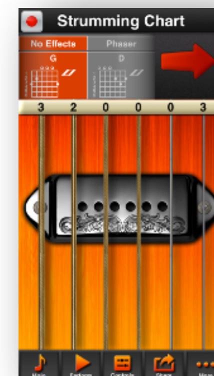
A Brief History of Physical Modeling Synthesis, Leading up to Mobile Devices and MPE



Pat Scandalis
Dr. Julius O. Smith III
Nick Porcaro

Berklee Voltage, March 10-11 2017

02/20/2017

Overview

The story of physical modeling stretches back nearly 1000 years (yup)! We now find ourselves in a place where each of us can be Jimi Hendrix with just a small device in the palm of our hands. Its a fun and deeply technical topic drawing on many fields including physics, acoustics, digital signal processing and music.

- Demo
- A few high points from the history
- Questions, possibly from the FAQ

The Full Presentation Can Be Found at:

moforte.com/berklee-voltage-physical-modeling/

It's also posted in the news section

of the moForte website

Physical Modeling Was Poised to be the "Next Big Thing" in 1994

So What Happened?

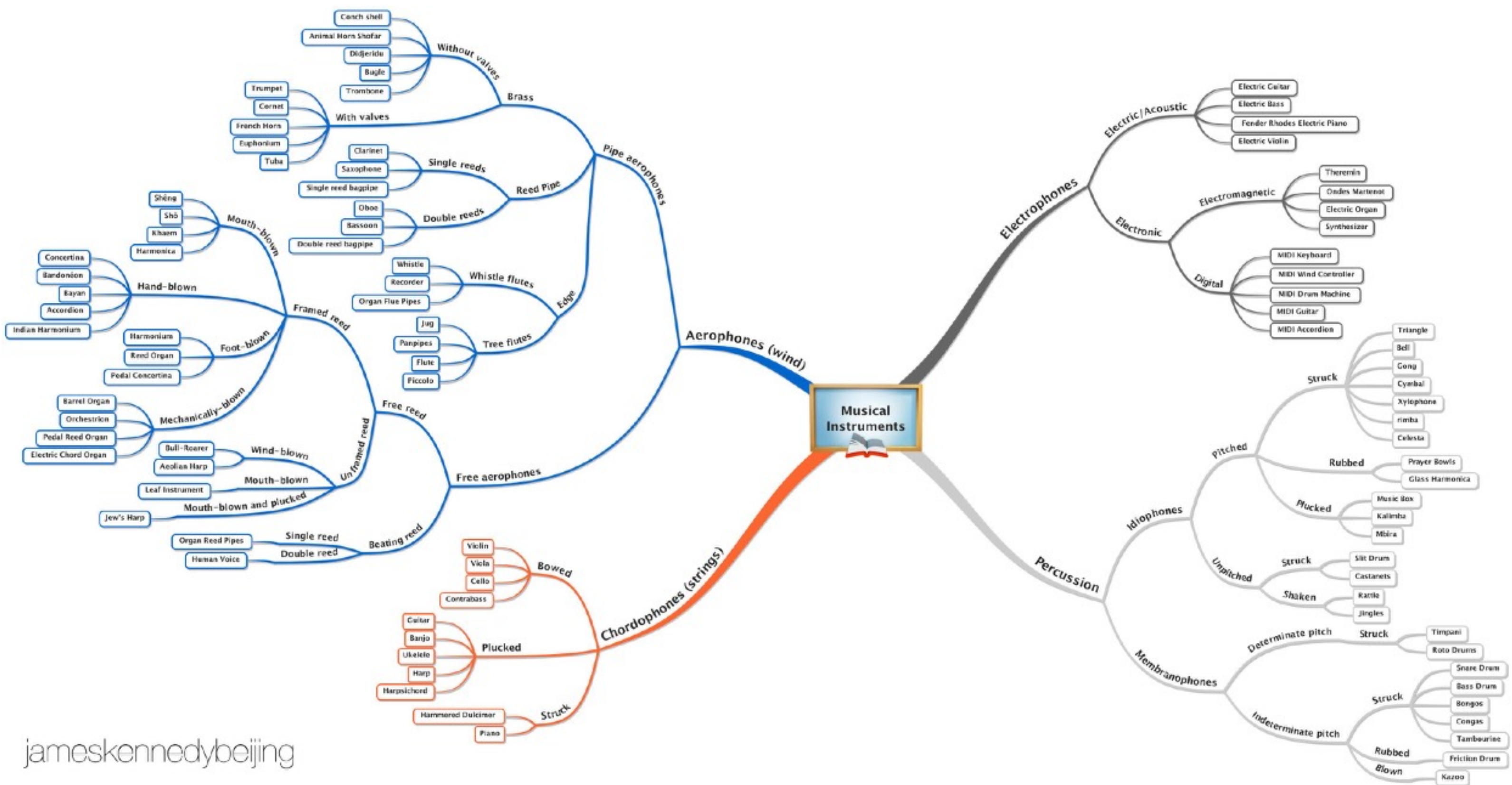
For Context, what is Physical Modeling Synthesis?

- Methods in which a sound is generated using a mathematical model of the physical source of sound.
- Any gestures that are used to interact with a real physical system can be mapped to parameters yielded an interactive an expressive performance experience.
- **Physical modeling is a collection of different techniques.**

$$\frac{\partial^2 y}{\partial t^2} = \frac{1}{v_w^2} \frac{\partial^2 y}{dt^2}$$

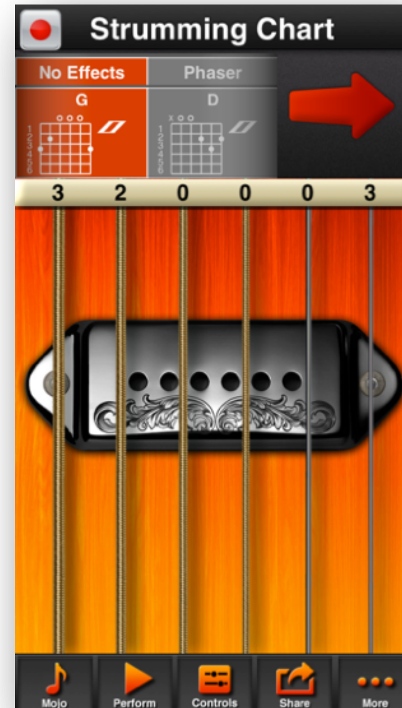
Taxonomy of Modeling Areas

Hornbostel–Sachs Classification



- Chordaphones - Guitars
- Aerophones - Woodwinds
- Membranophones - Drums
- Idiophones - Mallet Instruments
- Electrophones - Virtual Analog
- Game Sounds
- Voice

First a Quick Demo!



[Geo Shred Preview]

and

[Europa Demo]

[Modeled Guitar Features]

and

[Demo Reel]

Brief (though not complete) History of Physical Modeling Synthesis

As well as a some commercial products using the technology

Early Mechanical Voice Synthesis

- 1000 -1200 ce - Speech Machines, Brazen Heads
- 1791 - Wolfgang Von Kempelin, speaking machine.
- 1857 - Joseph Faber, Euphonia (pictured)

Its been know for a long time that the vocal tract can be modeled with a bellows, a reed, a number of different size resonators and special elements for the tongue, the mouth. See Exploratorium Vocal Vowels.

The Voder (1937-39) - Homer Dudley

- Analog Electronic Speech Synthesis
- Analog model of the vocal tract
- Develop from research on voice compression at Bell Labs.
- Featured at the 1939 Worlds fair
- [YouTube]

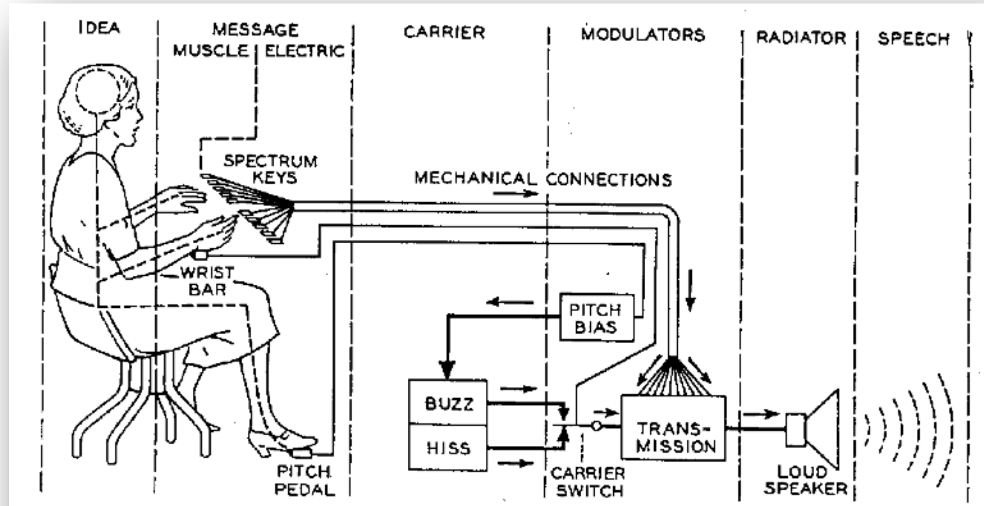
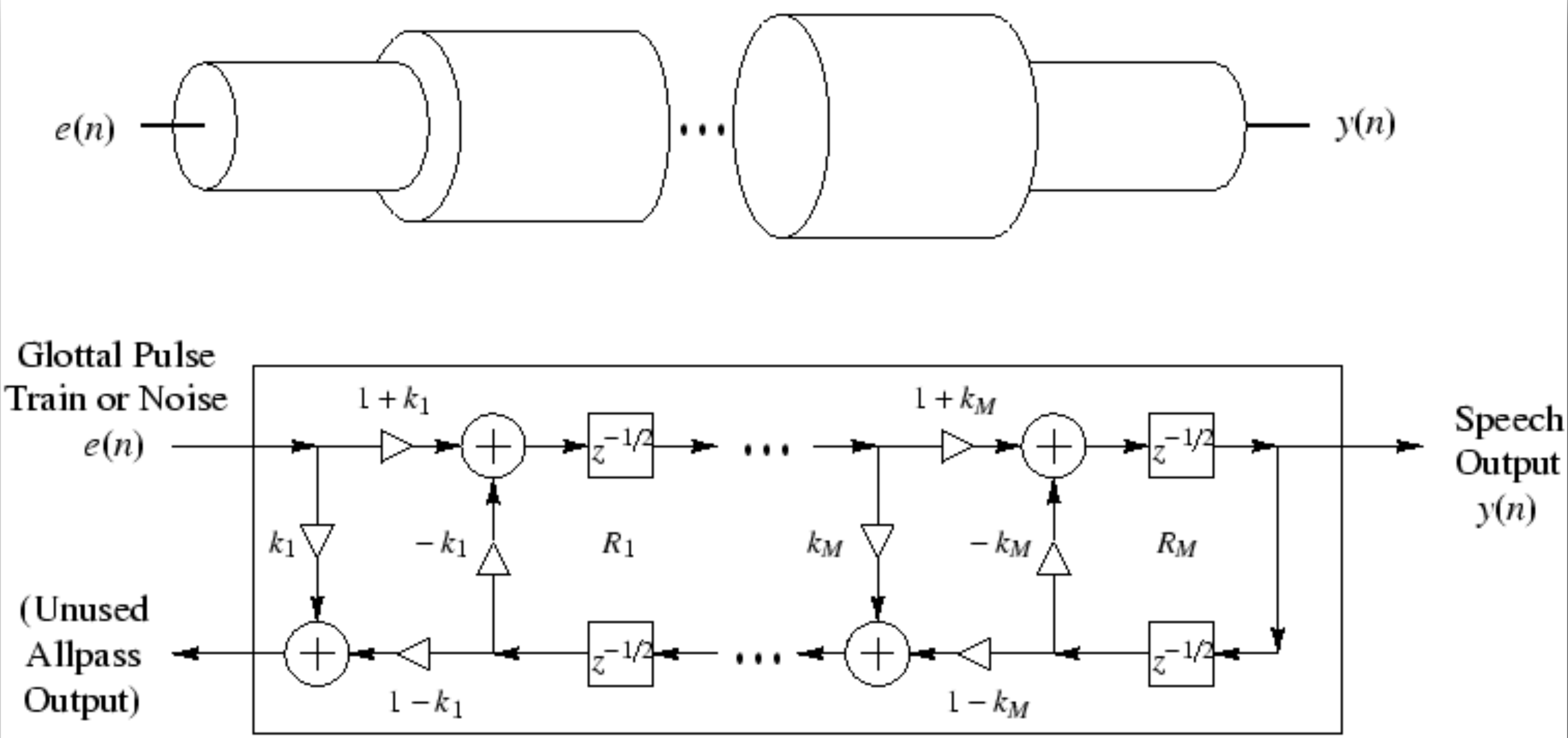


Fig. 8—Schematic circuit of the voder.

Kelly-Lochbaum Vocal Tract Model (1961)

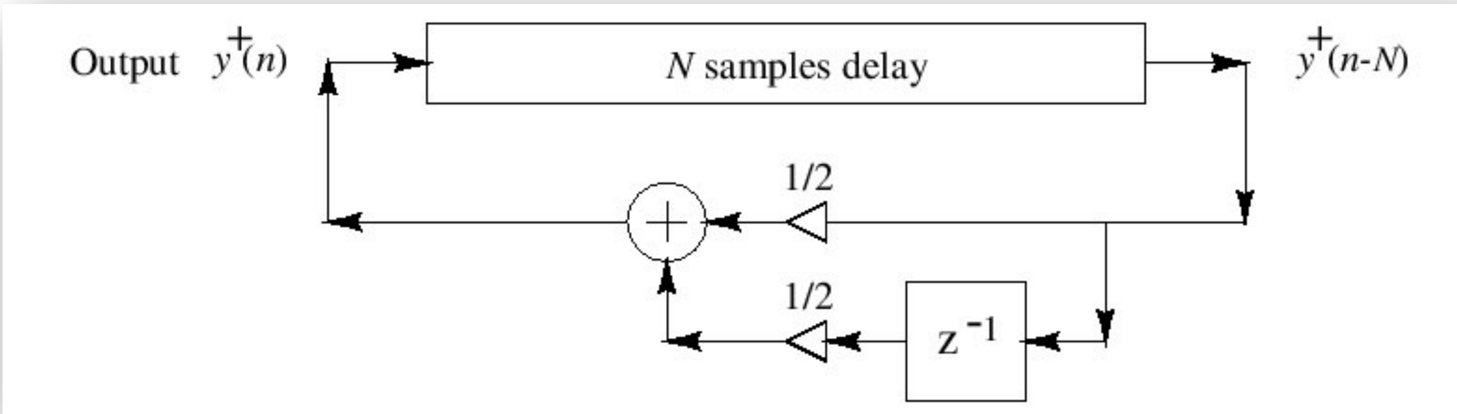


Kelly-Lochbaum Vocal Tract Model (Piecewise Cylindrical)

Daisy Bell (1961)

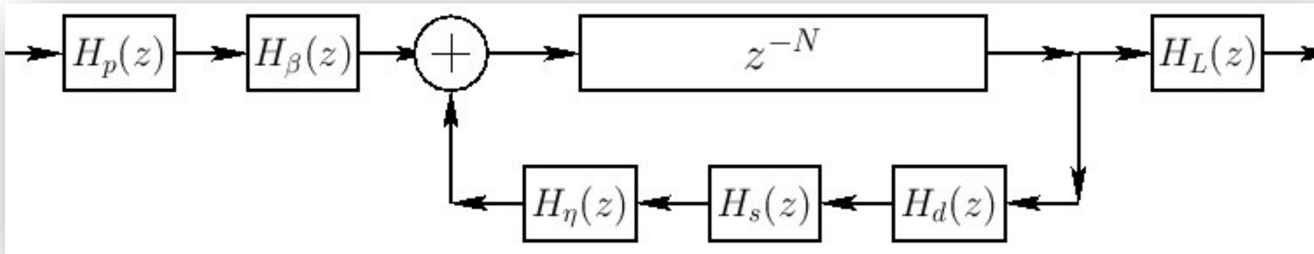
- Daisy Bell (MP3)
- Vocal part by Kelly and Lochbaum (1961)
- Musical accompaniment by Max Mathews
- Computed on an IBM 704
- Based on Russian speech-vowel data from Gunnar Fant's book
- Probably the first digital physical-modeling synthesis sound example by any method
- Inspired Arthur C. Clarke to adapt it for "2001: A Space Odyssey" the Hal 9000's "first song"

Karplus-Strong (KS) Algorithm (1983)



- Discovered (1978) as "self-modifying wavetable synthesis"
- Wavetable is preferably initialized with random numbers
- Licensed to Mattel
- The first musical use of the algorithm was in the work "*May All Your Children Be Acrobats*" written in 1981 by David A. Jaffe. (MP3)

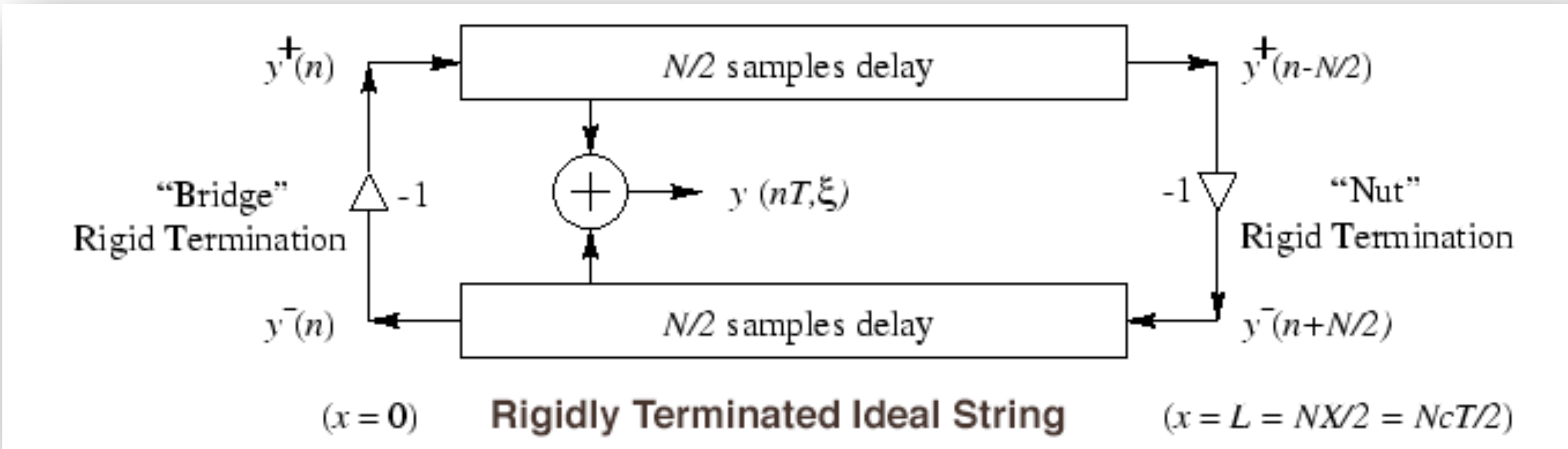
EKS Algorithm (Jaffe-Smith 1983)



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

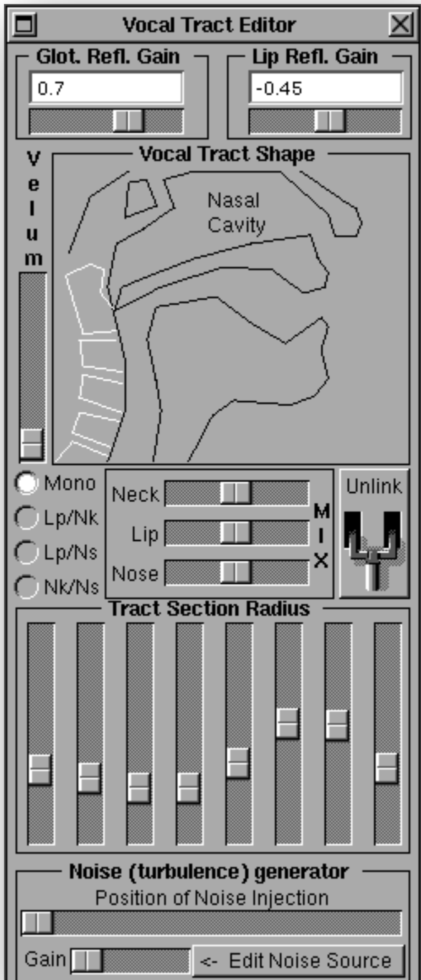
- Musical Example “Silicon Valley Breakdown” (Jaffe 1992) (MP3)
- Musical Example BWV-1041 (used to intro the NeXT machine 1988) YouTube

Digital Waveguide Models (Smith 1985)



- Equivalent to d'Alembert's Solution to the Partial Differential Equation for a string (1747)
- Useful for efficient models of
  - Strings
  - Bores
  - plane waves
  - conical waves

Sheila Vocal Track Modeling (Cook 1990)



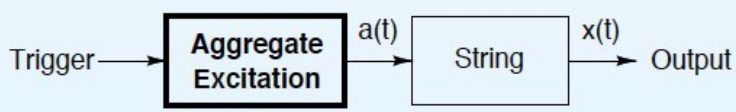
Perry Cook's SPASM **"Singing Physical Articulatory Synthesis Model"**

- Diphones: (MP3)
- Nasals: (MP3)
- Scales: (MP3)
- "Sheila": (MP3)

Commuted Synthesis (Smith) (1994)

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 Synthesis Examples

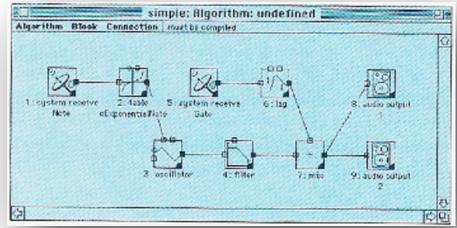
- Electric guitar, different pickups and bodies (Sondius) (MP3)
- Mandolin (STK) (MP3)
- Classical Guitar (Mikael Laurson, Cumhur Erkut, and Vesa Välimäki) (MP3)
- Bass (Sondius) (MP3)
- Upright Bass (Sondius) (MP3)
- Cello (Sondius) (MP3)
- Piano (Sondius) (MP3)
- Harpsichord (Sondius) (MP3)

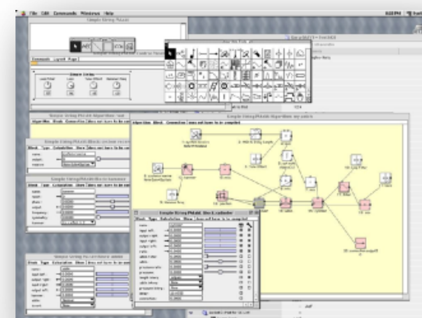
Yamaha VL Line (1994)

- Yamaha Licensed “Digital Waveguide Synthesis” for use in its products including the VL line (VL-1, VL-1m, VL-70m, EX-5, EX-7, chip sets, sound cards, soft-synth drivers)
- Shakuhachi: (MP3)
- Oboe and Bassoon: (MP3)
- Tenor Saxophone: (MP3)

Korg SynthKit Line (1994)

- SynthKit (1994)
- Prophecy (1995)
- Trinity (1995)
- OASYS PCI (1999)
- OASYS (2005)
- Kronos (2011)





02/20/2017

"The Next Big Thing" (1994)

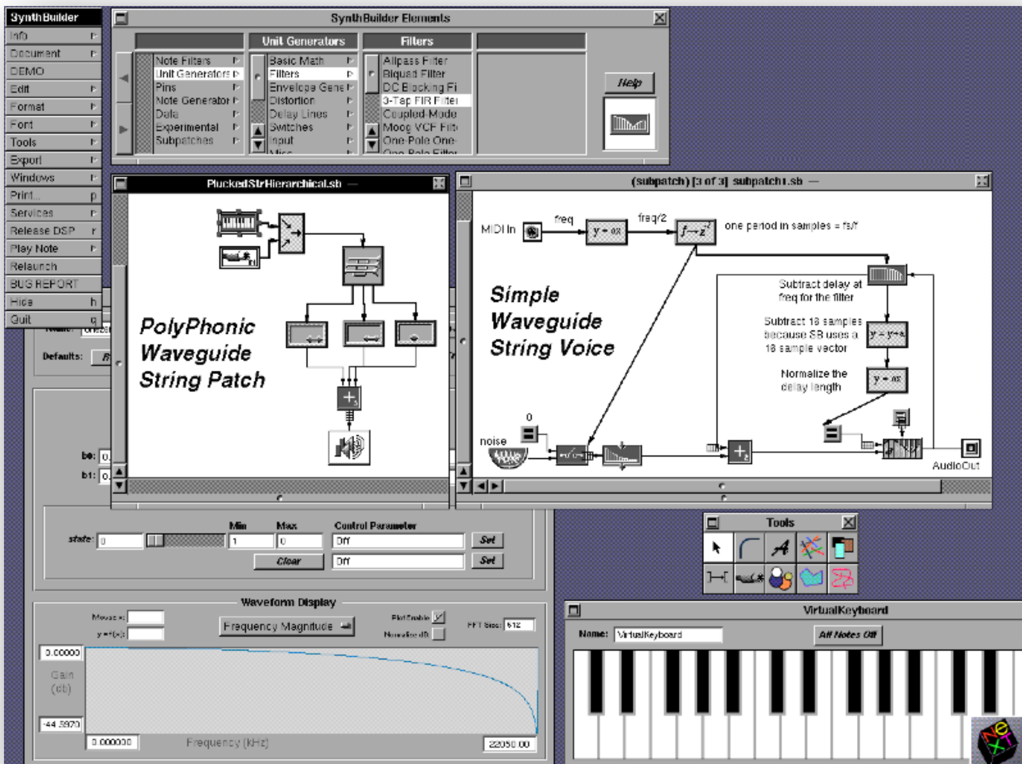
The Next Big Thing 2/94

The History of PM 9/94

Stanford Sondius Project (1994-1997)

- Stanford OTL/CCRMA created the Sondius project to assist with commercializing physical modeling technologies.
- The result was a modeling tool known as SynthBuilder, and a set of models covering about two thirds of the General MIDI set.
- Many modeling techniques were used including EKS, Waveguide, Commuted Synthesis, Coupled Mode Synthesis, Virtual Analog.

SynthBuilder (Porcaro, et al) (1995)



- SynthBuilder was a user-extensible, object-oriented, NEXTSTEP Music Kit application for interactive real-time design and performance of synthesizer patches, especially physical models.
- Patches were represented by networks consisting of digital signal processing elements called unit generators and MIDI event elements called note filters and note generators.

STK (1995)

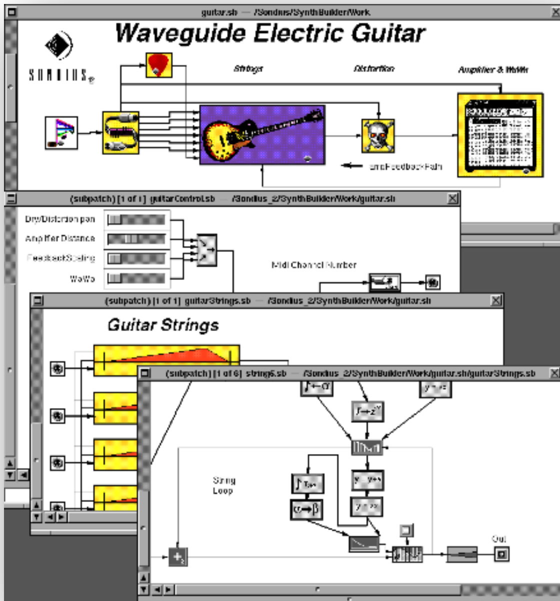
- Synthesis Tool Kit (STK) by Perry Cook, Gary Scavone, et al. distributed by CCRMA
- The **Synthesis Toolkit** (**STK**) is an open source API for real time audio synthesis with an emphasis on classes to facilitate the development of physical modeling synthesizers.
- Versions of the STK instrument classes have been integrated into ChucK, Csound, Real-Time Cmix, Max/ MSP (as part of PeRColate), SuperCollider and FAUST.
- Pluck example (MP3)
- STK Clarinet (MP3)

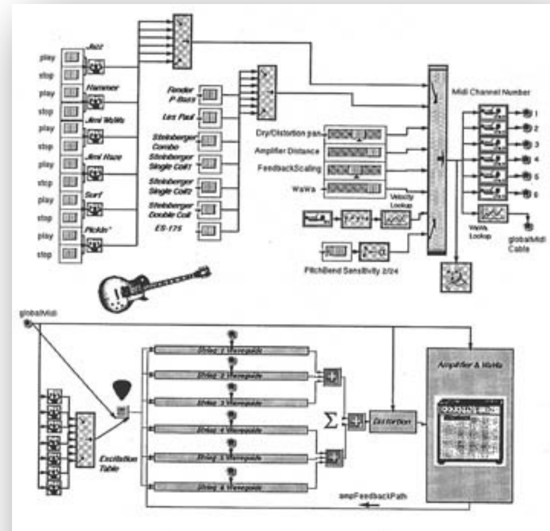
- Opcodes for a number of PM algorithms
  - Plucks,
  - Waveguide,
  - Woodwinds,
  - Brass,
  - Bowed
  - Bars
  - Flute

The Frankenstein Box (1996)

- The Frankenstein box was an 8 DSP 56k compute farm build by Bill Putnam and Tim Stilson
- There was also a single card version know as the “Cocktail Frank”
- Used for running models developed with SynthBuilder
- The distortion guitar ran on 6 DSPs with an additional 2 DSPs used for outboard effects.

The Sondius Electric Guitar (1996)





A six string electric guitar model with distortion, feedback and wawa, implemented in SynthBuilder. This model runs on on a single 72Mhz Motorola 56002 DSP.

- Pick model for different guitars/pickups (commuted synthesis, Scandalis)
- Feedback and distortion with amp distance (Sullivan)
- Wah-wah based on cry baby measurements (Putnam, Stilson)
- Reverb and flanger (Dattorro)
- Hybrid allpass delay line for pitchBend (Van Duyne, Jaffe, Scandalis)
- Performed using a 6-channel MIDI guitar controller.
- With no effects, 6 strings ran at 22k on a 72 Mhz Motorola 56002 DSP.
- Waveguide Guitar Distortion, Amplifier Feedback (MP3)

Sondius Sound Examples (1996)

- Waveguide Flute Model (MP3)
- Waveguide Guitar Model, Different Pickups (MP3)
- Waveguide Guitar Distortion, Amplifier Feedback (MP3)
- Waveguide Guitar Model, Wah-wah (MP3)
- Waveguide Guitar Model, Jazz Guitar (ES-175) (MP3)
- Harpsichord Model (MP3)
- Tibetan Bell Model (MP3)
- Wind Chime Model (MP3)
- Tubular Bells Model (MP3)
- Percussion Ensemble (MP3)
- Taiko Ensemble (MP3)
- Bass (MP3)
- Upright Bass (MP3)
- Cello (MP3)
- Piano (MP3)
- Harpsichord (MP3)
- Virtual Analog (MP3)

Coupled Mode Synthesis (CMS) (Van Duyne) (1996)

- Modeling of percussion sounds
- Modal technique with coupling
- Tibetan Bell Model (MP3)
- Wind Chime Model (MP3)
- Tubular Bells Model (MP3)
- Percussion Ensemble (MP3)

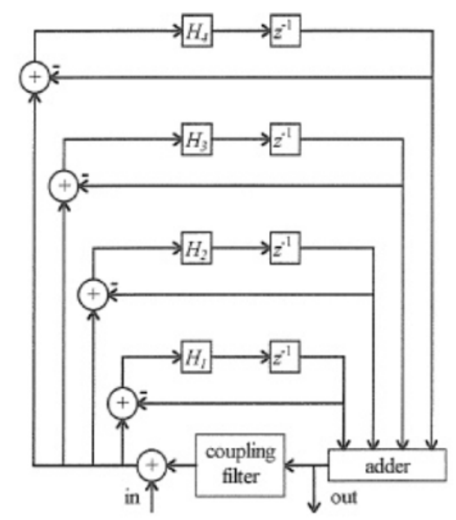
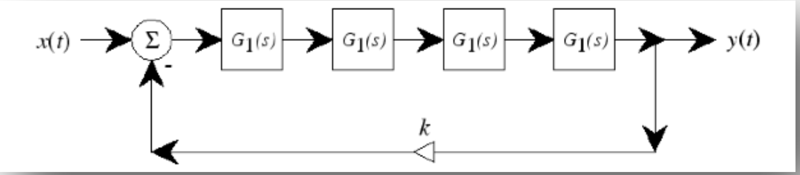
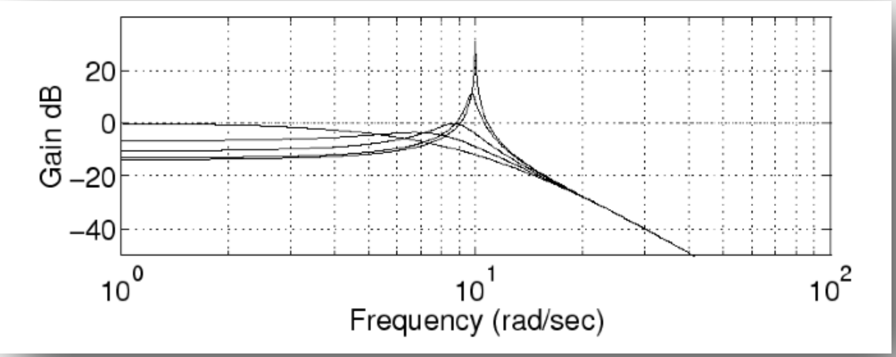


Figure 2: The Coupled Mode Filter Structure

Virtual Analog (Stilson-Smith) (1996)

- Alias-Free Digital Synthesis of Classic Analog Waveforms
- Digital implementation of the Moog VCF. Four identical one-poles in series with a feedback loop.
- Sounds great! (MP3) (youTube)





Seer Systems "Reality" (1997)

- Stanley Jungleib, Dave Smith (MIDI, Sequential Circuits)
- Ring-0 SW MIDI synth. Native Signal Processing.
- Offered a number of Sondius Models.

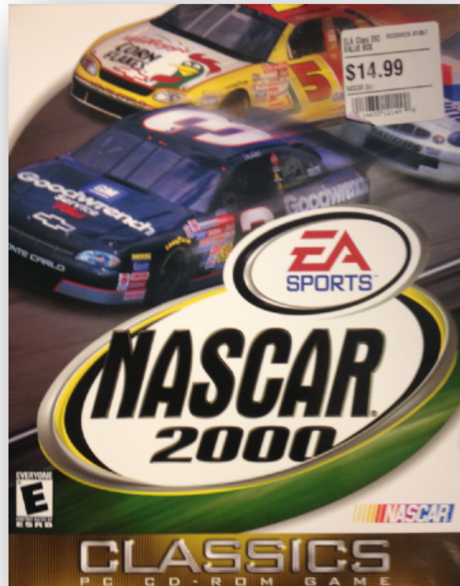
Aureal ASP 301 Chip (1995-1997)

- Targeted for Sound Cards
- Hardware implementation of Digital Waveguide
- A version of the electric guitar ran on this chip

Staccato SynthCore (1999)

- Staccato Systems spun out of Sondius in 1997 to commercialize Physical Modeling technologies.
- SynthCore was a ring-0 synthesis driver that supported both DLS (Down Loadable Sounds) and Staccato's proprietary Down Loadable Algorithms (DLAs). It was distributed in two forms.
- Packaged as a ring-0 “MIDI driver”, SynthCore could replace the wavetable chip on a sound card, as a software based XG-lite/DLS audio solution (SynthCore-OEM) (SigmaTel, ADI)
- Packaged as a DLL/COM service, SynthCore could be integrated into game titles so that games could make use of interactive audio algorithms (race car, car crashes, light sabers) (SynthCore-SDK) (Electronic Arts, Lucas Arts…)

SynthCore Game Models (2000)



- Jet (Stilson) (MP3)
- Race Car (Cascone, et al) (MP3)
- Example models from Staccato ~1999 (windows only)

SynthCore
Wavetable Chip Replacement

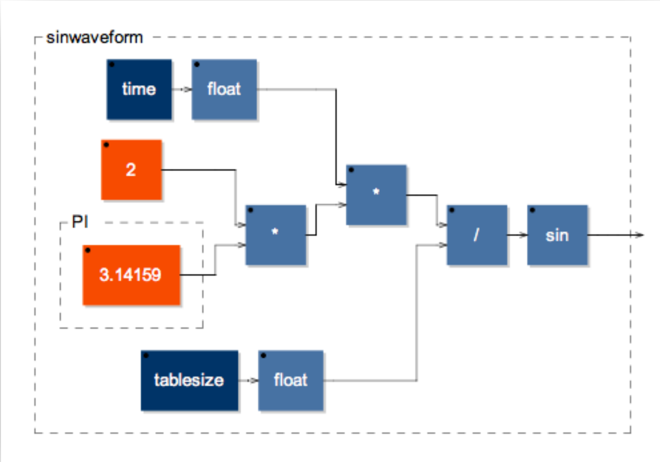
- About half of the General MIDI set was implemented with physical models though few existing MIDI scores could make use of the expression parameters.
- Staccato was purchased by Analog Devices in 2000. ADI combined Staccato's ring-0 software based XG-lite/DLS MIDI synth with a low cost AC97 codec and transformed the PC audio market from sound cards to built-in audio.

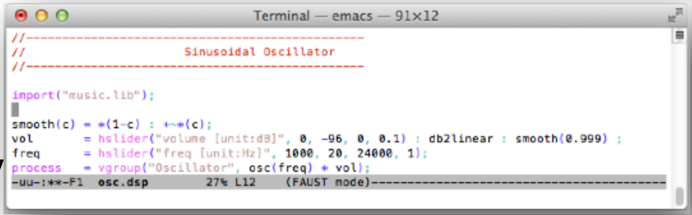
Smule Magic Fiddle (2010)

Smule | Magic Fiddle for iPad [St. Lawrence String Quartet] (youTube)

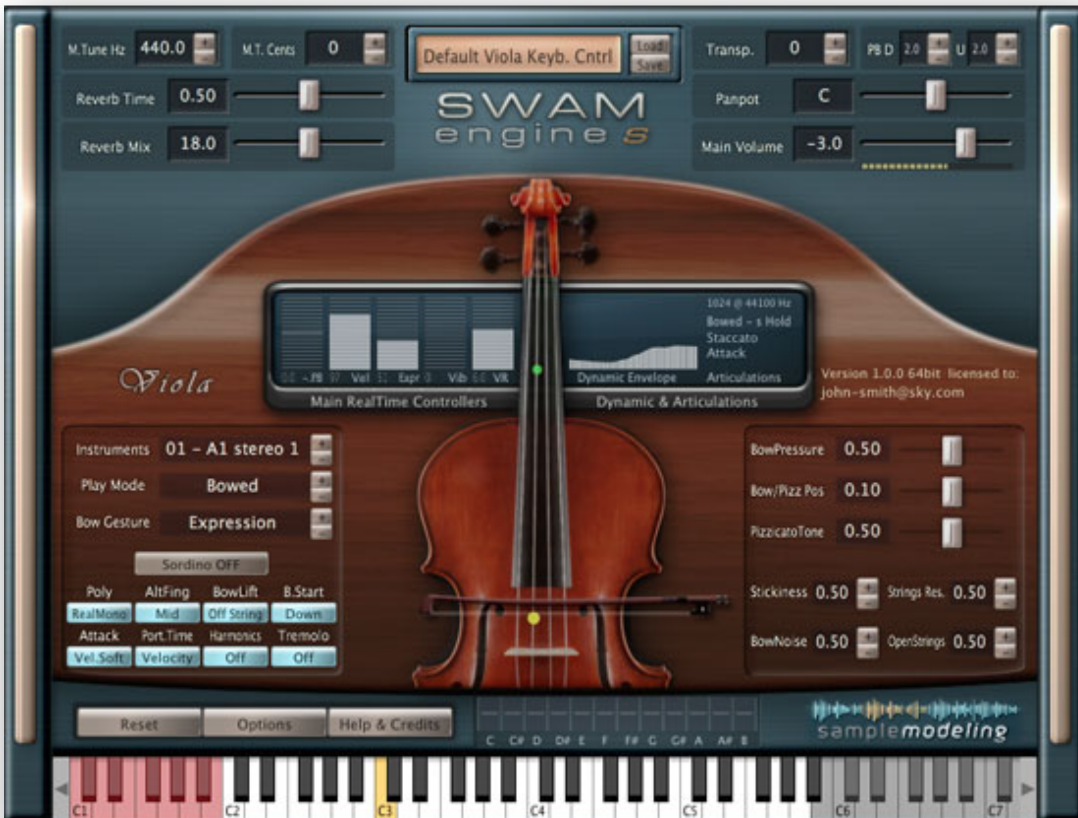
Faust-STK (2011)

- FAUST [Functional Audio Stream] is a synchronous functional programming language specifically designed for real-time signal processing and synthesis.
- The FAUST compiler translates DSP specifications into equivalent C++ programs, taking care of generating efficient code.
- The FAUST-STK is a set of virtual musical instruments written in the FAUST programming language and based on waveguide algorithms and on modal synthesis. Most of them were inspired by instruments implemented in the Synthesis ToolKit (STK) and the program SynthBuilder.





Sample Modeling Swam Engine
Bowed Strings and Reeds



- [YouTube]()

Compute for String Models Over Time

- NeXT Machine (1992)
  - Motorola DSP56001 25MHz 128k dram, 22k sample rate
    - 6 plucks
    - or 2-4 Guitar Strings
- Frankenstein, Cocktail Frank (1996)
  - Motorola DSP56301 72MHz 128k dram, 22k sample rate
    - 6 guitar strings, feedback and distortion,
    - Reverb, wah-wah, flange running on a additional DSPs
- Staccato (1999)
  - 500MHz Pentium, native signal processing, 22k sample rate
  - 6 strings, feedback and distortion used around 80% cpu
- iPhone 4S (2013)
  - 800 MHz A5, 44k sample rate
  - 6 strings, feedback and distortion use around 40% cpu
- iPad Air 2 (2014)
  - 1.5 GHz A8X, 44k sample rate
  - 6 strings, feedback and distortion use around 22% cpu
- iPad Pro (2015)
  - 2.2 GHz ARMv8-A, 44k sample rate
  - 6 strings, feedback and distortion use around 13% cpu

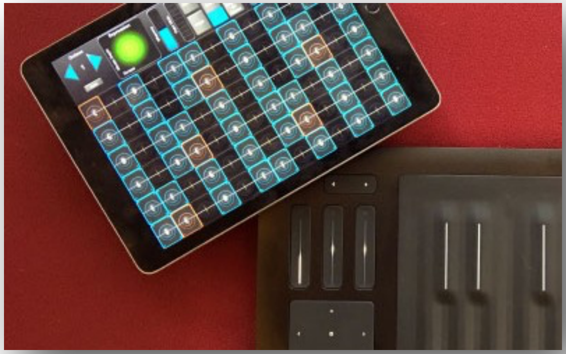
Year	Processor	PluckMark
1992	DSP 56k	3.00
1996	DSP 56k	6.00
1999	Pentium	6.75
2013	A5	13.50
2014	A8X	24.55
2015	ARMv8-A	41.54
2017*	Intel i7 4 cores	140.00
*Estimated		

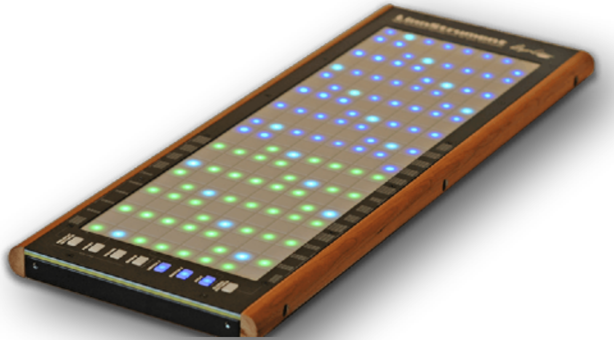
Physical Modeling Was Poised to be the “Next Big Thing” in 1994
So What Happened?

- Expressivity was not the only factor driving PM.
- By 1994, FM was the standard for PC Game Music, in part due to it’s small memory footprint.
- PM was seen as the successor to FM.
- However by 1997 memory was cheap and sampling became common.
- Further, voicing PM is difficult, voicing samples is more direct.

So Why is PM Back!

- Ubiquitous handheld mobile computing devices are packed with sensors that are great for parametrically controlled, physically modeled, virtual musical instruments.
- There is a new generation of MIDI Polyphonic Expression (MPE) controllers





And What About the Future?

- Many different types of expressive “traditional” instruments will become available.
- Lots of possibilities with hybrid PM and Sampling.
- New types expressive instruments. Expressivity in many new forms.

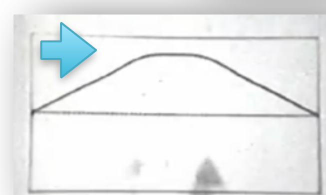
Thanks!

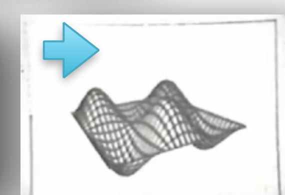
- Mary Albertson
- Chris Chafe
- John Chowning
- Perry Cook
- Jon Dattorro
- David Jaffe
- Joe Koepnick
- Scott Levine
- Fernando Lopez-Lezcano
- Stanford OTL
- Danny Petkevich
- Nick Porcaro
- Bill Putnam
- Kent Sandvik
- Gregory Pat Scandalis
- Julius Smith
- Tim Stilson
- David Van Brink
- Scott Van Duyne
- Stanford CCRMA
- Romain Michon
- Yamaha
- Korg

and CCRMA

About Pat…

- 34 years in the Silicon Valley as an Engineer
- Built my first monophonic electronic instrument in 1970 from a Radio Shack kit.
- Gigged with an Arp Avatar guitar synth (1978)
- Computer Modeling of strings and membranes (1981)
- Researcher in Physical Modeling at Stanford/CCRMA (1994)
- CEO/CTO of moForte





CCRMA

Technology FAQs

Disrupting the Uncanny Valley

- Aiming toward “Suspension of Disbelief”.
- Use modeling to get close to the real physical sound generation experience.
- Sometimes “go over the top”. Its expressive and fun!
- Use statistical variances to disrupt repetitive performance.

Controls With Statistical Variance

- velocity
- pickPosition
- brightness
- t60
- keyNum
- strumSeparationTime
- strumVariation (in auto strum mode)

DEMO:
Strum Variations

So what about Android?

- The Latency Situation with Android has improved.
- At the moment a small percentage of Android Devices support "Low Latency Audio".
- This will certainly improve over time.

Can users jam together across the internet? (1 of 2)

- moForte has investigated this area but is NOT currently working on creating a platform for jamming across the internet.

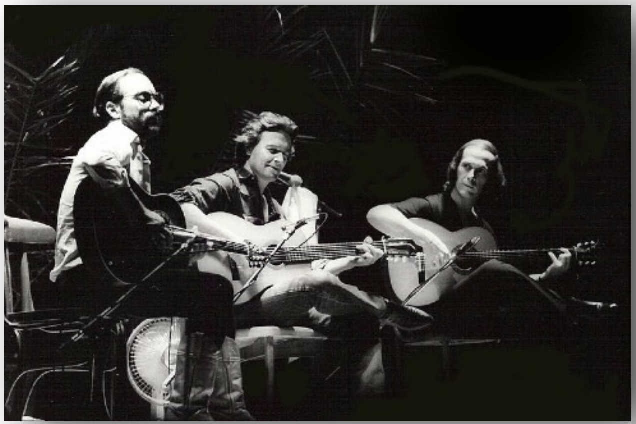
- Latency is a significant issue.
  - see http://en.wikipedia.org/wiki/Latency_(audio)

- The shared performance experience is particularly sensitive to perceived latency. Within the MI (Musical Instrument) industry its a rule of thumb that if **key->sound latency is much larger than 11ms**, the performer will need to "play ahead" leading to a performance that is “loose”, error prone and even frustrating.

- Audio latency facts:
  - Audio Latency in air at sea level/room temp ~1ms/ft
  - Using the speed of light the fastest round trip around the earth is 135ms (vacuum) - 200ms (FO cable).
  - Real inter-network latencies can be much greater and more variable.

Can users jam together across the internet? (2 of 2)

- In Flamenco music the interaction between two players is referred to as **Duende** "It comes from inside as a physical/emotional response to art. It is what gives you chills, makes you smile or cry as a bodily reaction to an artistic performance that is particularly expressive". These players are performing and syncing with around 3ms of air latency. This is typical of many performance situations.



- Some types of performances are possible:
  - Slow performances
  - Cascaded
  - Side by side (one player after the other)
  - Electrifying, tight duets, or real ensembles are less likely to work.

- **For consumers an experience like a band jamming across the internet is not likely be a good experience**

What about Latency?

- The largest source of latency (for ios) appears to between screen interaction and the guitar model. Note that the audio buffer latency is about 5ms.
- We started at 180ms screen to audio out.
- We brought this down to 25-35ms by replacing Apple's gesture handlers with a custom gesture handler. This makes sense. Gesture handling requires analysis of a moderate amount of state to initiate an action.
- MIDI to Audio Latency is about 20-30ms.
- PowerStomp which is audio-in/effects chain/audio out is around 18ms.

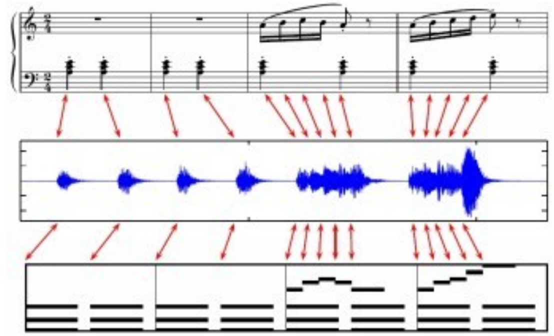
What about wireless audio out of the device?

- We've looked a number of wireless audio solutions. Most are intended for playback of recorded music and have significant latency; some as much as 1 second.
- We've not found a solution yet with reasonable latency.
- We've also looked a number of "legacy" wireless FM transmitters. None of what we have tried have good audio performance.
- We may need to build our own technology in this area.

What about wireless synchronized performances (virtual orchestras)?

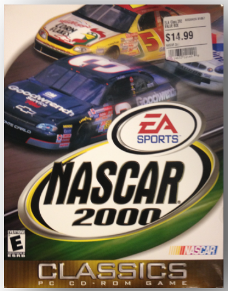
- We have been experimenting with the idea of wireless conductor/performer.
- One device is the conductor and the source of time.
- Each device (performer) has its own part.
- The performers receive temporal corrections from the conductor using techniques similar to NTP.
- These temporal corrections can be very minimal data in the wireless network. We estimate that temporal corrections can be as infrequent as once every 30 seconds.
- This will enable a large number of devices in a wireless network to coordinate a performance.

Can the app listen to your music library and automatically generate charts to play?



- We've been looking at various MIR (Music Information Retrieval) technologies to support this idea.
- There are a number of products on the market that try to do harmonic context recognition (the chords) with various degrees of success.
  - CAPO an assisted/manual transcription program used by music transcribers has some support to recognize chords using spectral techniques.
  - A website called chordify.net that works to recognize the chords for a song using MIR techniques.
- This is an active area of research.
- We may partner with other companies that work in this area. The goal would be to get them to generate our chart XML based on MIR techniques.

Physical Models for Games?



- At Staccato we did physical models for games:

  http://www.scandalis.com/Jarrah/PhysicalModels/index.html#Staccato

  - We had adoption success (1997-2000): The race car and crashes in the EA Nascar line of games, a light sabre for Lucas Arts.
  - The monetization opportunity was not there. The studios wanted to pay as little as $5k/title for a buyout of the technology.

- In 1999 games were selling upwards of $50/seat. Today a game is a few dollars and we don't think that there is a reasonable monetization opportunity.

How accurate is the timing in moForte Guitar?

- In iOS for audio we are using CoreAudio with 5ms buffers.
- The sequencer is very accurate. In iOS we are using a CoreAnimation timer which is tied to the graphics refresh rate.
- We are using standard techniques to manage jitter (~2ms on average).

Why even model a guitar, don't samples sound great?

Currently implement Articulations
Apagado
Arpeggio strum
Bend
Bend by distressing the neck
Burn or destroy guitar
Feedback harmonics
Finger picking
Glissando
Hard dive with the whammy bar
Harmonic
Muted strum
Pinch harmonic
Play harmonics with tip of finger and
Polyphonic bend
Polyphonic slide, Polyphonic slide +
Scrape
Slide
Staccato
Steinberger trans- trem
Strum
Surf apagado
Surf quick slide up the neck
Tap time
Vibrato
Walk bass
Whammy bend
Whammy spring restore

- Sampled guitars do sound great. But they are not interactive, and they can have a flat repetitive playback experience.
- By modeling the guitar its possible to make interactive features like, feedback, harmonics, pick position, slides brightness, palm muting part of a performance.
- moForte has identified a list of around 70 guitar articulations that can be used by players. The physicality of the model makes it possible for these articulations to be used in performances.

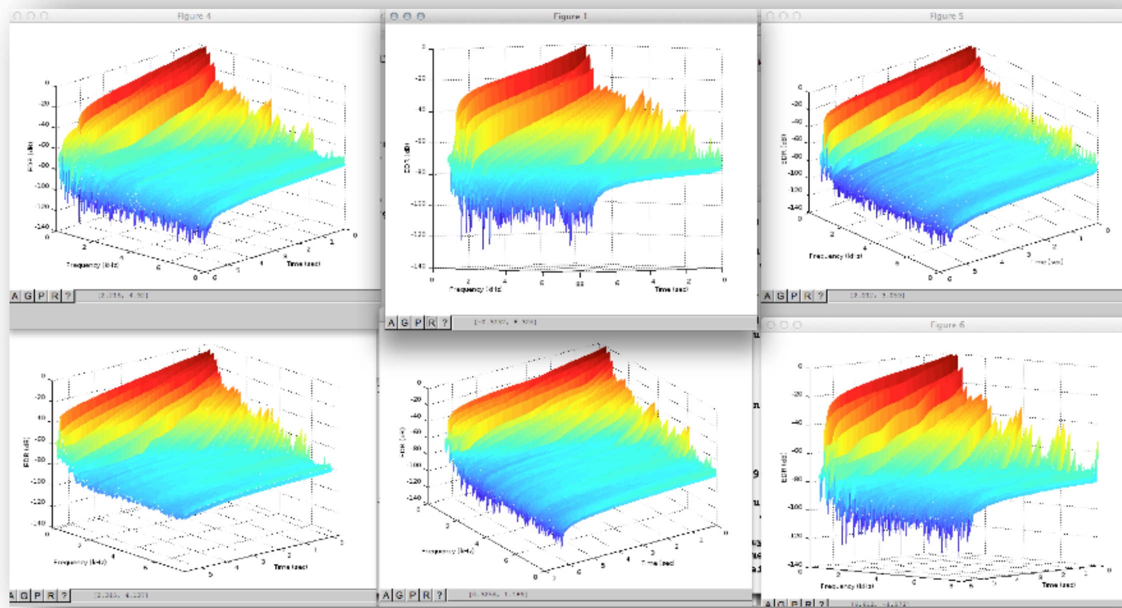
Future Articulations
Bottleneck (portamento Slide)
Bowing
Bridge/neck short strings
ebowing
Finger Style (Eddie Van Halen)
Hammer, polyphonic hammer
Individual String Pitch Bend
Legato
Pluck, sharp or soft pick
Pop
Prepared string (masking tape)
Pull, polyphonic pull
Rasqueado
Reverb spring Bang.
Scrape+ (ala Black Dog)
Slap
Strum and body tap
Strum and string tap
Touching Ungrounded Cable
Trill
Trill up the neck into echo
Vibrato onset delay
Volume pedal swell
Volume pedal swell into delay device

Do you model all oscillation modes of the string, x-y-torsional. Coupling, multi stage decay?

- We are modeling one of the primary modes.
- We are looking at adding bridge coupling
- As available compute increases we may add a second primary mode as well as other features.

What about acoustic guitars and all the other chordophones?

- Yes we are working on many different types of electric and acoustic chordophones.
- moForte is developing a calibration process that will allow us to generate model data for these different instruments.
- These instruments will be offered as in-app purchases for moForte Guitar.



When will moForte offer a Ukulele?

- We are working on modeling a ukulele along with a number of other chordophones.
- These instruments will be offered as in-app purchases for moForte Guitar.
- **The ukulele is one of the most requested instruments that we are asked about ;-)**