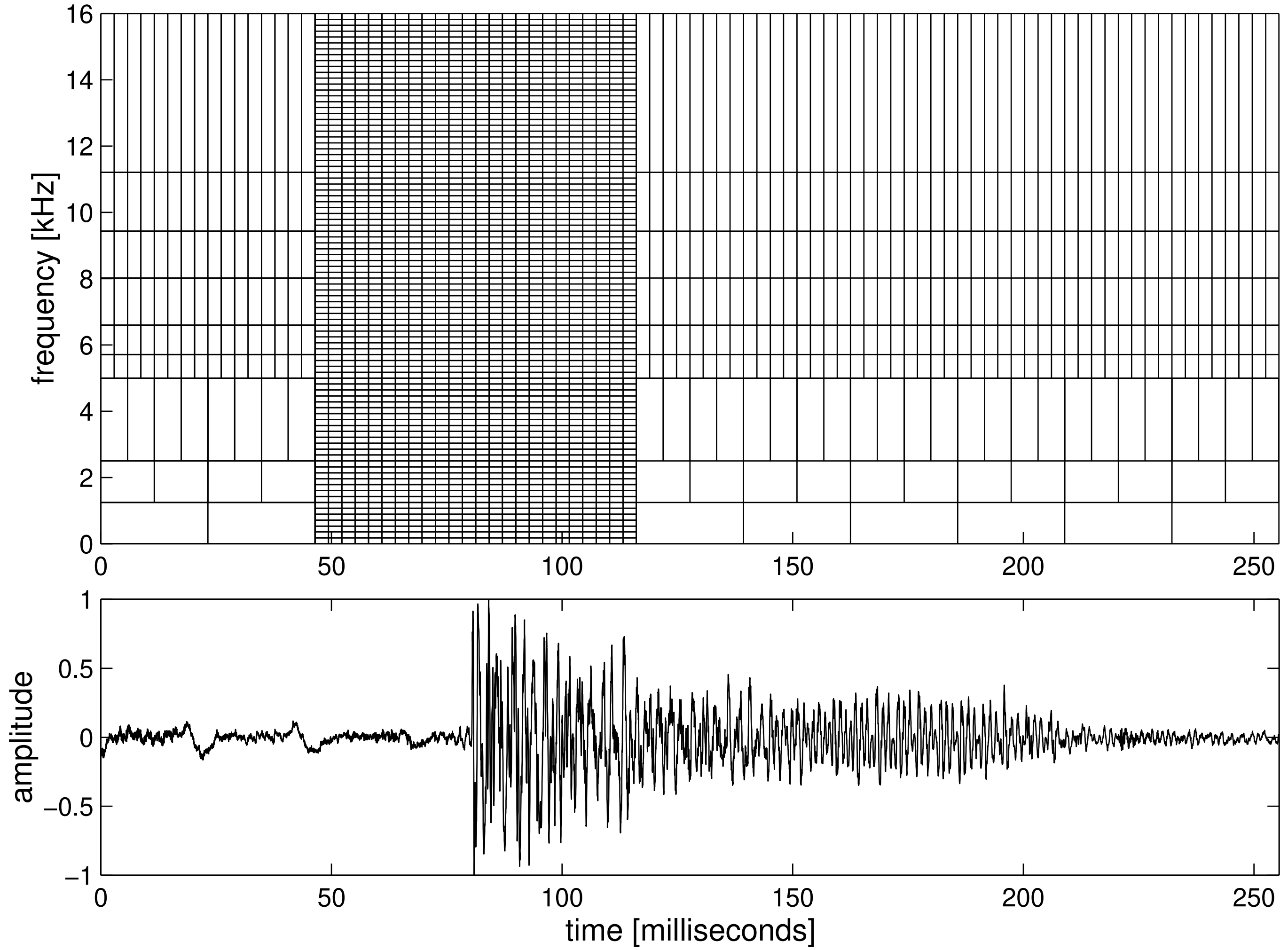
Time Scale Modification of Sines + Noise + Transients Models



Time-Scale Modification (TSM) becomes *well defined*:

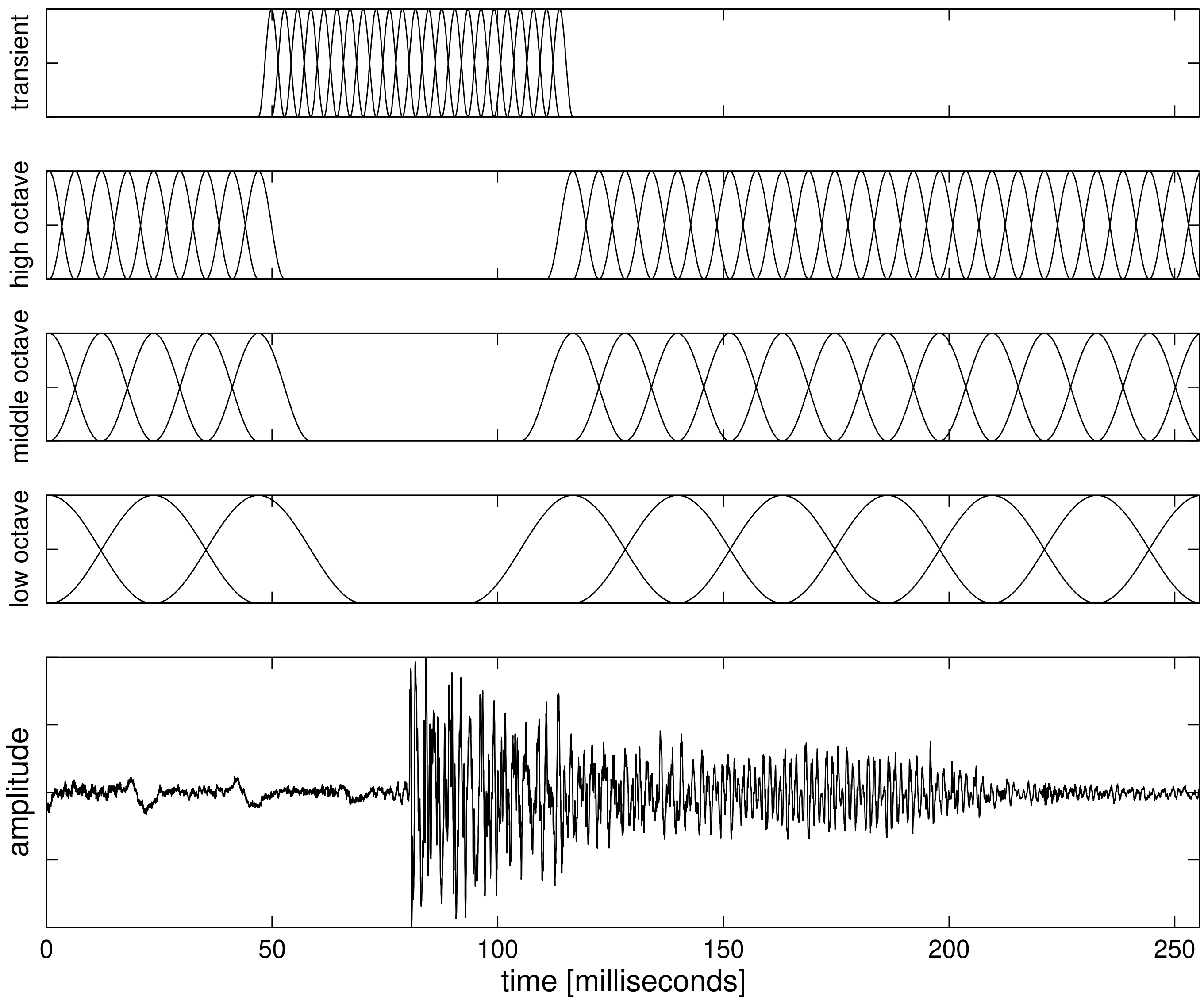
- Transients are *translated* in time
- Sinusoidal envelopes are *scaled* in time
- Noise-filter envelopes also *scaled* in time
- Dual of TSM is *frequency scaling*

Sines + Noise + Transients Time-Frequency Map

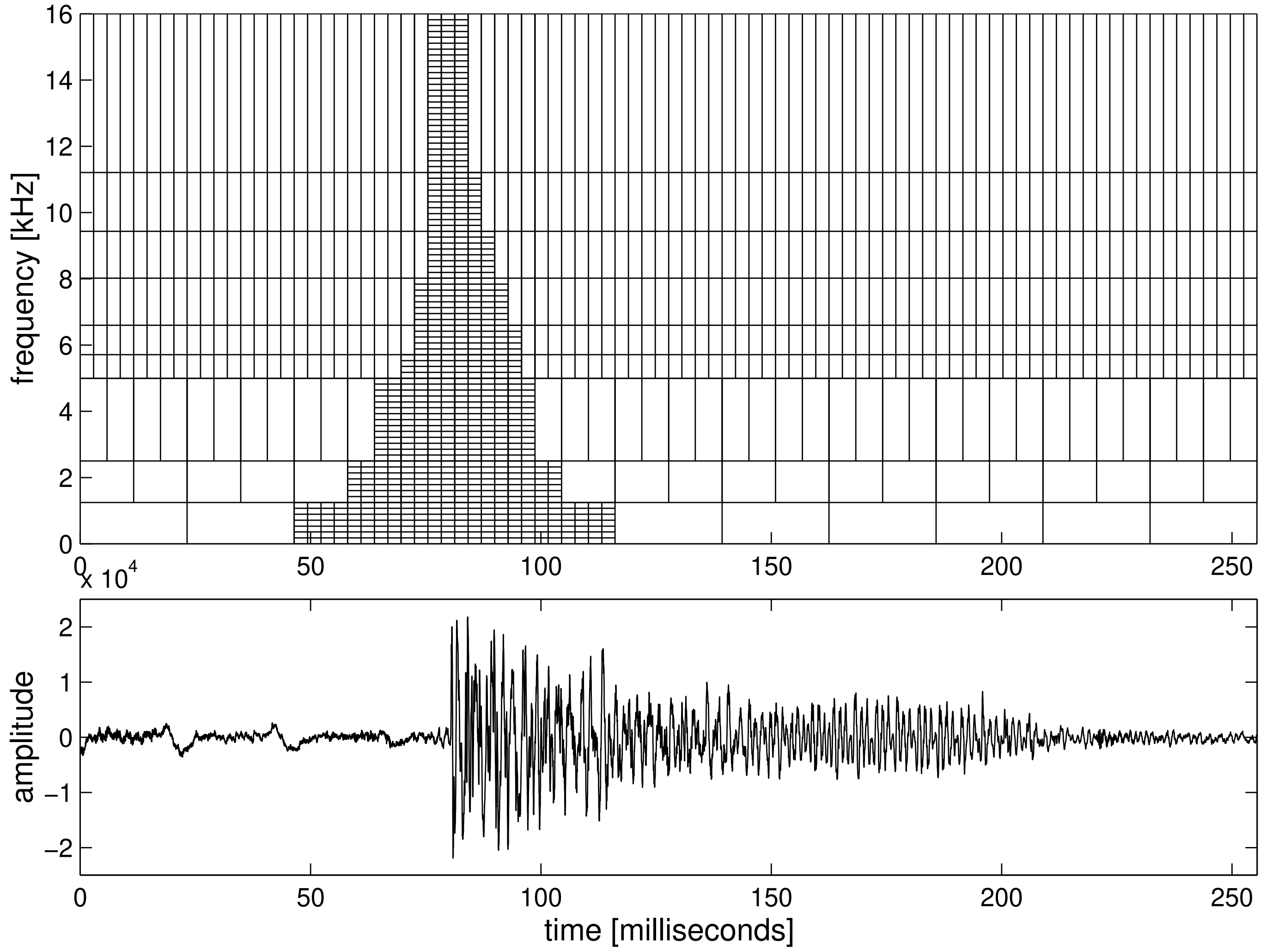


(Levine 1998)

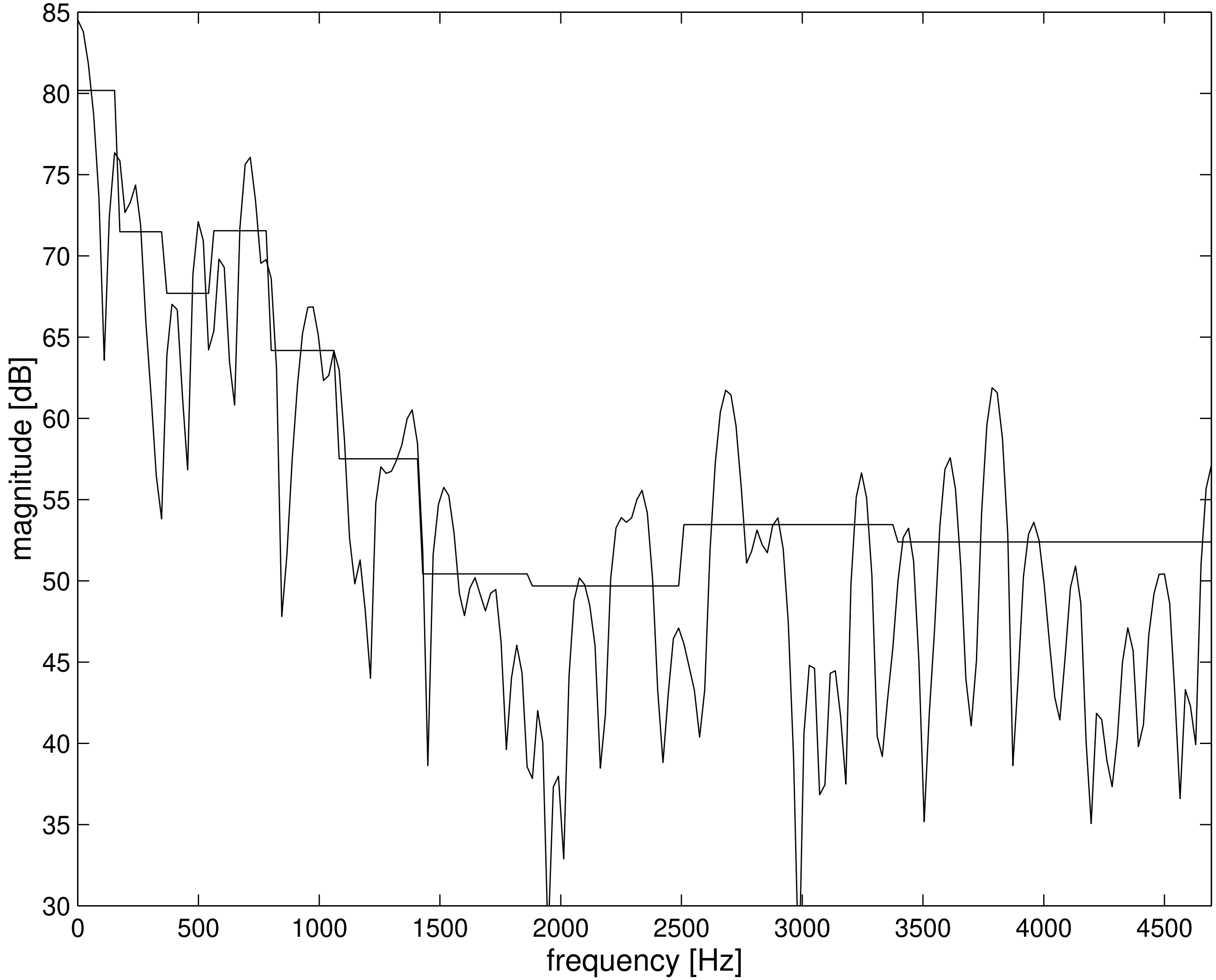
Corresponding Analysis Windows



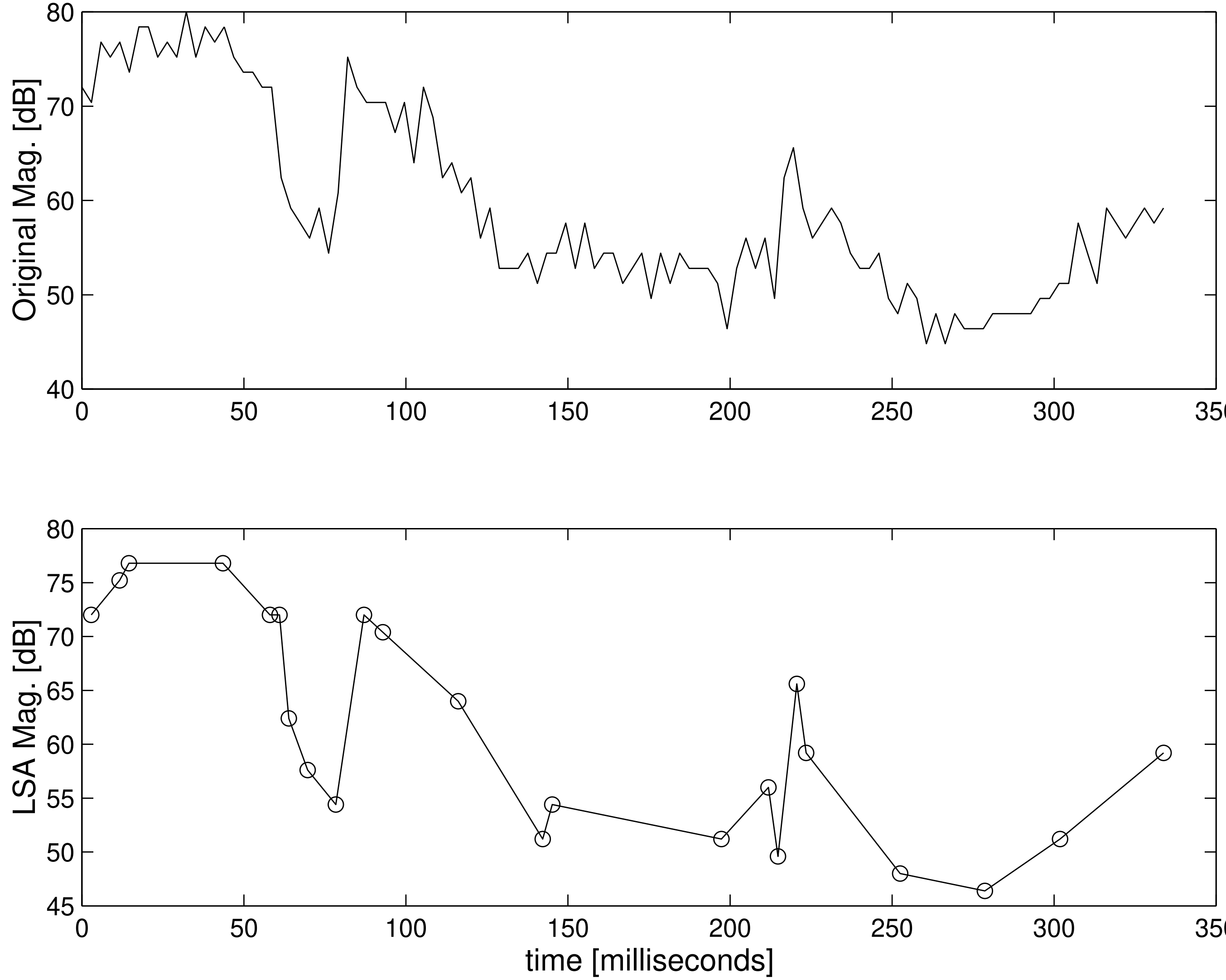
Quasi-Constant-Q (Wavelet) Time-Frequency Map



Bark-Band Noise Modeling at High Frequencies (Levine 1998)



Amplitude Envelope for One Noise Band



For more information, see Scott Levine's thesis.[2]

[2] http://ccrma.stanford.edu/~scottl/thesis.html

Sines + Noise + Transients Sound Examples

Scott Levine Thesis Demos (Sines + Noise + Transients at 32 kbps)
(`http://ccrma.stanford.edu/~scottl/thesis.html`)

Mozart's Le Nozze di Figaro

- Original
- Compressed using MPEG-AAC at 32 kbps
- Compressed using sines+transients+noise at 32 kbps

- Multiresolution sinusoids alone
- Residual Bark-band noise
- Transform-coded transients (AAC)
- Bark-band noise above 5 kHz

Rock Example

Scott Levine Thesis Demos (Sines + Noise + Transients at 32 kbps) (`http://ccrma.stanford.edu/~scottl/thesis.html`)

"It Takes Two" by Rob Base & DJ E-Z Rock

- Original
- MPEG-AAC at 32 kbps
- Sines+transients+noise at 32 kbps

- Multiresolution sinusoids
- Residual Bark-band noise
- Transform-coded transients (AAC)
- Bark-band noise above 5 kHz

Time Scale Modification using Sines + Noise + Transients

Scott Levine Thesis Demos (Sines + Noise + Transients at 32 kbps)
(`http://ccrma.stanford.edu/~scottl/thesis.html`)

Time-Scale Modification (pitch unchanged)

- S+N+T time-scale factors [2.0, 1.6, 1.2, 1.0, 0.8, 0.6, 0.5]

S+N+T Pitch Shifting (timing unchanged)

- Pitch-scale factors [0.89, 0.94, 1.00, 1.06, 1.12]

Spectral Modeling History Highlights

- Fourier's theory (1822)
- Teleharmonium (1906)
- Hammond organ (1930s)
- Channel Vocoder (1939)
- Phase Vocoder (1966)
- "Additive Synthesis" (1969)
- FFT Phase Vocoder (1976)
- Sinusoidal Modeling (1977,1979,1985)
- Sines+Noise (1989)
- Sines+Transients (1989)
- Sines+Noise+Transients (1998)

Perceptual audio coding:

- Princen-Bradley filterbank (1986)
- K. Brandenburg thesis (1989)
- *Auditory masking* usage
- Dolby AC2
- Musicam
- ASPEC
- MPEG-I,II,IV (incl. S+N+T "parametric sounds")

Future Prospects

Observations:

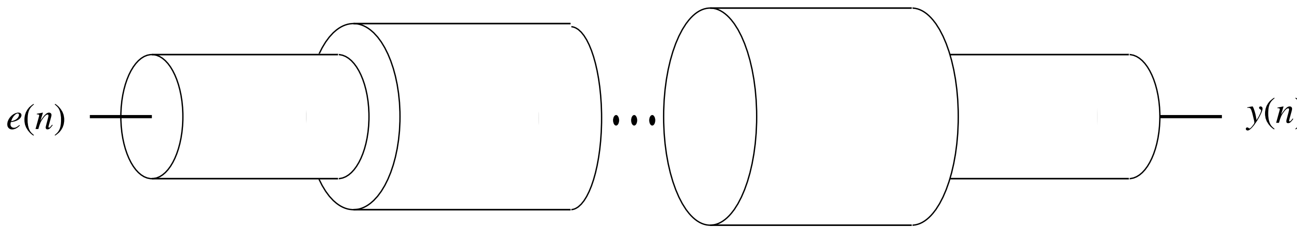
- Sinusoidal modeling of sound is “Unreasonably Effective”
- Basic “auditory masking” discards ≈ 90% information
- Interesting neuroscience observation:

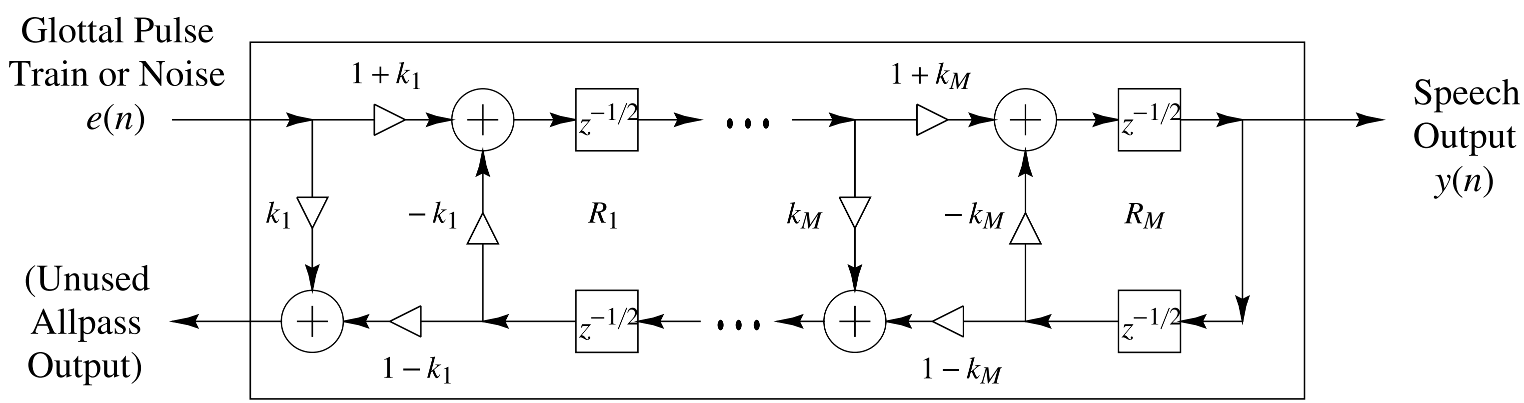
  > “... most neurons in the primary auditory cortex A1 are silent most of the time ...”

  (from “Sparse Time-Frequency Representations”, Gardner and Magnesco, PNAS:103(16), April 2006)
- What is a true and correct “psychospectral model” for sound?
  - The cochlea of the ear is a real-time spectrum analyzer
  - How is the “ear’s spectrogram” represented at higher levels?

Physical Modeling Synthesis (Historical Summary)

Kelly-Lochbaum Vocal Tract Model



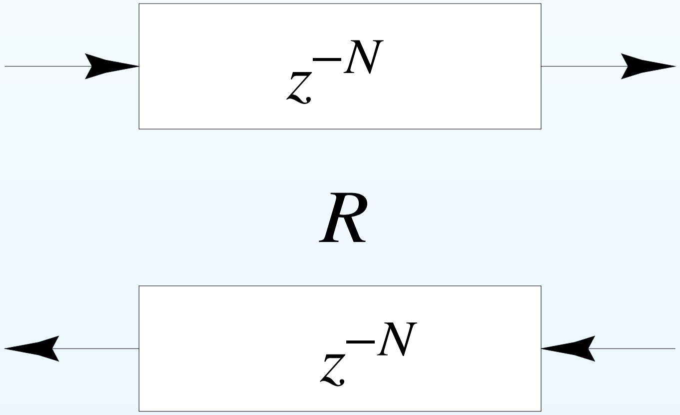


Kelly-Lochbaum Vocal Tract Model (Piecewise Cylindrical)

John L. Kelly and Carol Lochbaum (1962)

Digital Waveguide Models (1985)

Lossless digital waveguide $\stackrel{\Delta}{=}$ *bidirectional delay line* at some *wave impedance* R



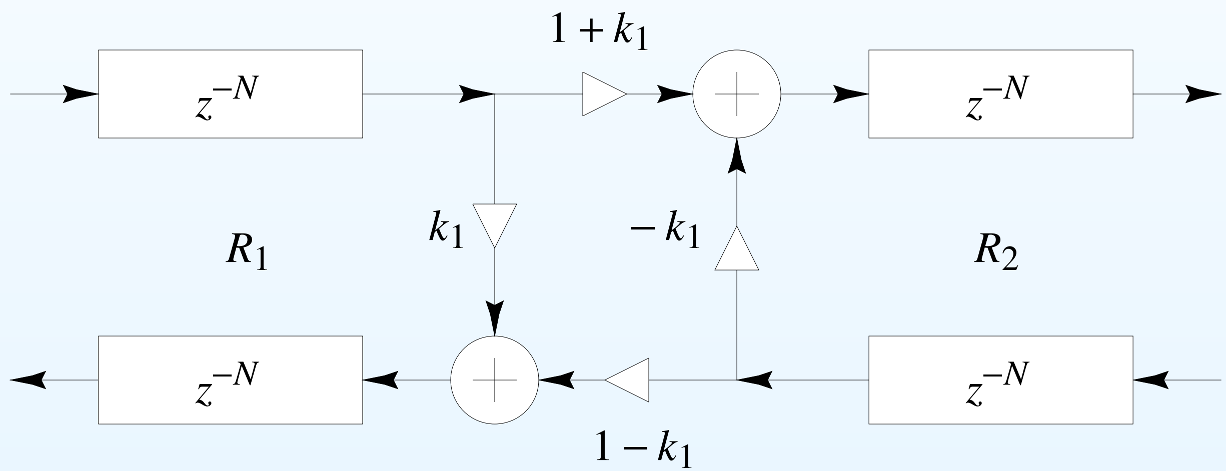
Useful for efficient models of

- strings
- bores
- plane waves
- conical waves

Signal Scattering

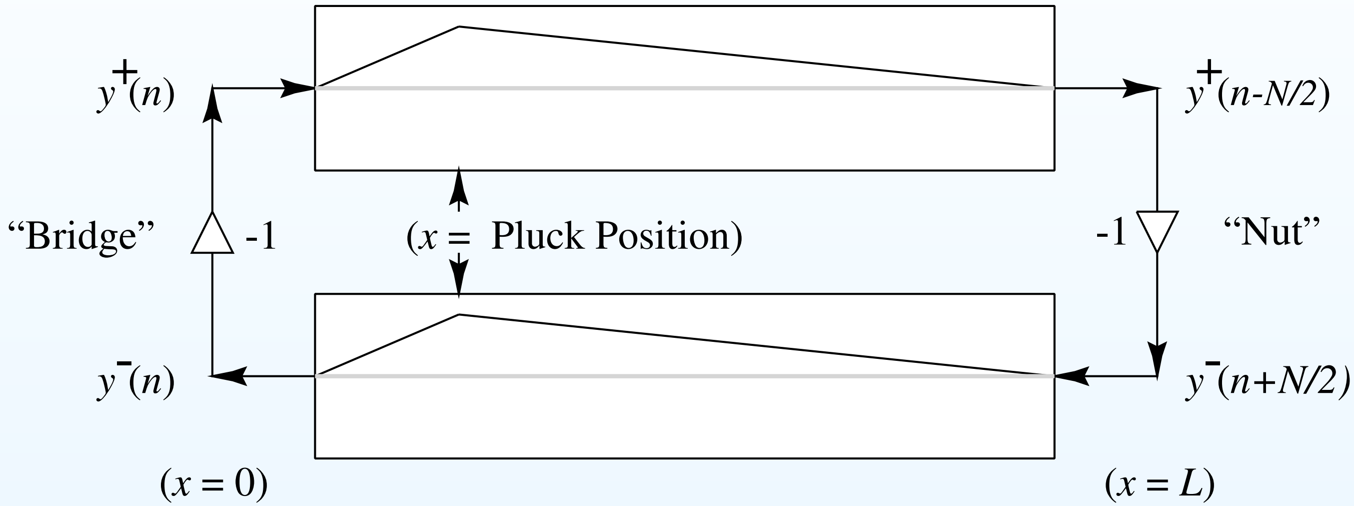
Signal *scattering* is caused by a *change* in wave impedance R:

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



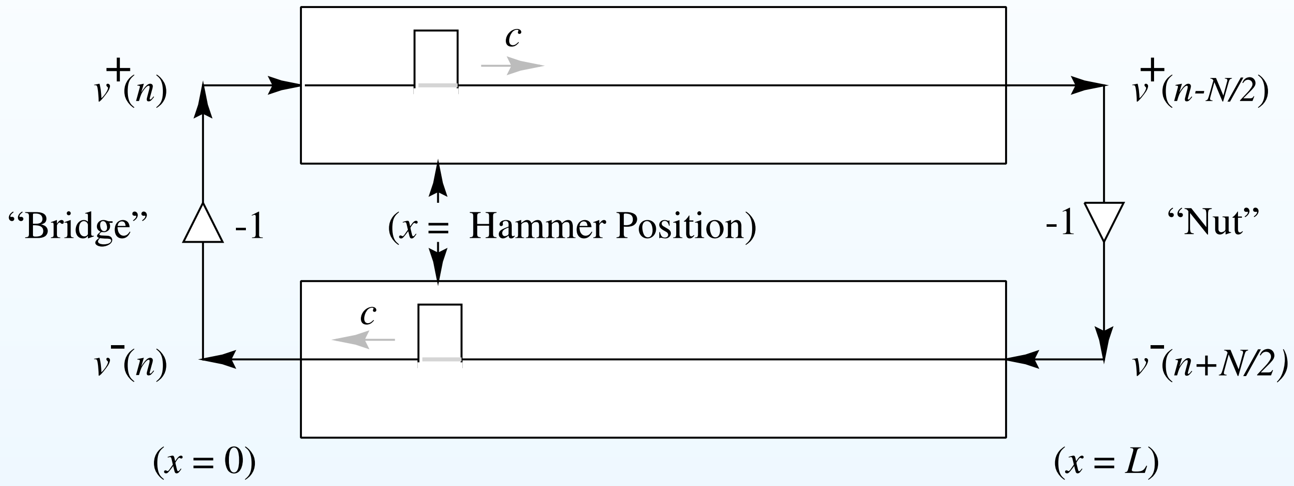
If the wave impedance changes *every spatial sample*, the Kelly-Lochbaum vocal-tract model results.

Ideal Plucked String (Displacement Waves)



- Load each delay line with *half* of initial string displacement
- Sum of upper and lower delay lines = string displacement

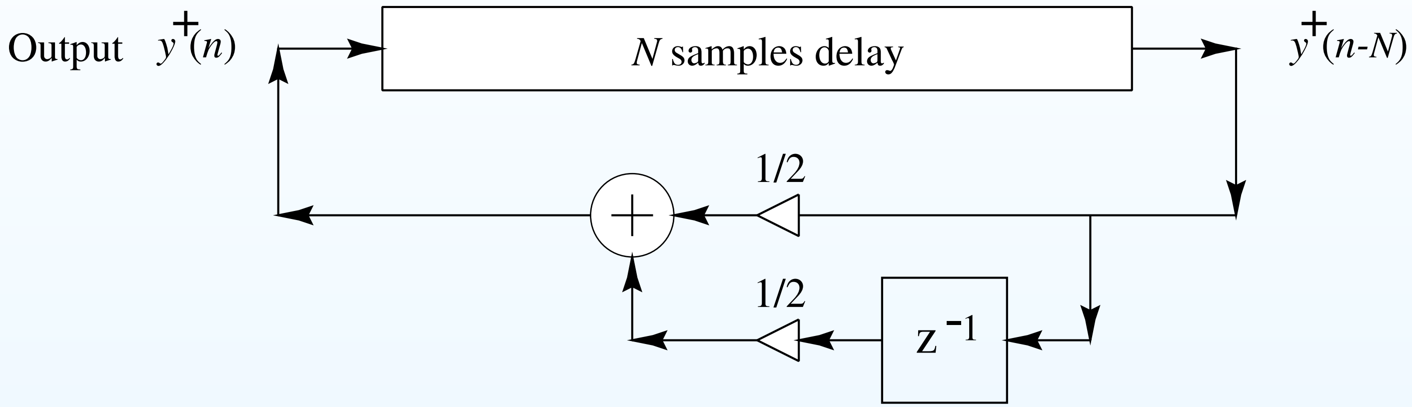
Ideal Struck String (Velocity Waves)



Hammer strike = *momentum transfer* = velocity step:

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

Karplus-Strong (KS) Algorithm (1983)

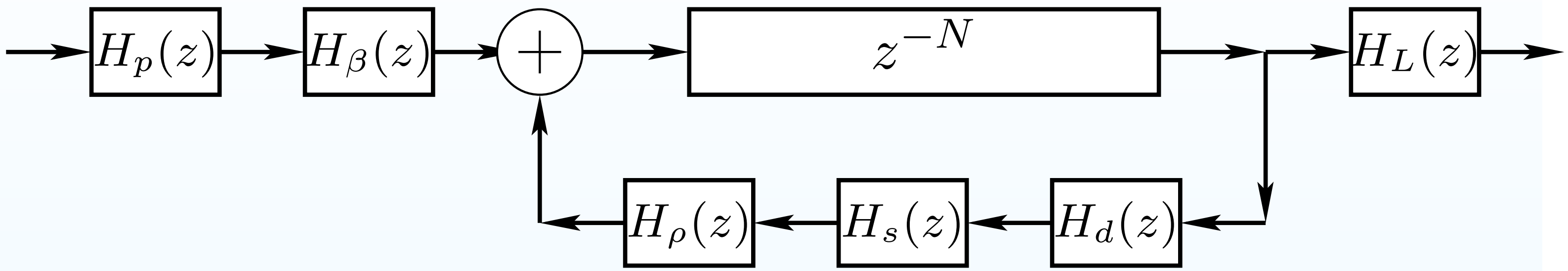


- Discovered (1978) as “self-modifying wavetable synthesis”
- Wavetable is preferably initialized with random numbers

Karplus-Strong Sound Examples

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

EKS Algorithm (Jaffe-Smith 1983)



$$N = \text{pitch period } (2\times \text{ string length) in samples}$$

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

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

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

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

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

$$H_L(z) = \frac{1-R_L}{1-R_L\,z^{-1}} = \text{dynamic-level lowpass filter}$$

STK EKS Sound Examples

- Synthesis Tool Kit (STK) by Perry Cook, Gary Scavone, and others — distributed by CCRMA:
  `Google` search: *STK ToolKit*

STK Plucked String: (WAV) (MP3)

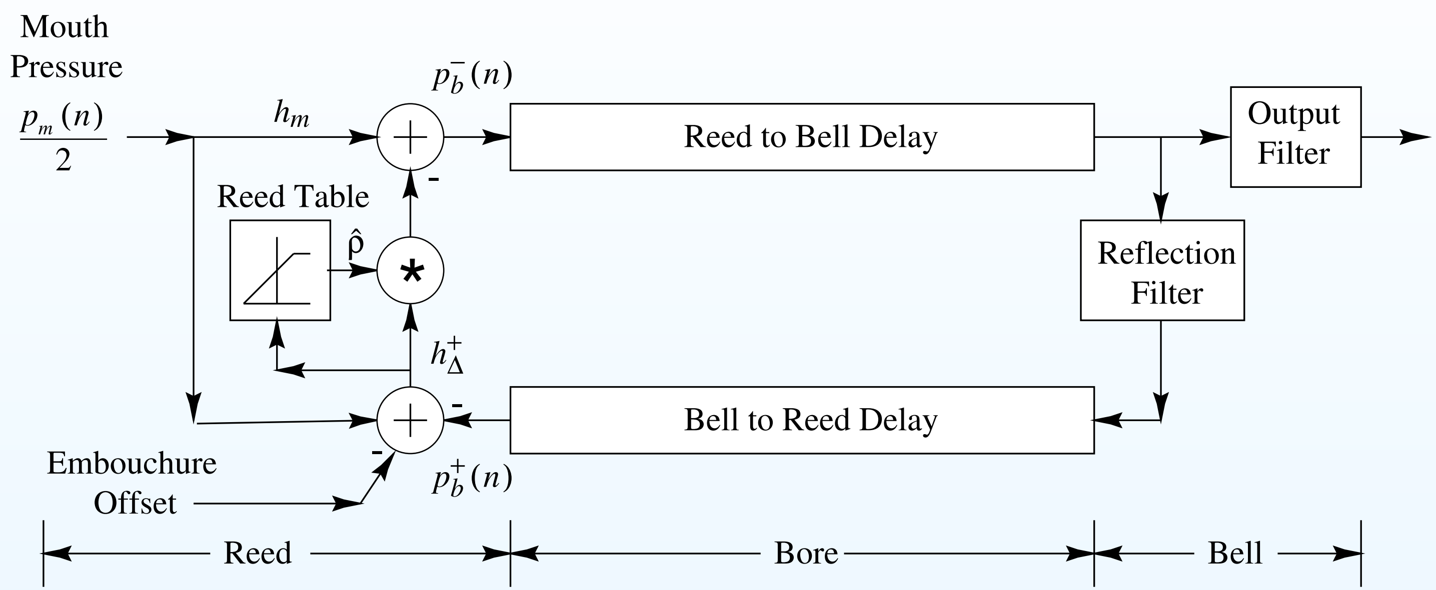
- Plucked String 1: (WAV) (MP3)
- Plucked String 2: (WAV) (MP3)
- Plucked String 3: (WAV) (MP3)

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

Digital Waveguide Single Reed, Cylindrical Bore Model (1986)



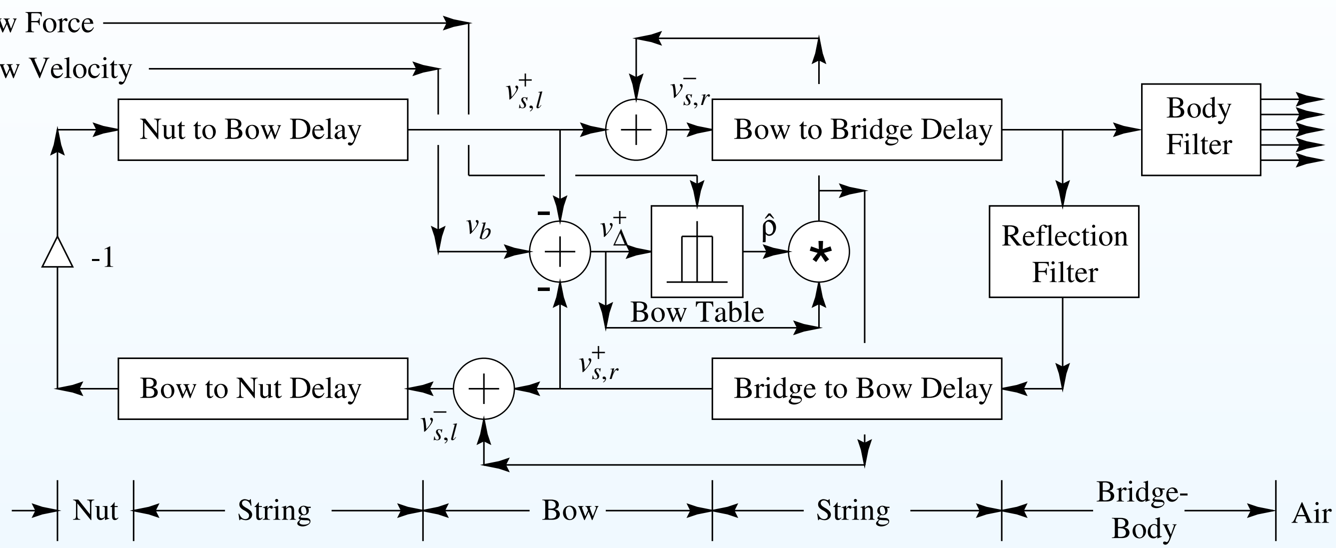
Digital waveguide clarinet

- 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*
  - Synthesis Tool Kit (STK) by Perry Cook, Gary Scavone, and others — distributed by CCRMA:
    `Google` search: *STK ToolKit*
- 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)

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)

“Electric Cello” Sound Examples (Peder Larson)

- Staccato Notes: (WAV) (MP3)
  (short strokes of high bow pressure, as from a bouncing bow)
- Bach’s First Suite for Unaccompanied Cello: (WAV) (MP3)

Overview

Early Digital Synthesis

Spectral Modeling

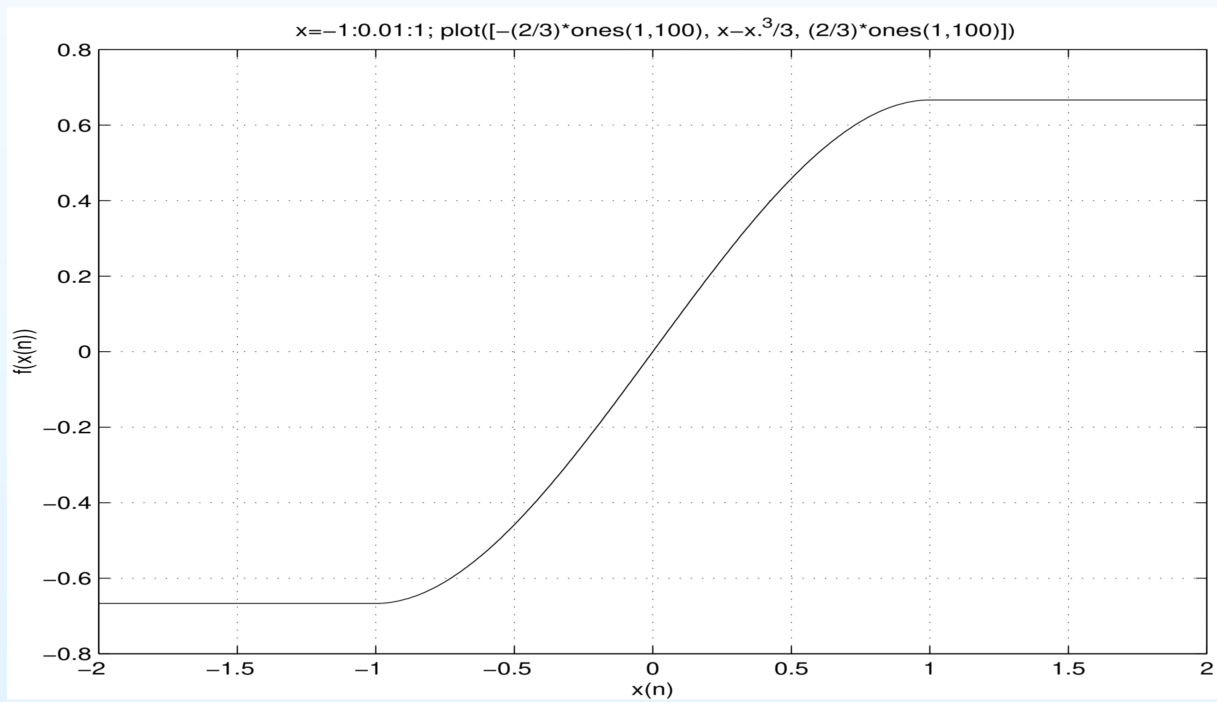
Physical Modeling

- KL Music
- Digital Waveguide
- Signal Scattering
- Plucked String
- Struck String
- Karplus Strong
- EKS Algorithm
- Clarinet
- Wind Examples
- Bowed Strings
- Distortion Guitar
- Acoustic Strings
- Sound Examples
- Linearized Violin
- Commuted Piano
- Pulse Synthesis
- Complete Piano
- Sound Examples
- Phy Audio Coding
- Phy Audio Coding?

Summary

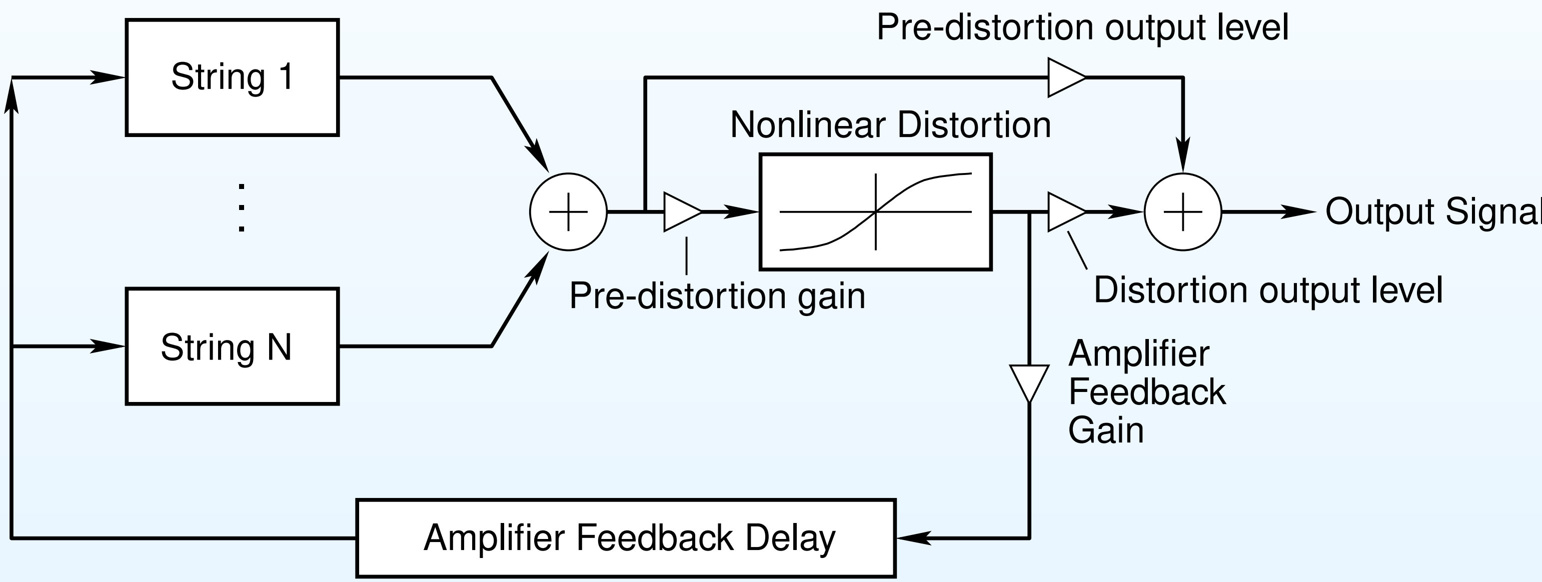
Soft Clipper

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



Amplifier Distortion + Amplifier Feedback

Sullivan 1990



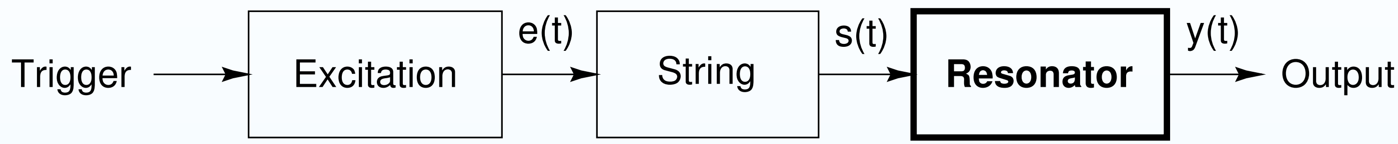
Distortion output signal often further filtered by an *amplifier cabinet filter*, representing speaker cabinet, driver responses, etc.

Distortion Guitar Sound Examples

(Stanford Sondius Project, ca. 1995)

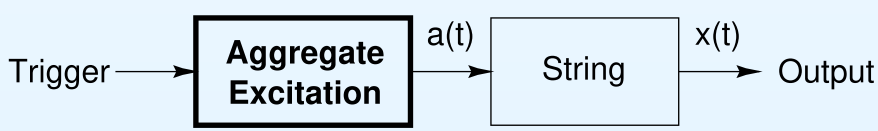
- Distortion Guitar: (WAV) (MP3)
- Amplifier Feedback 1: (WAV) (MP3)
- Amplifier Feedback 2: (WAV) (MP3)

Commuted Synthesis of Acoustic Strings (1993)



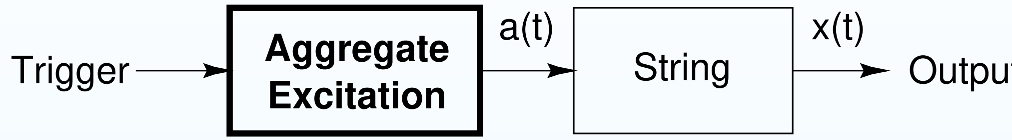
Schematic diagram of a stringed musical instrument.

Equivalent diagram in the linear, time-invariant case.

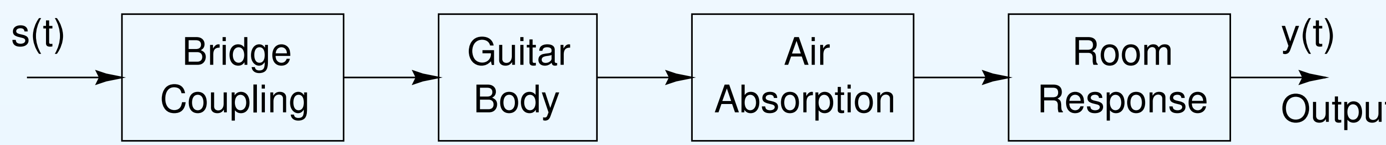


Use of an aggregate excitation given by the convolution of original excitation with the resonator impulse response.

Commuted Components



"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

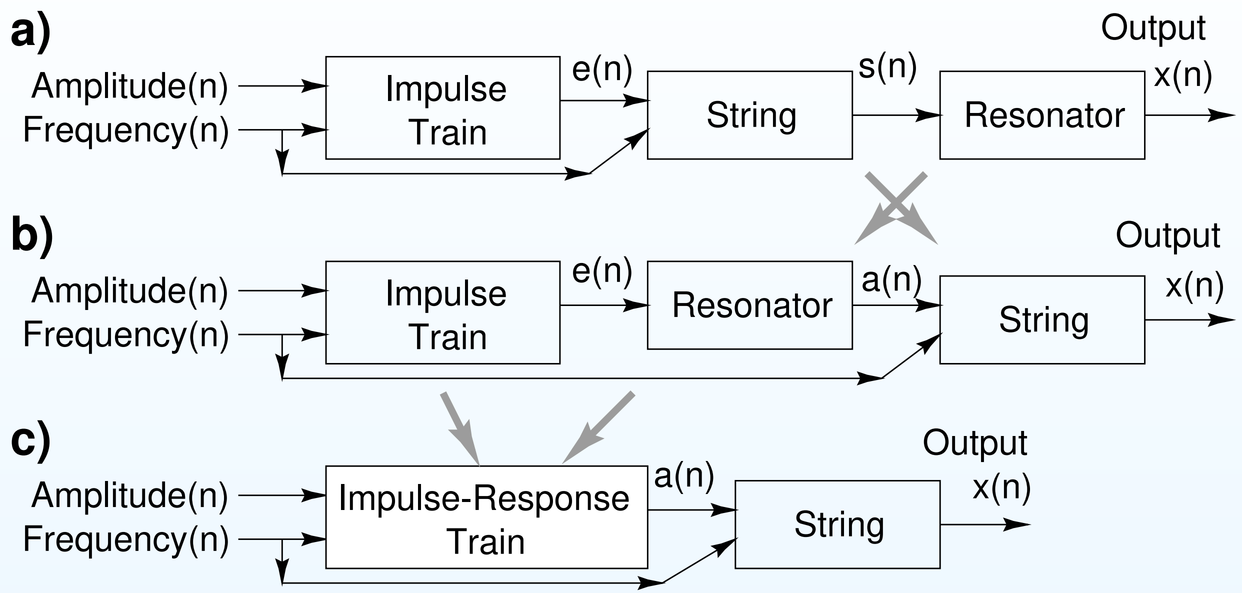
- Bach Prelude in E Major: (WAV) (MP3)
- soundexamplewavBach silenceLoure in E Major: (WAV) (MP3)

Virtual performance by Dr. Mikael Laurson, Sibelius Institute

Virtual guitar by Helsinki Univ. of Tech., Acoustics Lab[3]

[3] `http://www.acoustics.hut.fi/`

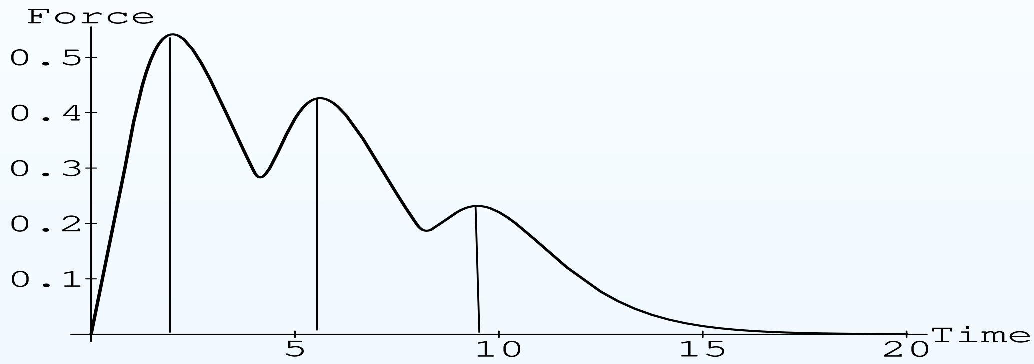
Commuted Synthesis of Linearized Violin



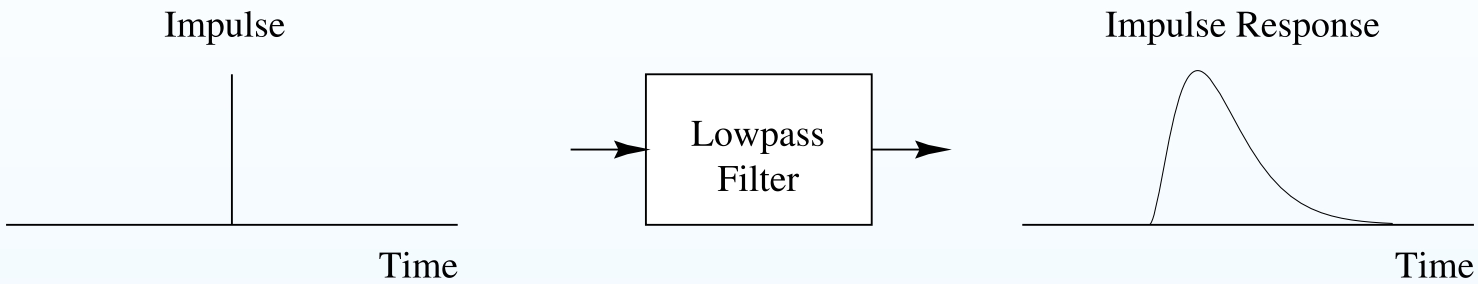
- Assumes *ideal Helmholtz motion* of string
- Sound Examples (Stanford Sondius project, ca. 1995):
  - Bass: (WAV) (MP3)
  - Cello: (WAV) (MP3)
  - Viola 1: (WAV) (MP3)
  - Viola 2: (WAV) (MP3)
  - Violin 1: (WAV) (MP3)
  - Violin 2: (WAV) (MP3)
  - Ensemble: (WAV) (MP3)

Commuted Piano Synthesis (1995)

Hammer-string interaction pulses (force):



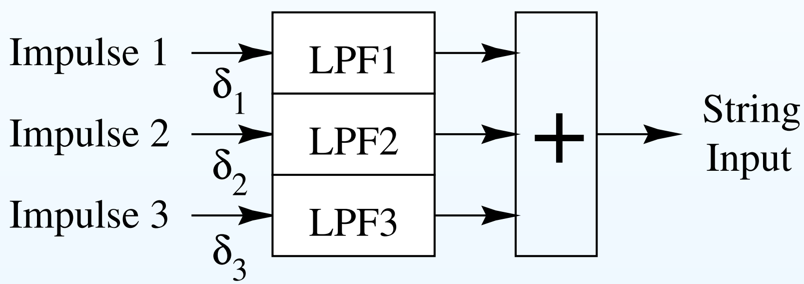
Synthesis of Hammer-String Interaction Pulse

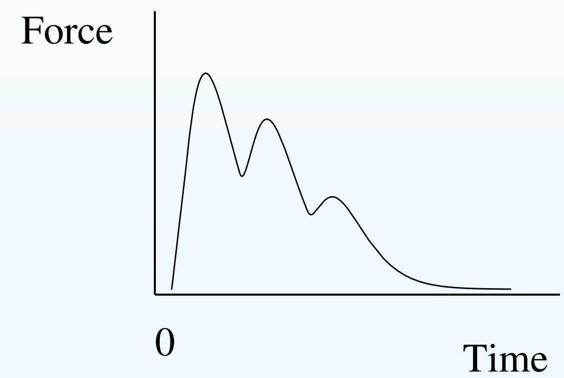


- Faster collisions correspond to *narrower* pulses (*nonlinear filter*)
- For a *given velocity*, filter is linear time-invariant
- Piano is “linearized” for each hammer velocity

Multiple Hammer-String Interaction Pulses

Superimpose several individual pulses:





Overview

Early Digital Synthesis

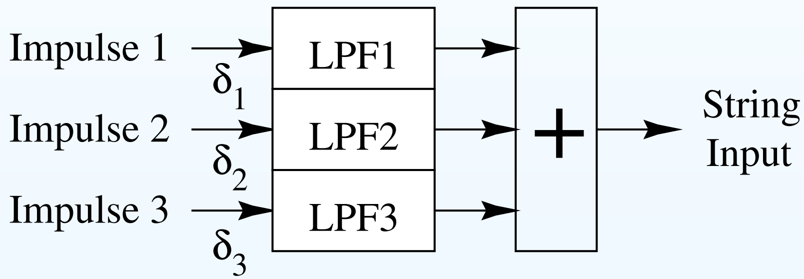
Spectral Modeling

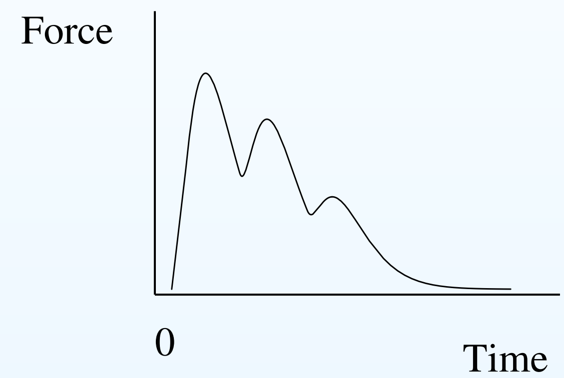
Physical Modeling
- KL Music
- Digital Waveguide
- Signal Scattering
- Plucked String
- Struck String
- Karplus Strong
- EKS Algorithm
- Clarinet
- Wind Examples
- Bowed Strings
- Distortion Guitar
- Acoustic Strings
- Sound Examples
- Linearized Violin
- Commuted Piano
- Pulse Synthesis
- Complete Piano
- Sound Examples
- Phy Audio Coding
- Phy Audio Coding?

Summary

Multiple Hammer-String Interaction Pulses

Superimpose several individual pulses:

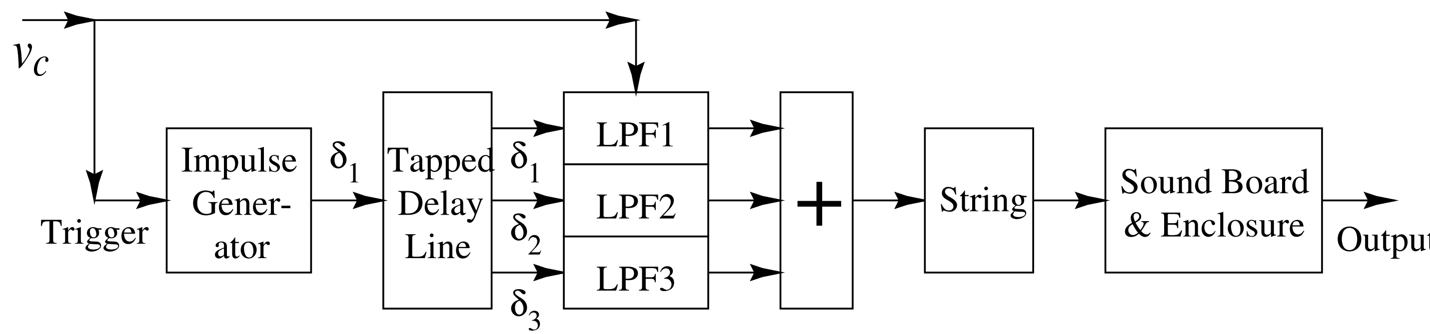




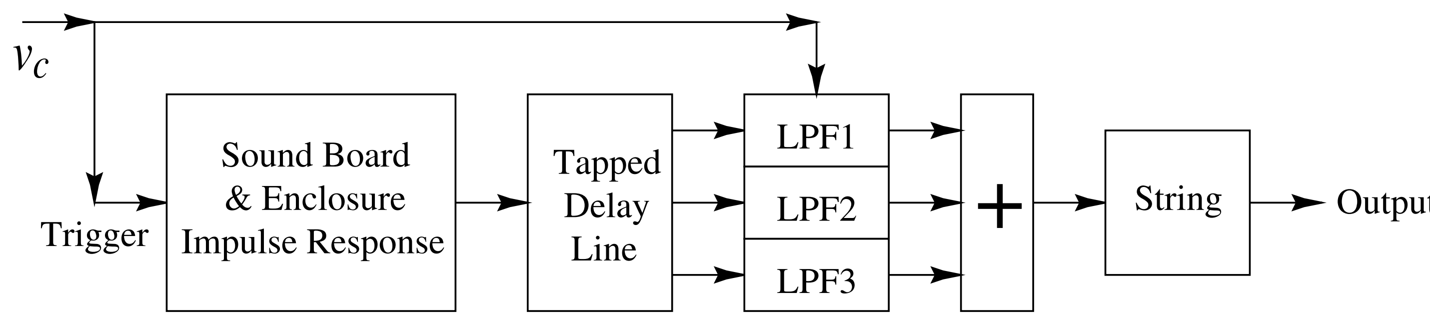
As impulse amplitude grows (faster hammer strike), output pulses become *taller and thinner*, showing less overlap.

Complete Piano Model

Natural Ordering:



Commuted Ordering:



- 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)

Reference:

"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

`Google` search: *Harpsichord Sound Synthesis*

Physical Modeling in Audio Coding

Spectral modeling synthesis is finding application in audio coding. Can physical modeling synthesis be used as well?

- MPEG-4/SAOL already supports essentially all sound synthesis methods
- Ability to *encode* sounds *automatically* is limited
  - Codebook-Excited Linear Prediction (CELP) is a successful *source-filter* model (not quite physical)
  - There are many isolated examples of model-fitting to recorded data
  - Good *model-based denoising* results have been obtained
  - Coder problem much harder when many sources are mixed

Best Known Model-Based Audio Coders

A “cover band” can put together a very convincing facsimile of popular music performance

JOS high-school band “Bittersweet”

Future Physical Modeling in Audio Coding?

A "*Cover Band*" Approach to Model-Based Audio Coding:

1. Recognize individual "audio streams" in a mix (CASA)
   ("*I hear a trap set, electric bass, Fender Rhodes, and a strat*")
2. For each stream, calibrate its model heuristically
   ("*Here is what I hear the bass part doing: ...*")
3. Fine-tune the synthetic mix to the real mix
   (joint "maximum likelihood estimation")

Features of "Cover-Band Coding" (CBC):

- The "playing experience" of each "virtual performer" prevents artifacts — "musically unreasonable" parameters are made unlikely ("Bayesian priors")
- An incorrect instrument must "imitate" its assigned stream
- New arrangements can be synthesized by deliberately choosing a new ensemble!

Overview
Early Digital Synthesis
Spectral Modeling
Physical Modeling
Summary

Summary

Summary

We have reviewed a “CCRMA-centric slice” of the history of digital sound synthesis (usually starting with results from Bell Labs):

- Wavetable (one period)
- Subtractive
- Additive
- FM
- Sampling
- Spectral Modeling
- Physical Modeling (more in tomorrow’s 4:30 PM masterclass)
- Connections to audio coding

Sound Acknowledgment

Sound Acknowledgment

Thanks to Emu / Creative Labs for providing a superb-quality *external D/A converter* for this talk (an E-Mu 0404/USB)