Spectral Audio Modeling: Why Did It Evolve and Do We Need It Now?

Julius Smith
CCRMA, Stanford University

ADC-23

November 15, 2023

Overview

Goal:

A *zoomed-out overview* of spectral audio modeling, from evolution to AI

Intended Audience:

- Audio signal-processing engineers interested in latest AI audio developments
- AI practitioners interested in more about audio signal-processing techniques

Relevant Questions:

- Why did we evolve spectrum analyzers in our ears?
- How did that drive our use of spectrum analysis and processing?
- How did AI pick up on all that, and will explicit spectra even survive in AI?

Outline of Topics

- Human Hearing
- Why Spectra?
- Spectral Synthesis
- AI Audio Synthesis
- Spectra in AI

Download These Overheads

- Downloadable from the ADC23 website
- or the JOS Home Page at CCRMA
  (Web-search for “Julius Smith CCRMA”):
  `https://ccrma.stanford.edu/~jos/pdf/ADC23.pdf`
- *Click on the many sound-example links!*

CCRMA Spectral Modeling Origins

Stanford AI Lab (SAIL) 60s-80s

John Chowning

Stanford Knoll (main campus)

Max Mathews

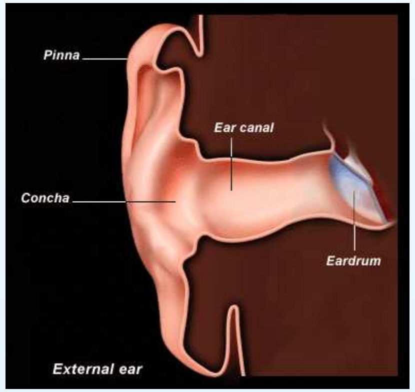
CCRMA was at SAIL (60s-80s) then The Knoll (President's Residence then Music Dept.)

JOS Courses Developed

- **Music 320A**: Audio Spectrum Analysis
- **Music 320B**: Audio Filter Analysis and Structures
- **Music 420A**: **Physical** Audio Signal Processing
- **Music 421A**: **Time-Frequency** Audio Signal Processing

420A

$\longrightarrow$

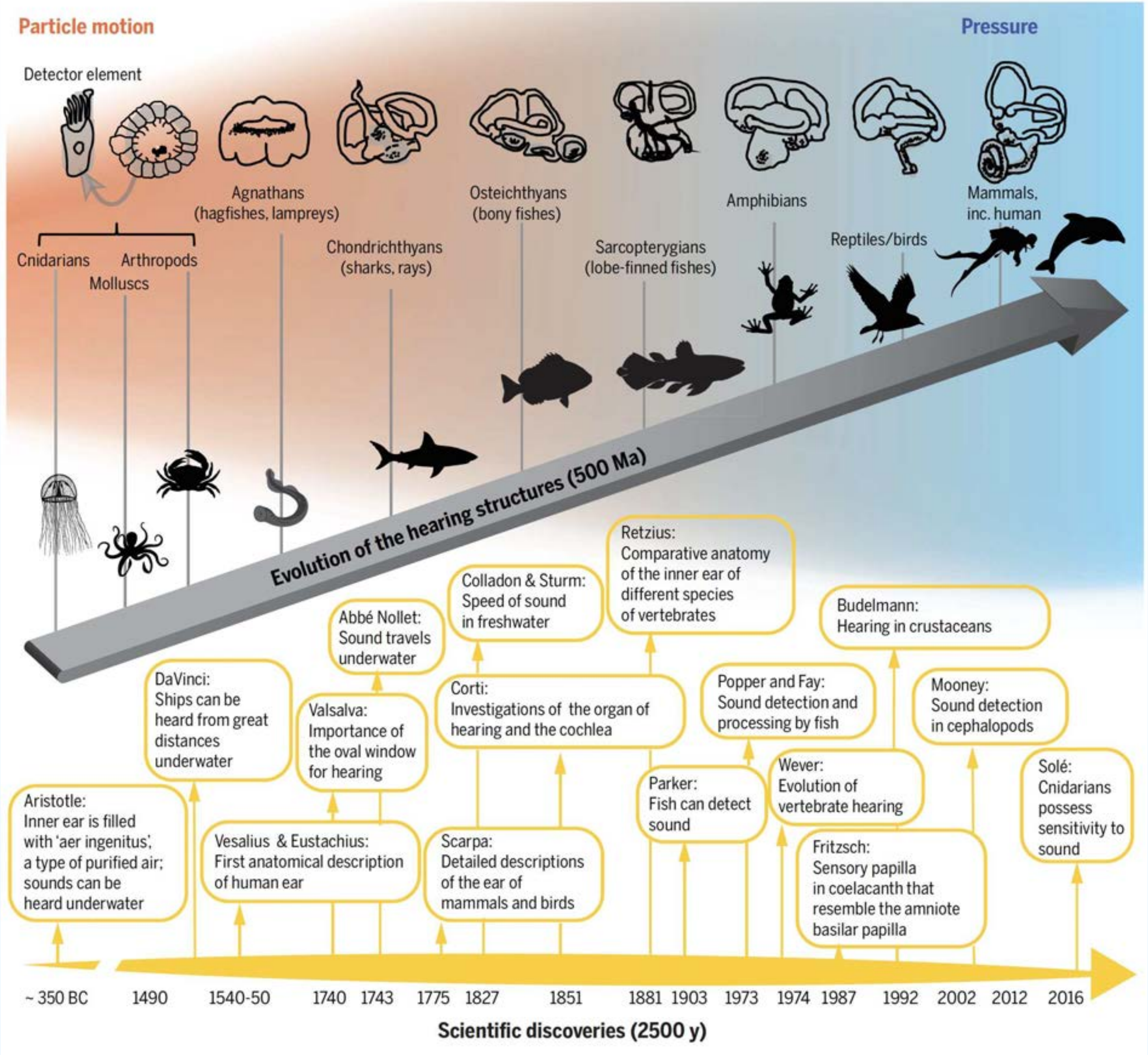


421A

All four textbooks **free online**

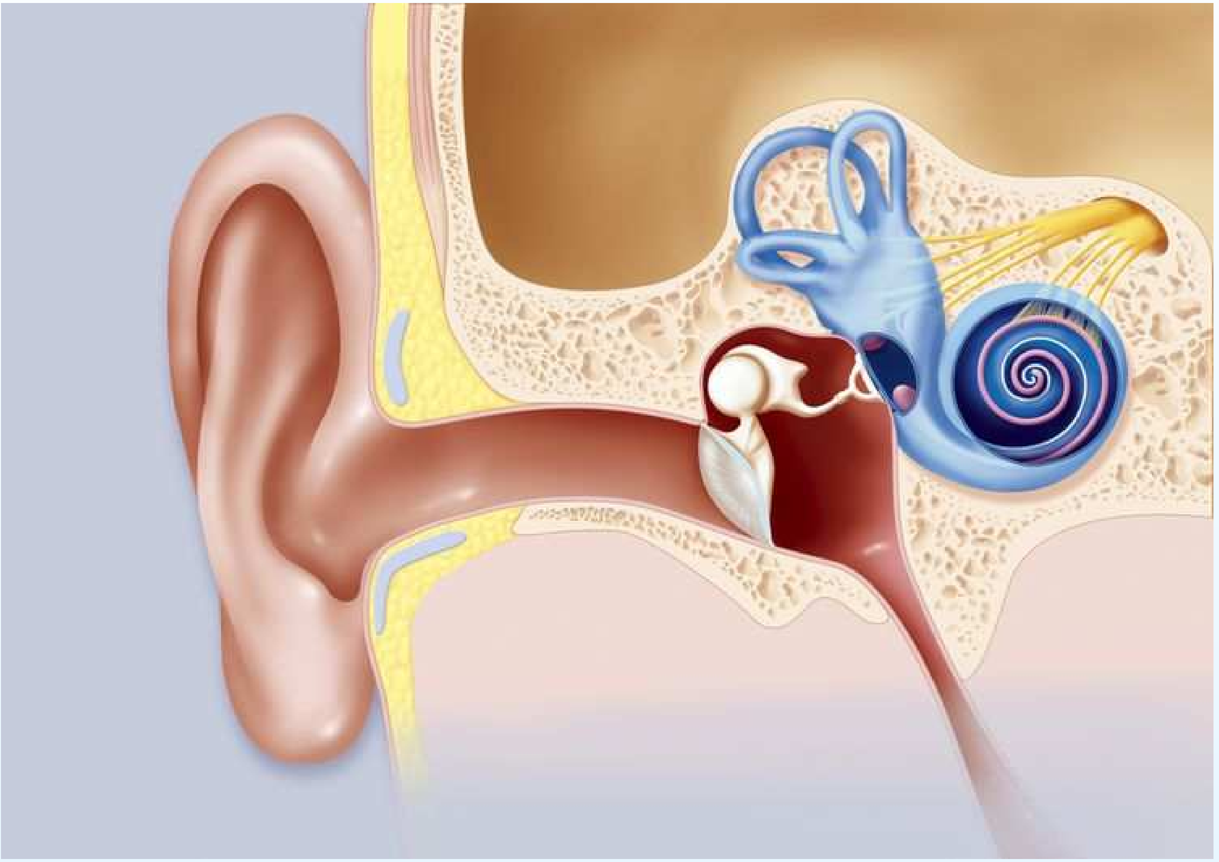
Early Spectral Audio Processing

Evolution of Hearing



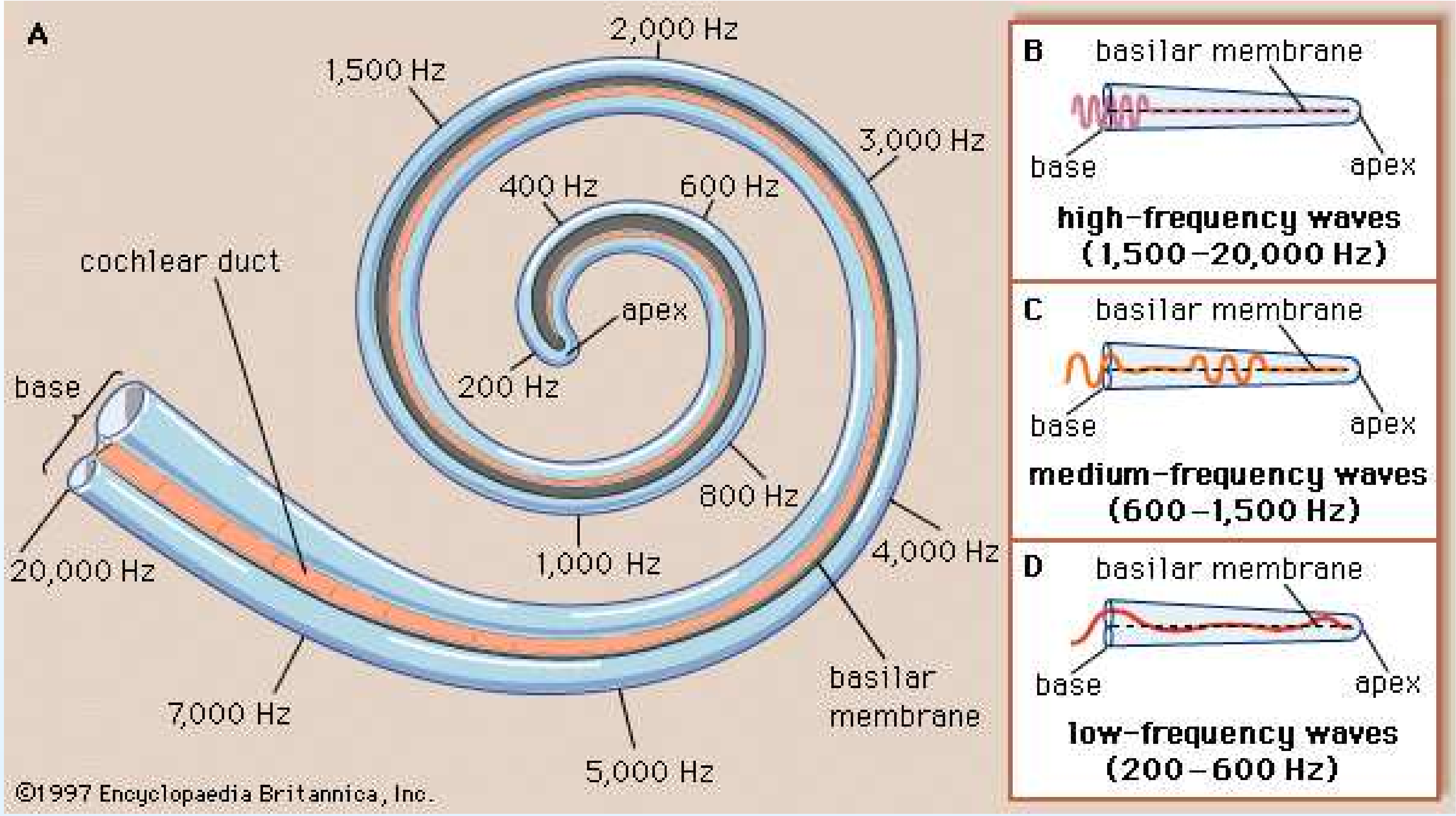
http://science.sciencemag.org/

Human Ears



https://www.verywellhealth.com/cochlea-anatomy-5069393

Inner Ear Spectral Processor



https://www.britannica.com

First Known Polyphonic Spectral Audio Synthesis

https://www.classicfm.com

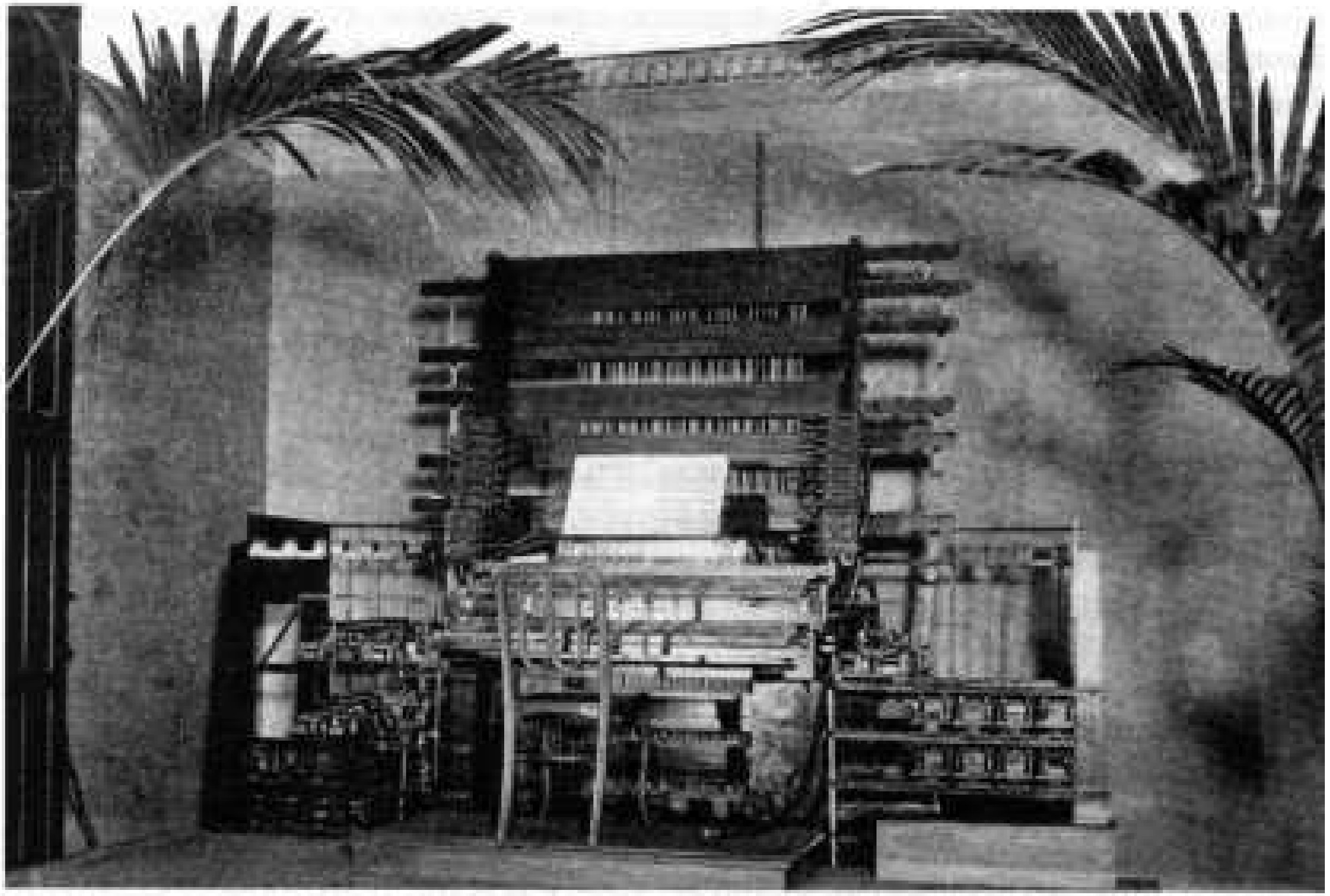
- Greek “Hydraulis” Organ
- aka “Water Organ”
- 3rd Century BC
- This one from ≈1435 AD
- Direct manipulation of pitch
- Neanderthal bone flutes are said to go back 60k years,
  so they win in the “mono” category
- Aurignacian flutes (bone and ivory) said to be created 43k-35k years ago

Pipe organs did complex “additive synthesis” (before “sinusoids”)

Telharmonium (1898)

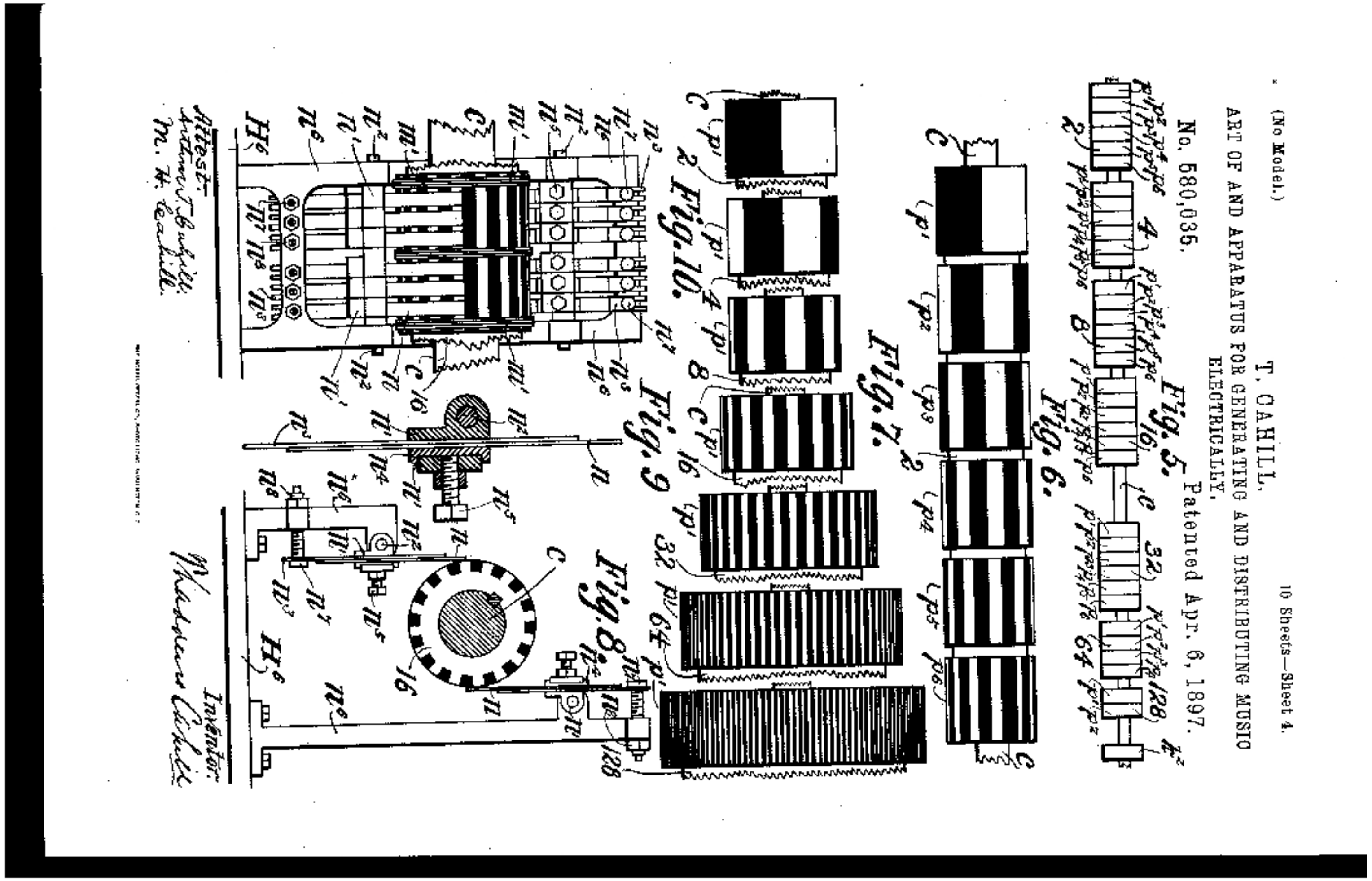
Telharmonium (Cahill 1898)

U.S. patent 580,035:
"Art of and Apparatus for Generating and Distributing Music Electrically"

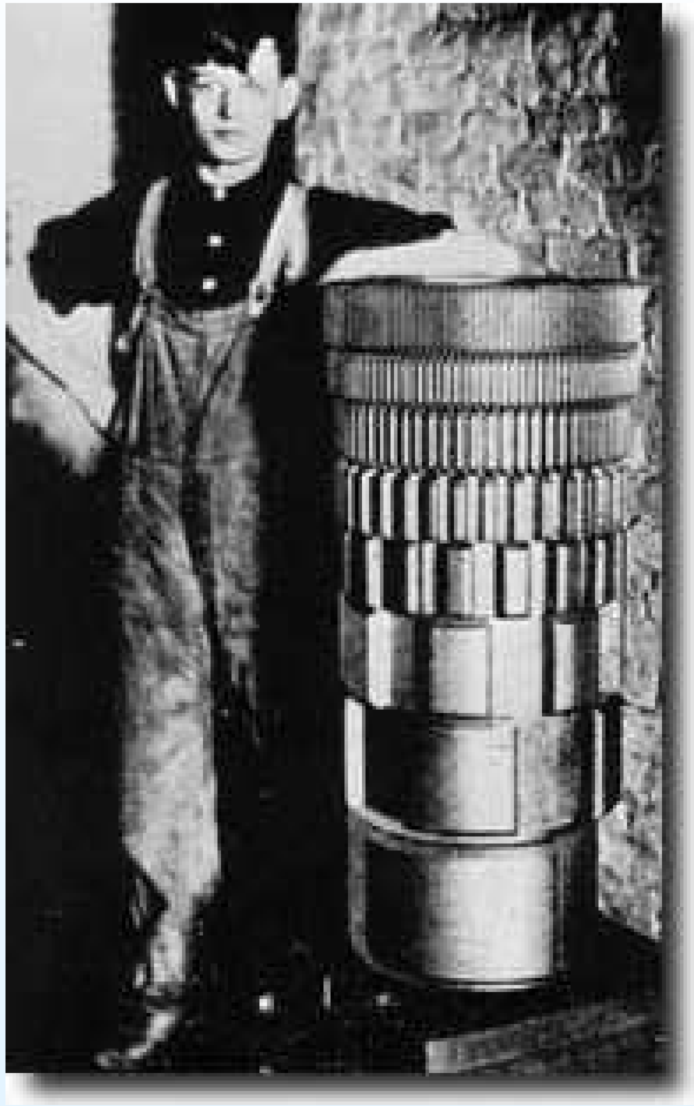


Telharmonium Rheotomes

Forerunner of the Hammond Organ Tone Wheels



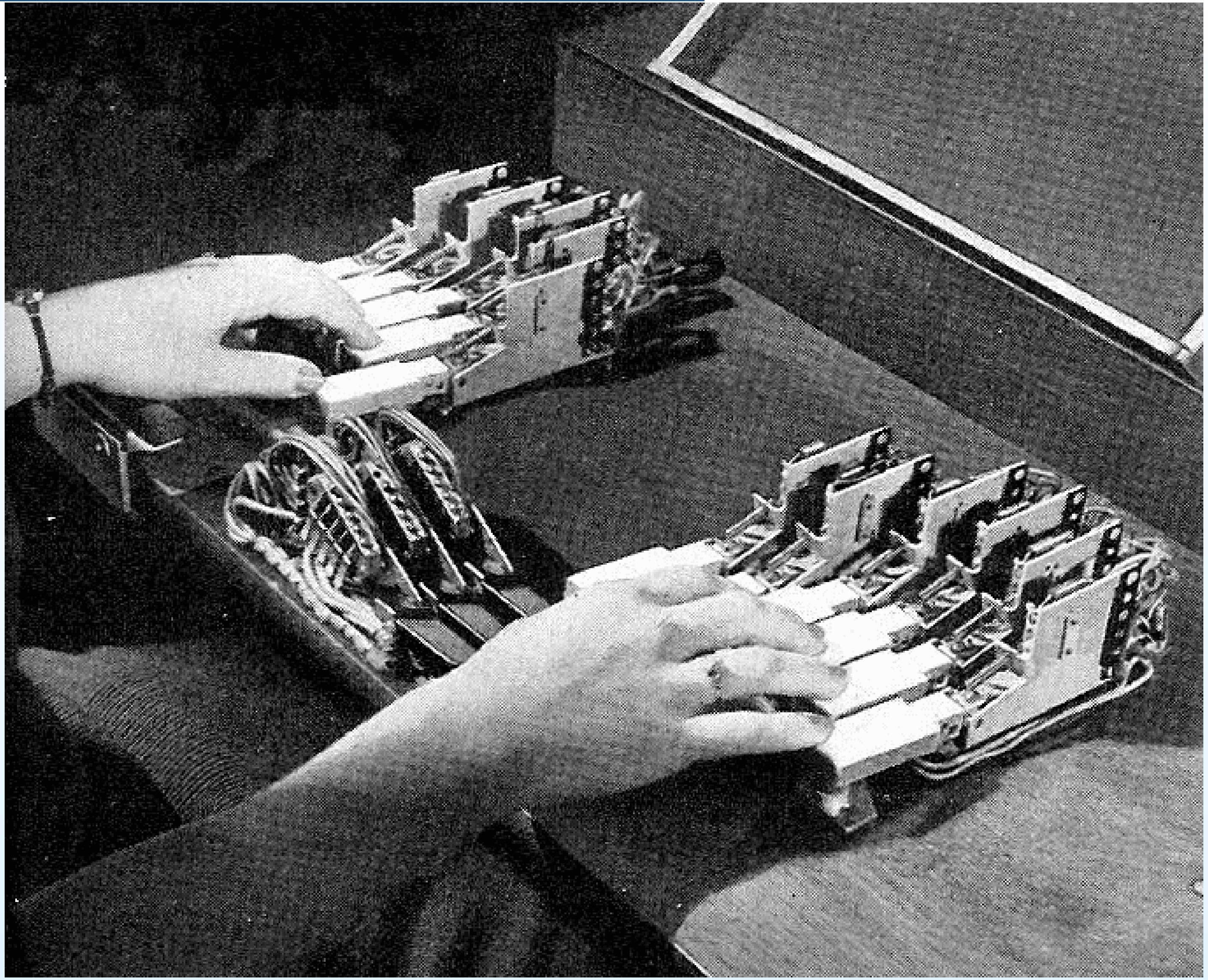
Telharmonium Rotor (early “Tonewheel”)



Hammond influenced: `https://en.wikipedia.org/wiki/Tonewheel`

The Voder (1939)

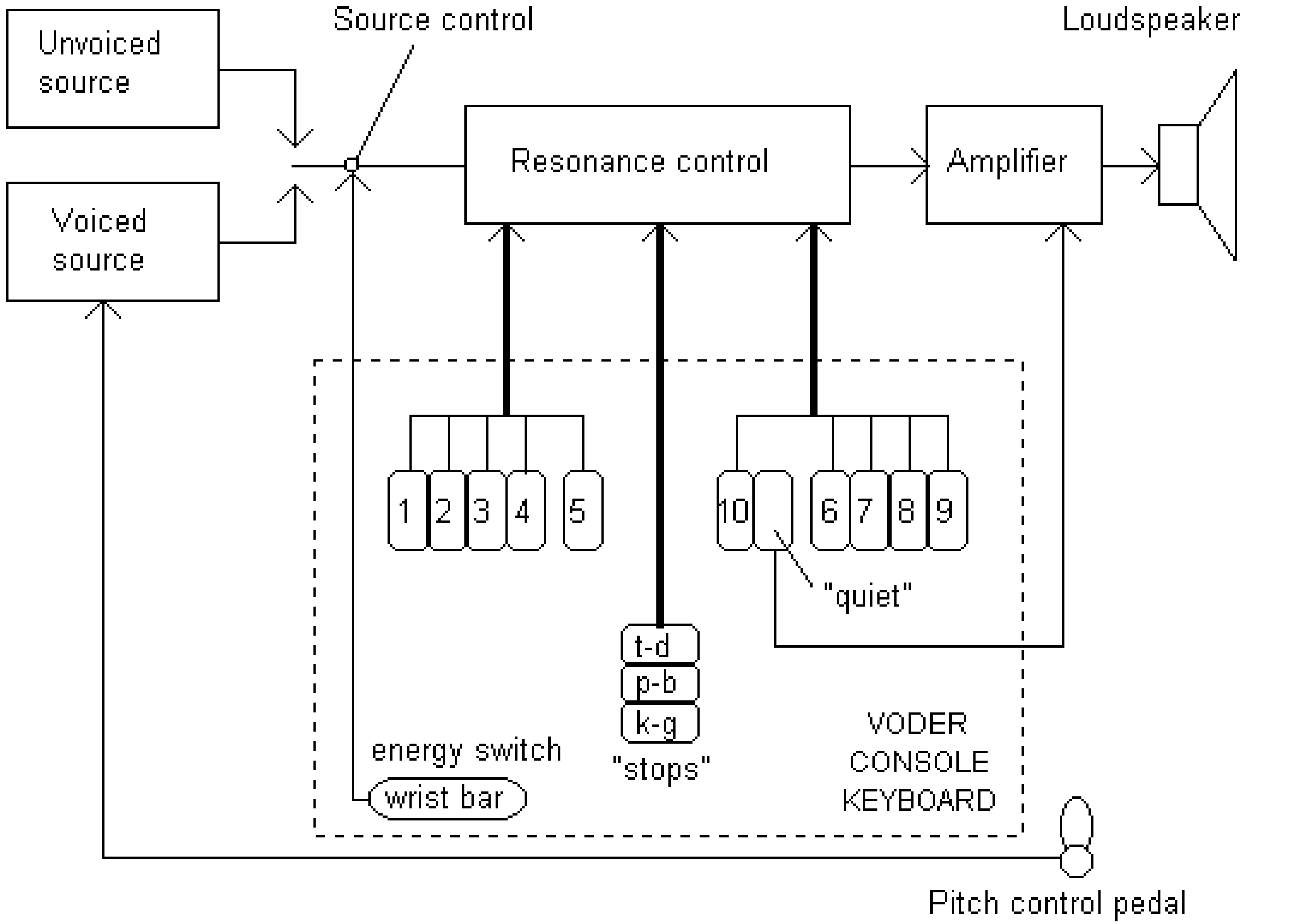
The Voder (Homer Dudley — 1939 Worlds Fair)



$>$

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

Voder Keyboard



http://www.acoustics.hut.fi/publications/files/theses/lemmetty_mst/chap2.html — (from Klatt 1987)

Voder Schematic

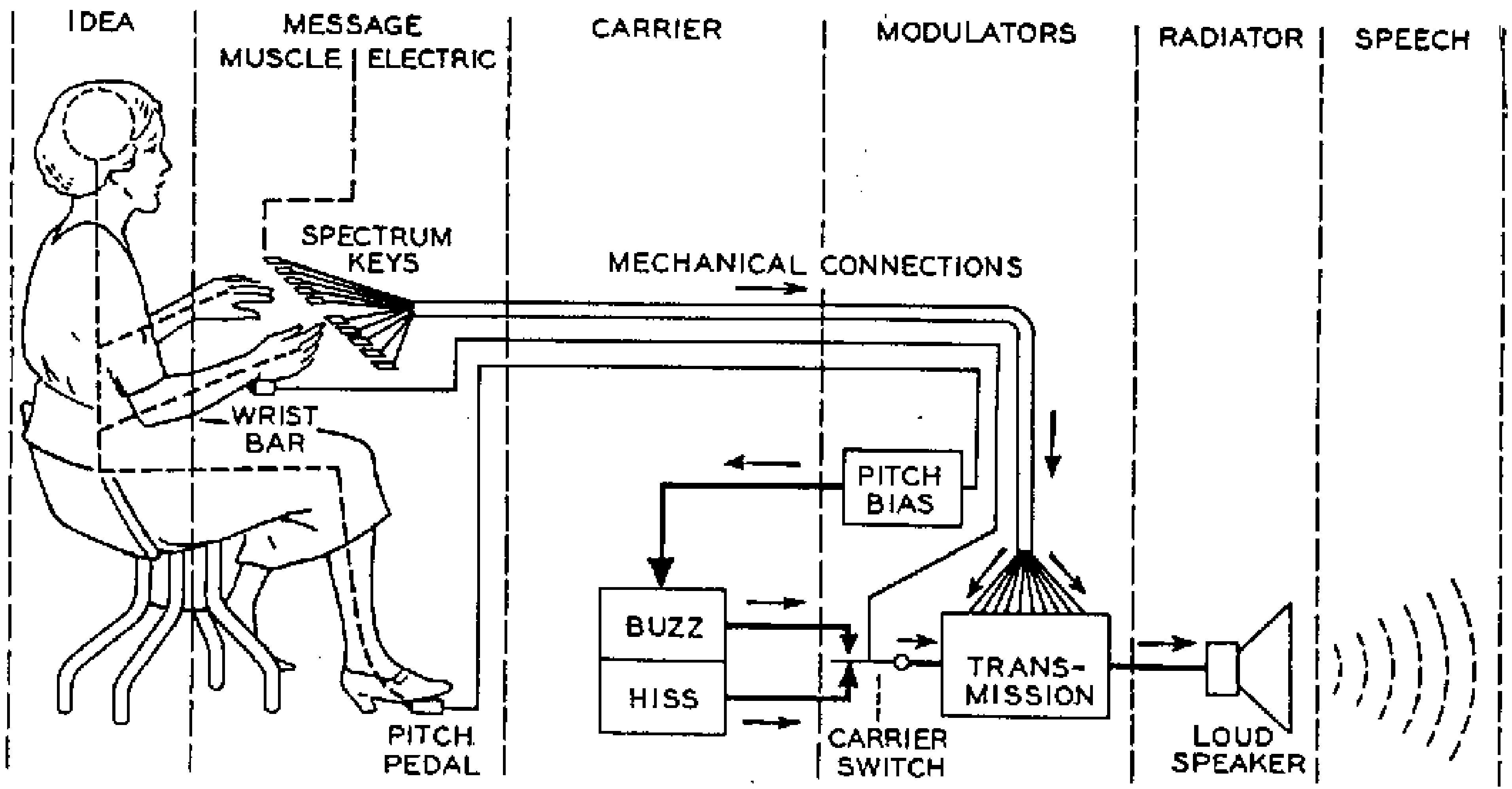


Fig. 8—Schematic circuit of the voder.

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

Voder Demos

- Voder Demo (Audio and Video)
- More Voder Demos - Audio Only [Demos Begin]

Channel Vocoder (1928)
(“Voice Coder”)

Vocoder Analysis & Resynthesis (Dudley 1928)

Analysis:

- Ten analog bandpass filters between 250 and 3000 Hz: Bandpass $\to$ rectifier $\to$ lowpass filter $\to$ *amplitude envelope*
- Voiced/Unvoiced decision made
- Fundamental frequency F_0 measured for voiced case

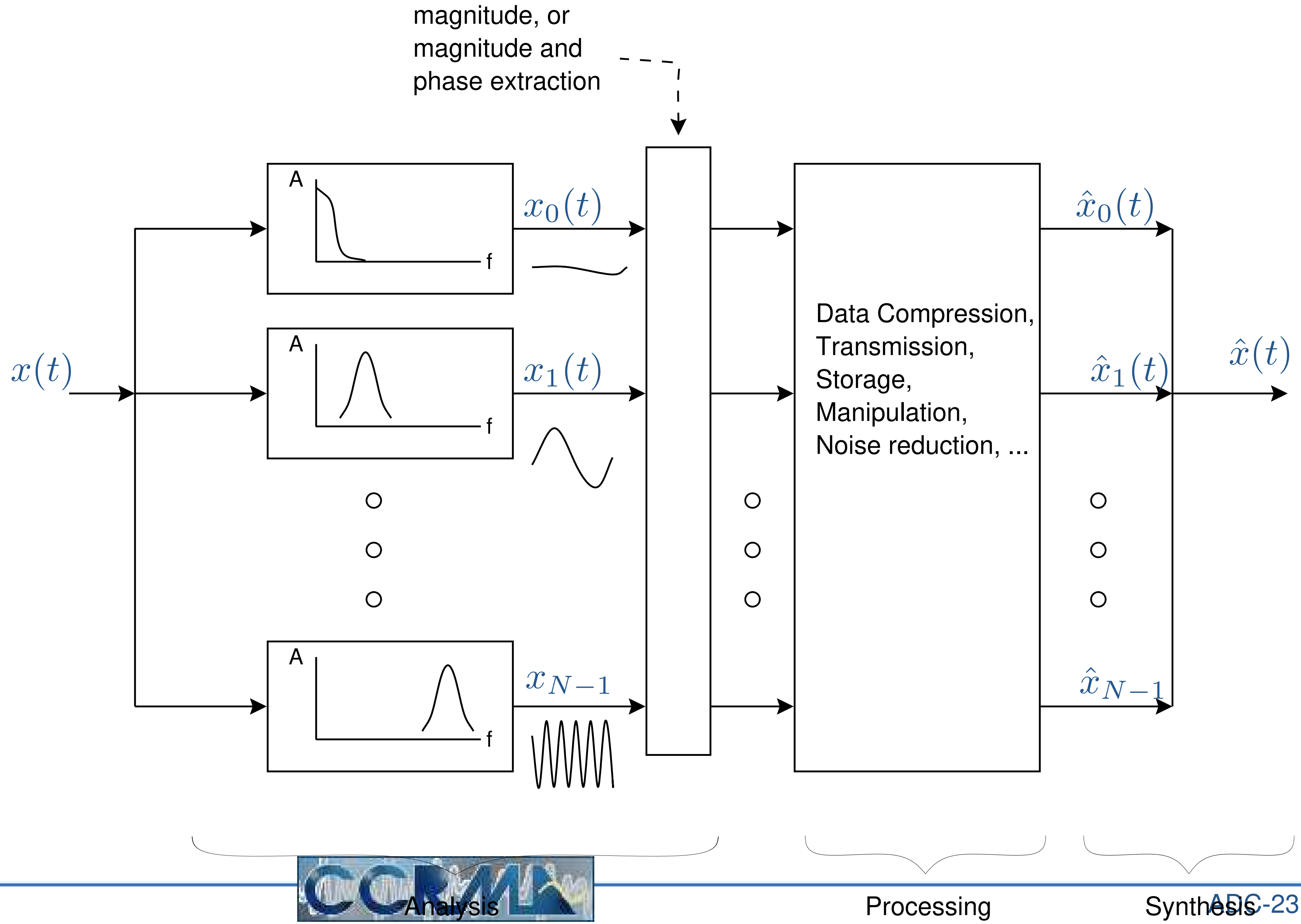
Synthesis:

- Ten matching bandpass filters driven by a
  - “buzz source” (voiced), or
  - “hiss source” (unvoiced)
- Bands were scaled by amplitude envelopes and summed
- Said to have an “unpleasant electrical accent”

Related Speech Models:

- The Vocoder is an early *source-filter* model for speech
- *Linear Predictive Coding* (LPC) of speech is another

Vocoder Filter Bank Analysis/Resynthesis



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

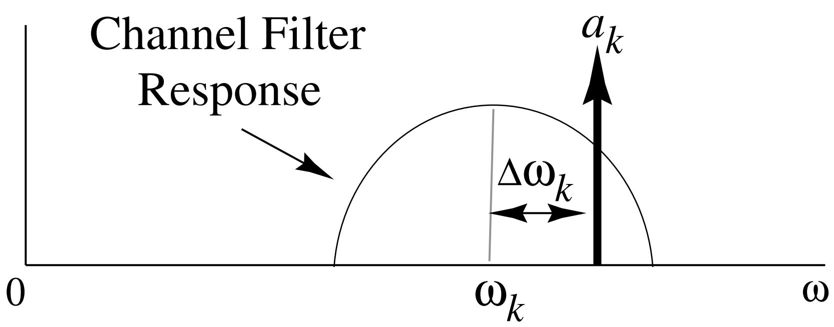
- ***Phase* Vocoder:** Identity system in absence of modifications

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

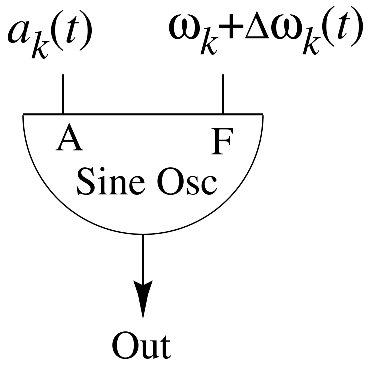
The Phase Vocoder (1966)

Phase Vocoder Analysis for Additive Synthesis (1976)

Analysis Model

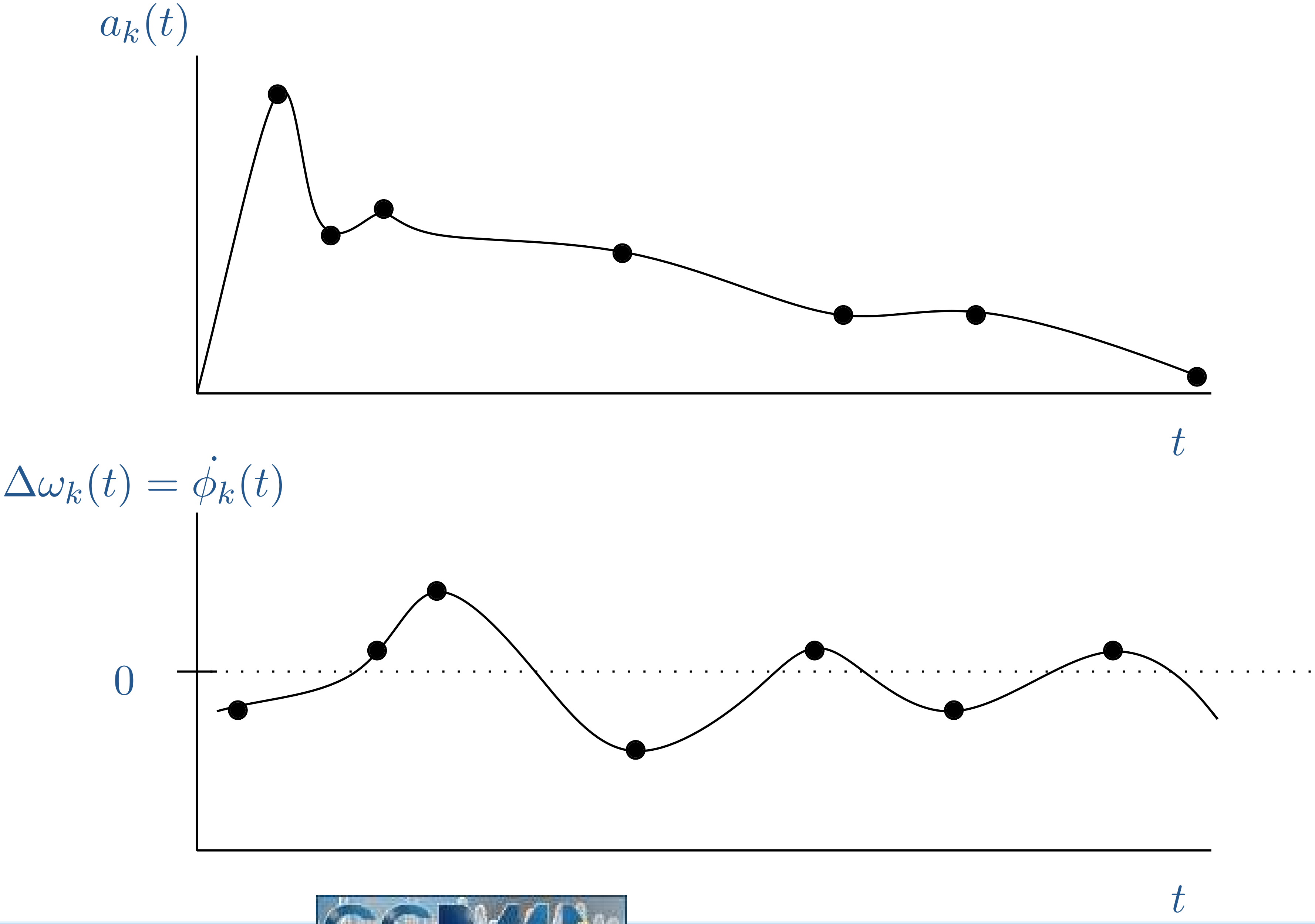


Synthesis Model



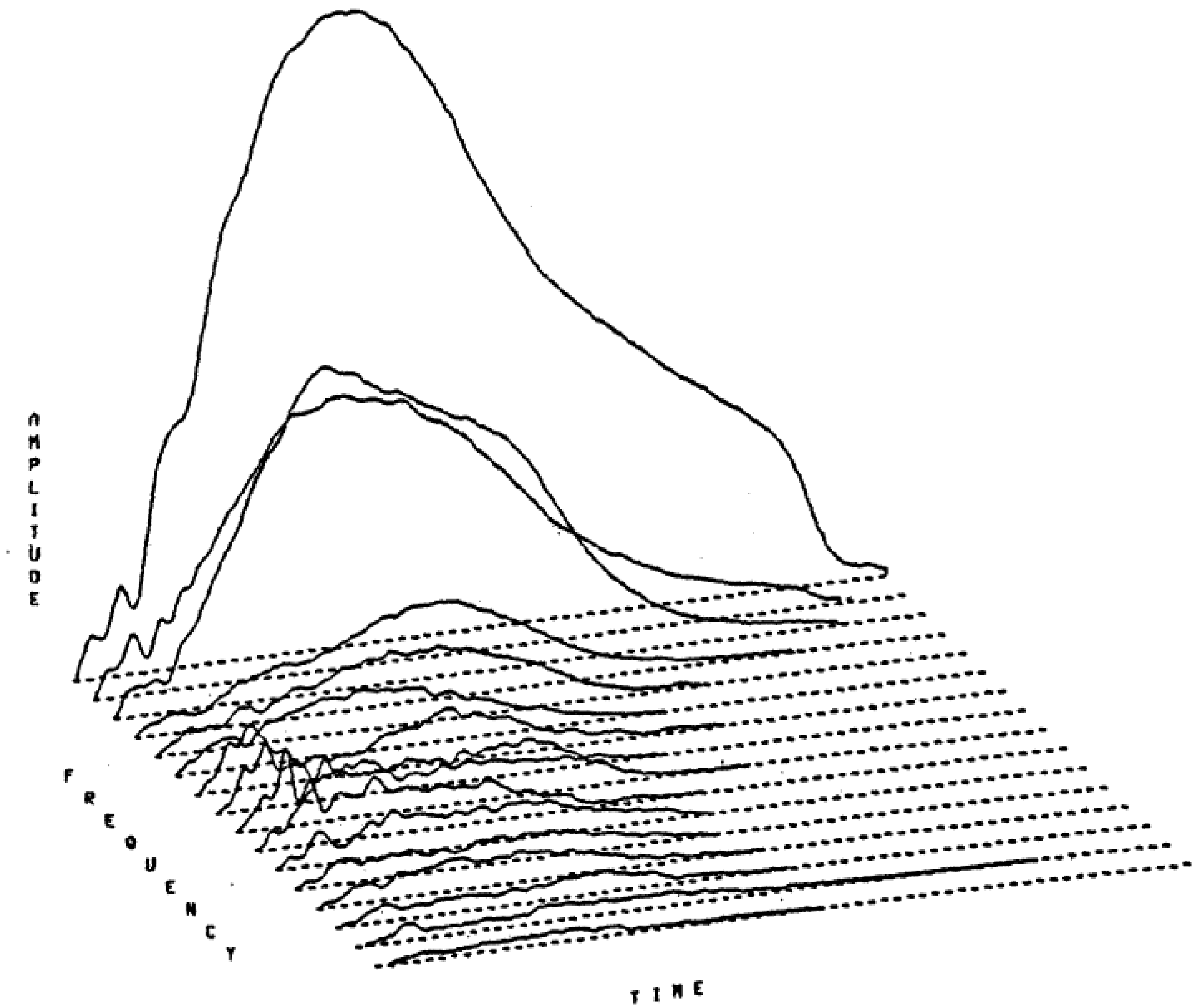
- Early "channel vocoder" implementations (hardware) only measured amplitude $a_k(t)$ (Dudley 1939)
- The "phase vocoder" (Flanagan and Golden 1966) added phase tracking in each channel
- Portnoff (1976) developed the FFT phase vocoder, replacing the heterodyne comb in computer-music additive-synthesis analysis (James A. Moorer 1975)
- Inverse FFT synthesis (Rodet and Depalle 1992) gave faster sinusoidal oscillator banks

Amplitude and Frequency Envelopes



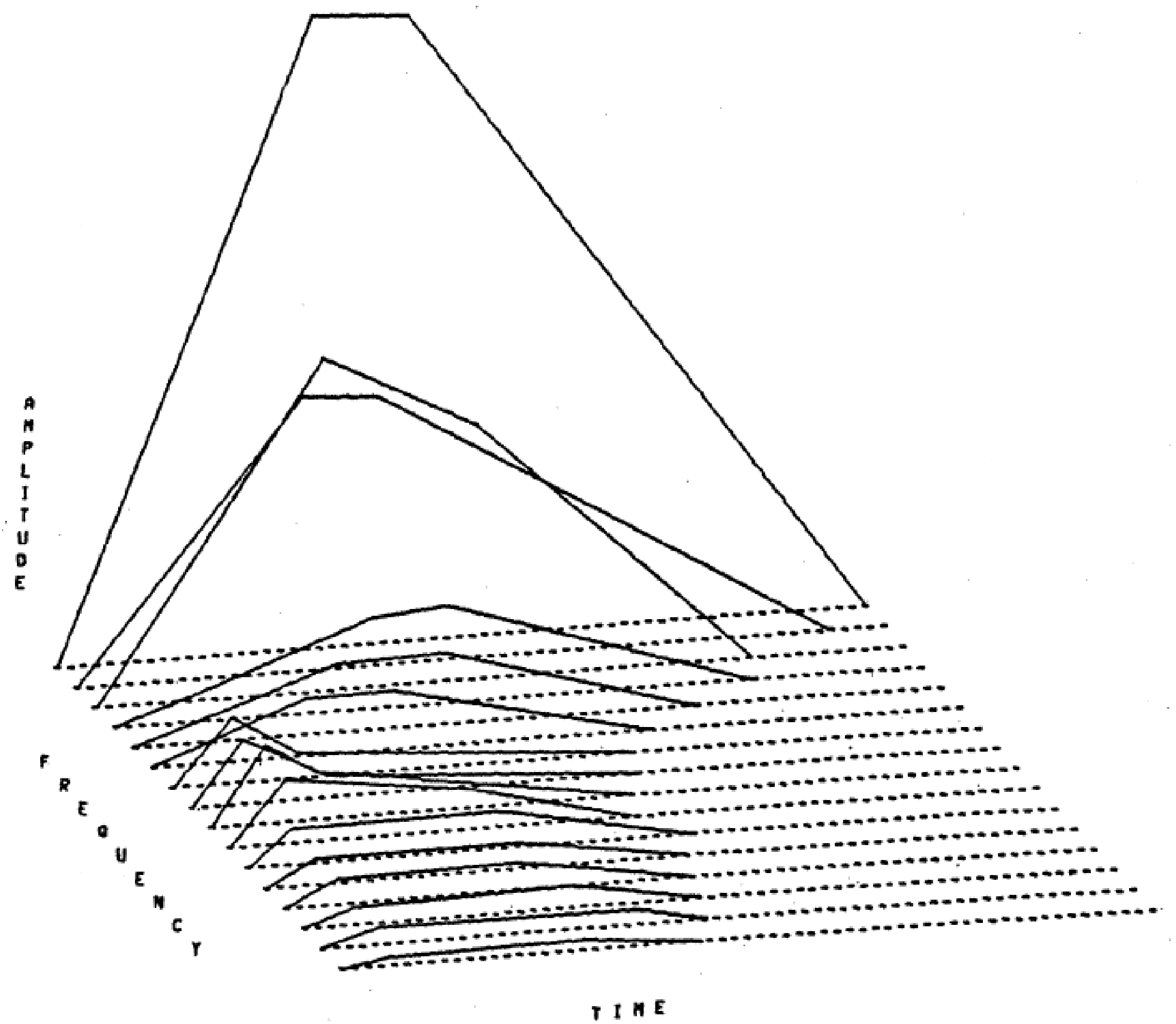
Additive Synthesis (1969)

Classic Additive-Synthesis Analysis (Heterodyne Comb)



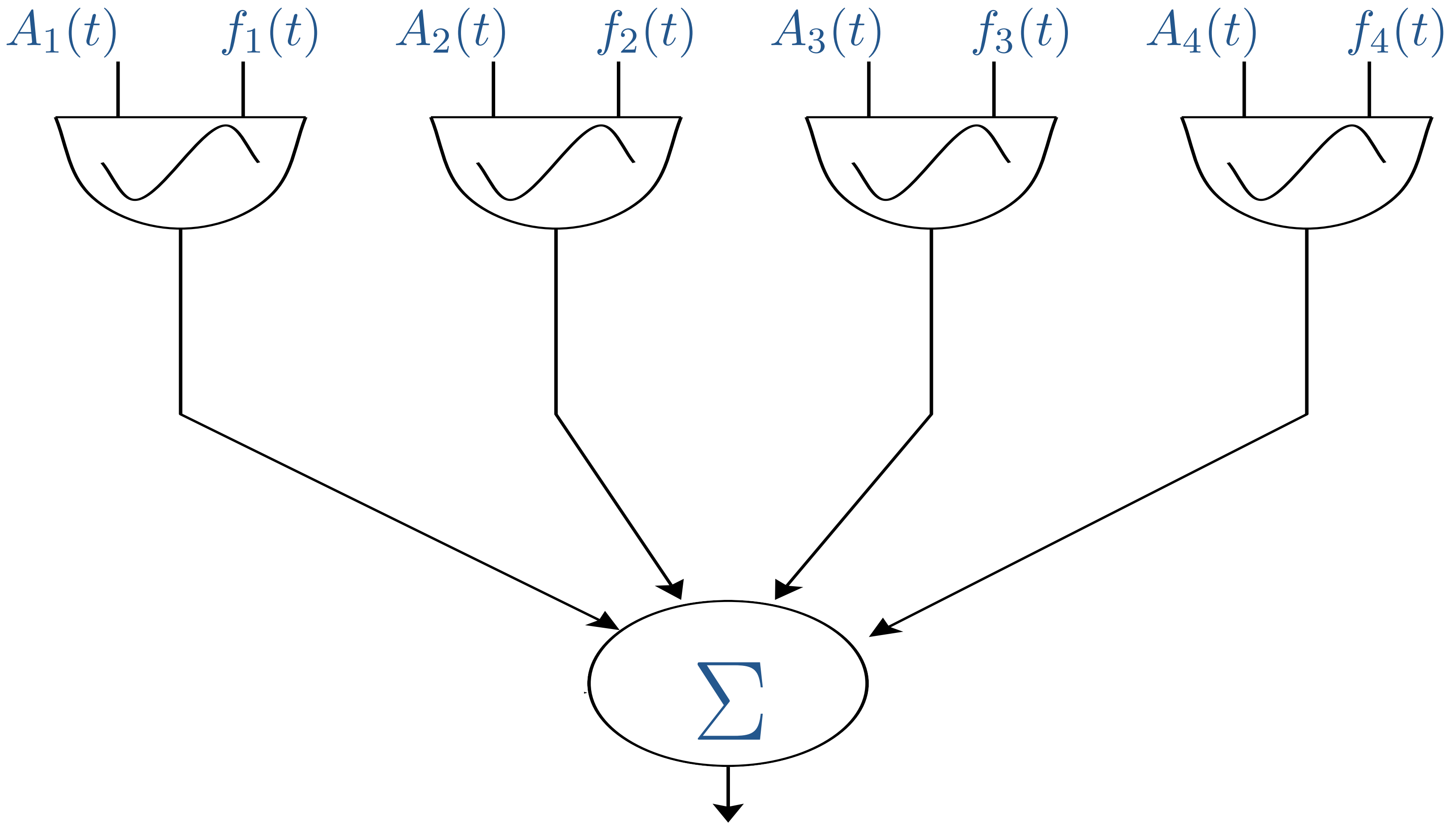
John Grey 1975 — CCRMA Tech. Reports 1 & 2
(CCRMA "STANM" reports — available online)

Classic Additive-Synthesis (Sinusoidal Oscillator Envelopes)



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

Classic Additive Synthesis Diagram (Computer Music, 1960s)



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

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 Synthesis (1973)

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

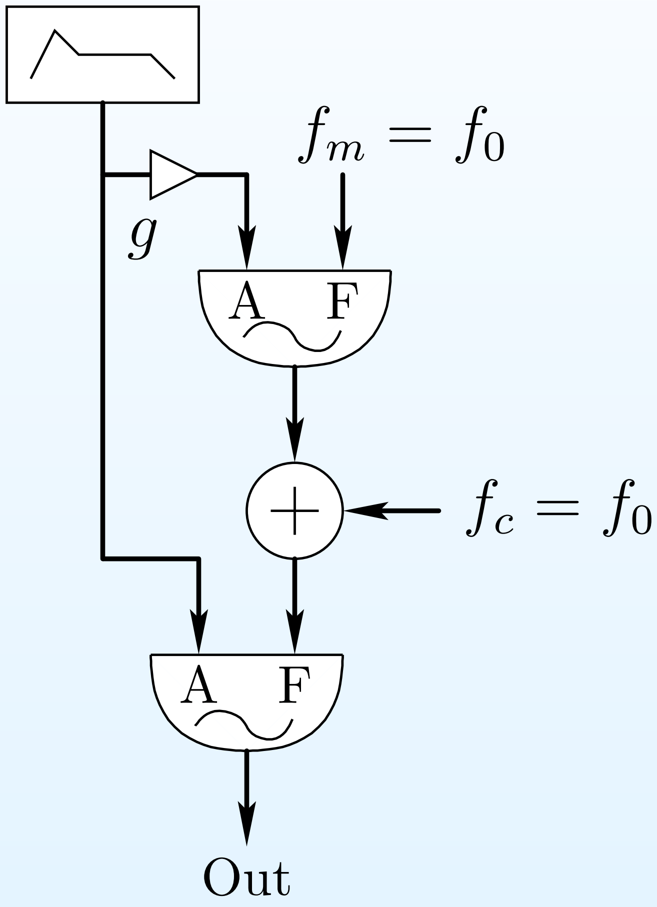
(A_c, ω_c, ϕ_c) specify the *carrier* sinusoid
(A_m, ω_m, ϕ_m) specify the *modulator* sinusoid

Can also be called *phase modulation*

Simple FM "Brass" Patch (Chowning 1970–)

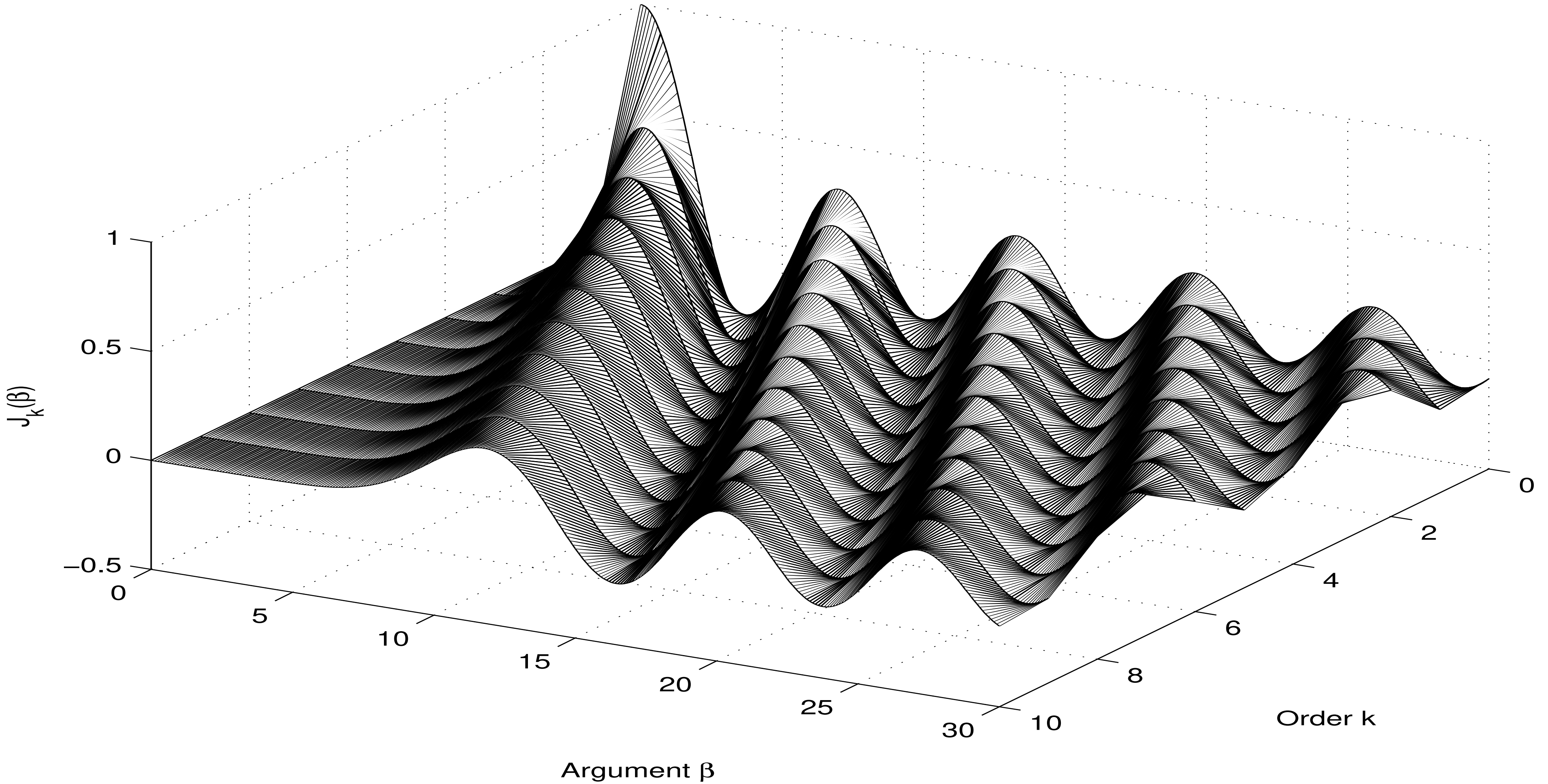
Jean-Claude Risset observation (1964–1969):

Brass bandwidth $\propto$ amplitude



FM Harmonic Amplitudes (Bessel Function of First Kind)

Harmonic number k, FM index β:



- Introduction
- Origins
- Telharmonium
- Voder
- Channel Vocoder
- Phase Vocoder
- Additive Synthesis
- FM Synthesis
  - FM Synthesis
  - FM Formula
  - FM Patch
  - FM Spectra
  - FM Examples
  - FM Voice
- Sinusoidal Modeling
- DDSP
- Audio in Generative AI
- Spectra or Not in AI
- Summary

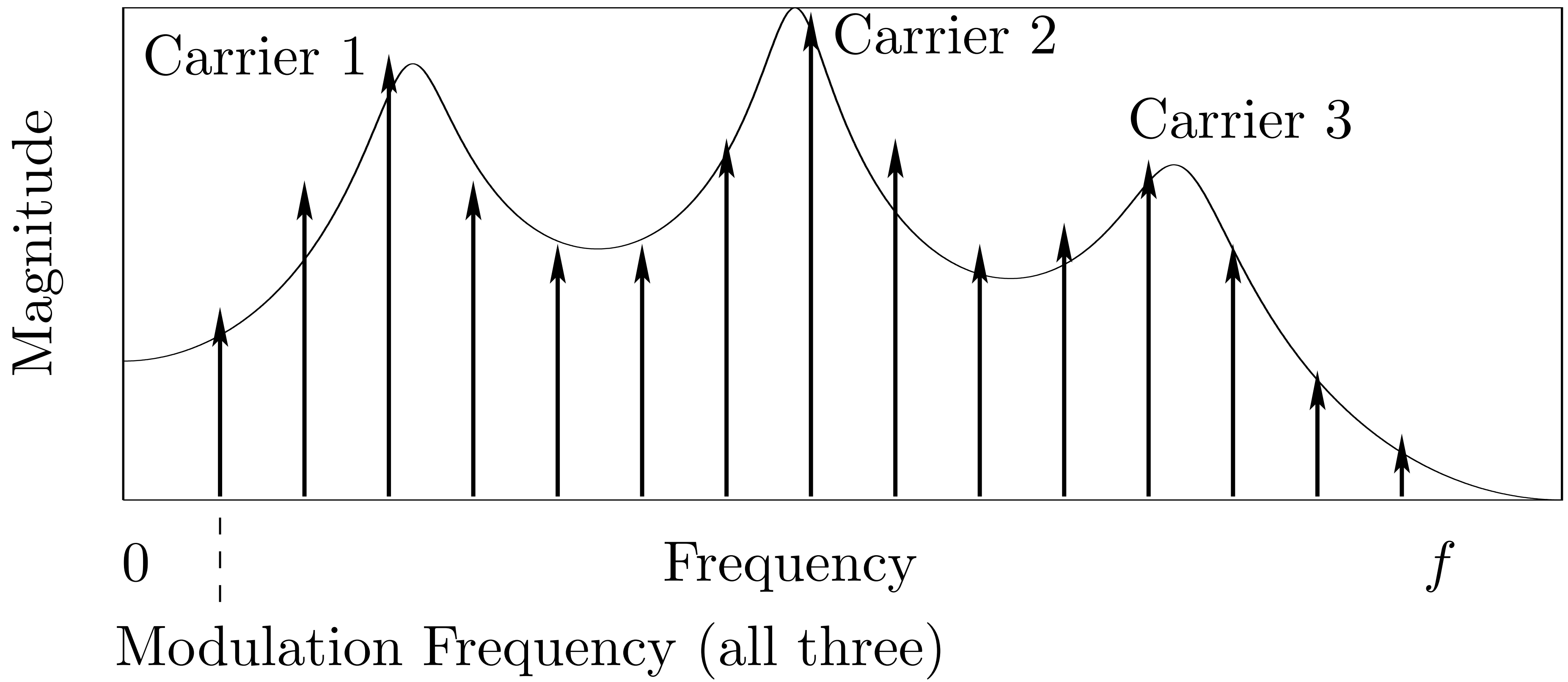
Frequency Modulation (FM) Examples

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

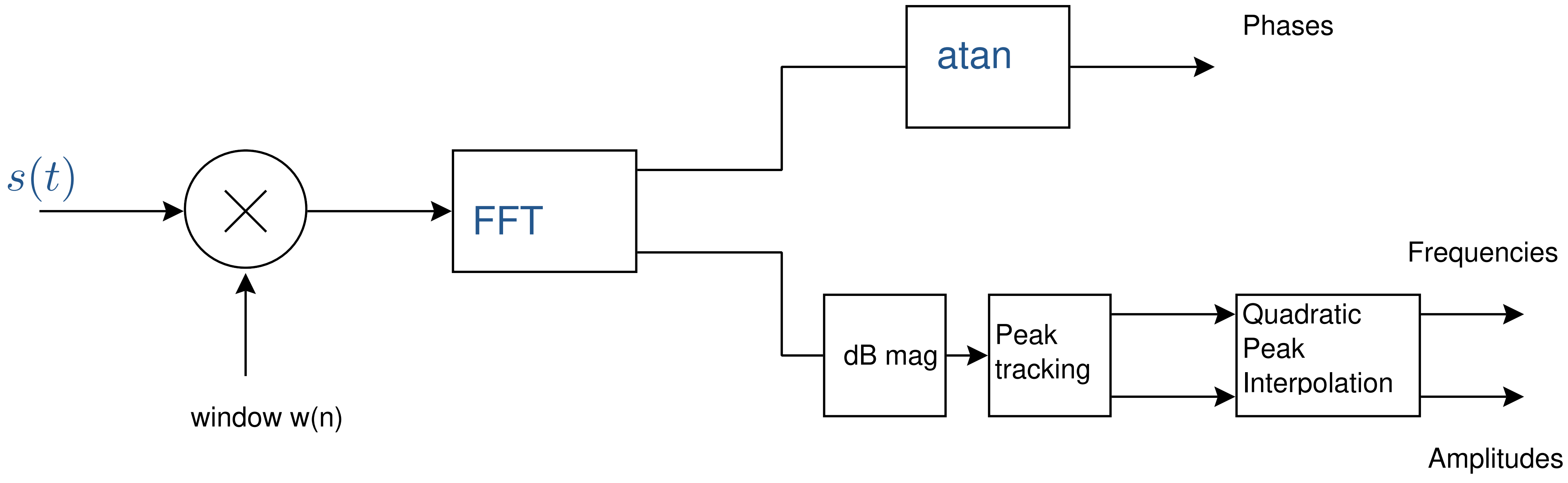
FM Voice

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



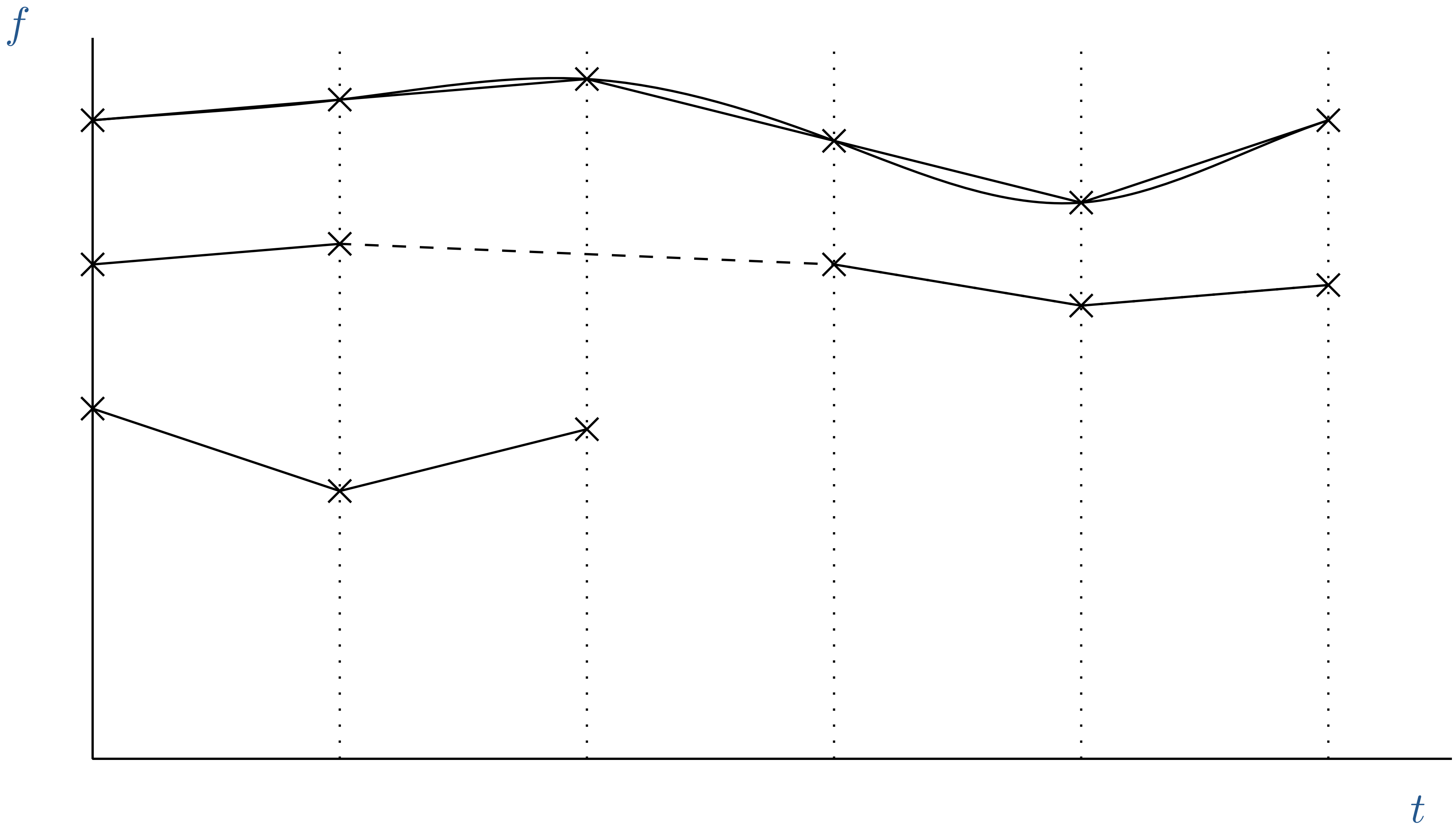
Sinusoidal Modeling Synthesis (1988)

Tracking Spectral Peaks in the Short-Time Fourier Transform

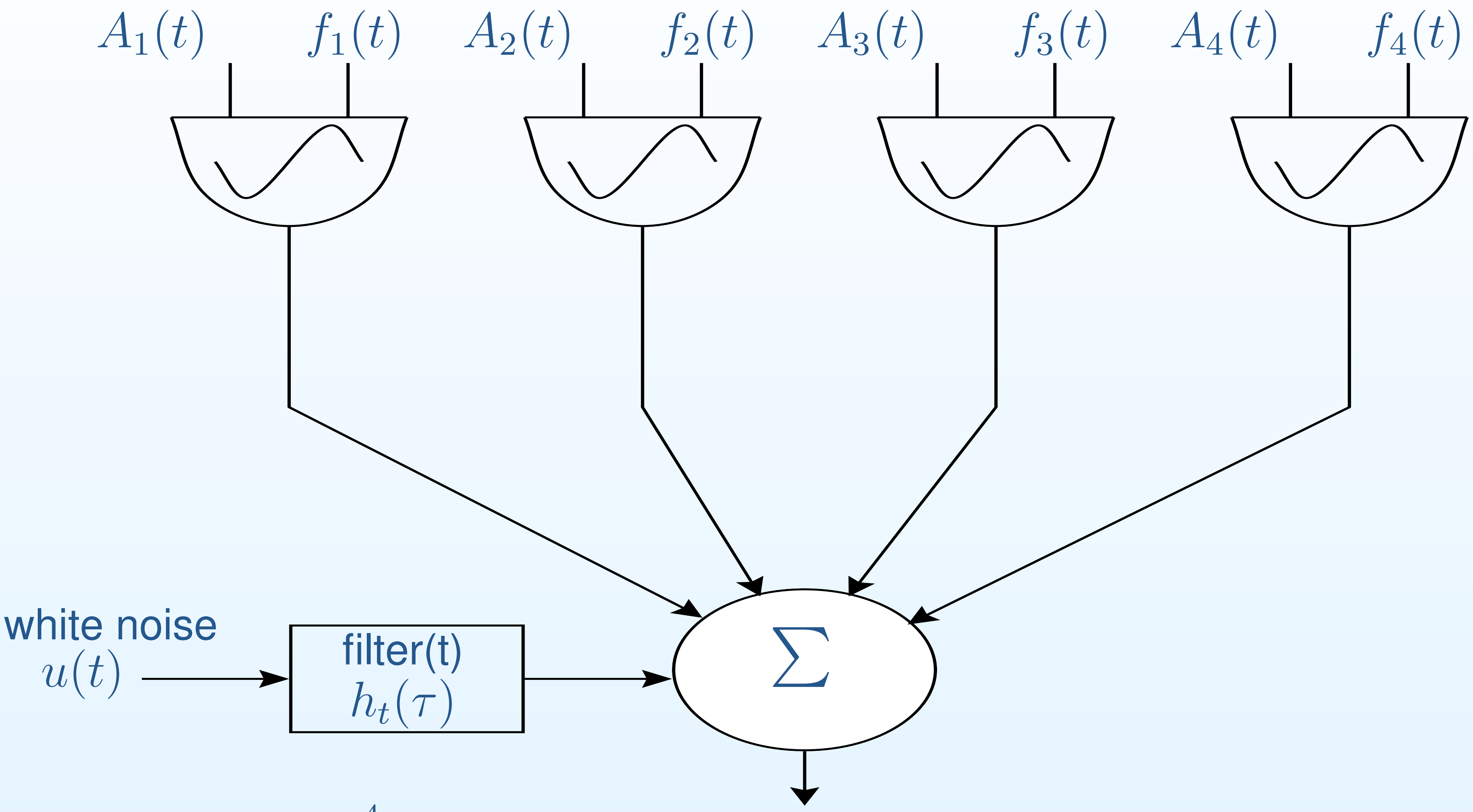


- STFT peak tracking at CCRMA: mid-1980s (PARSHL program)
- Motivated by vocoder analysis of piano tones
- Influences: STFT (Allen and Rabiner 1977), ADEC (1977), MAPLE (1979)
- Independently developed for speech coding by McAulay and Quatieri at Lincoln Labs (1985)

Example Spectral Trajectories



Parametric Spectral Modeling (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 simple 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)

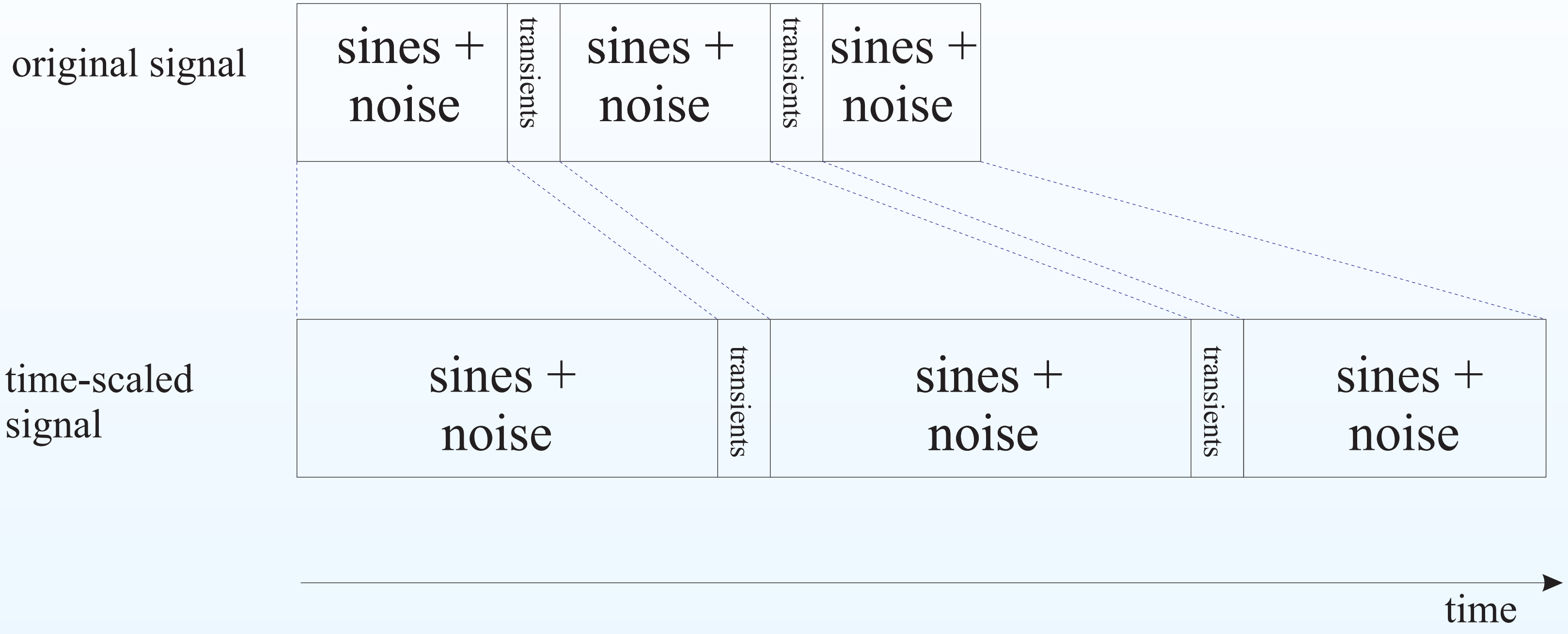
Why Model Transients Separately?

- 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)

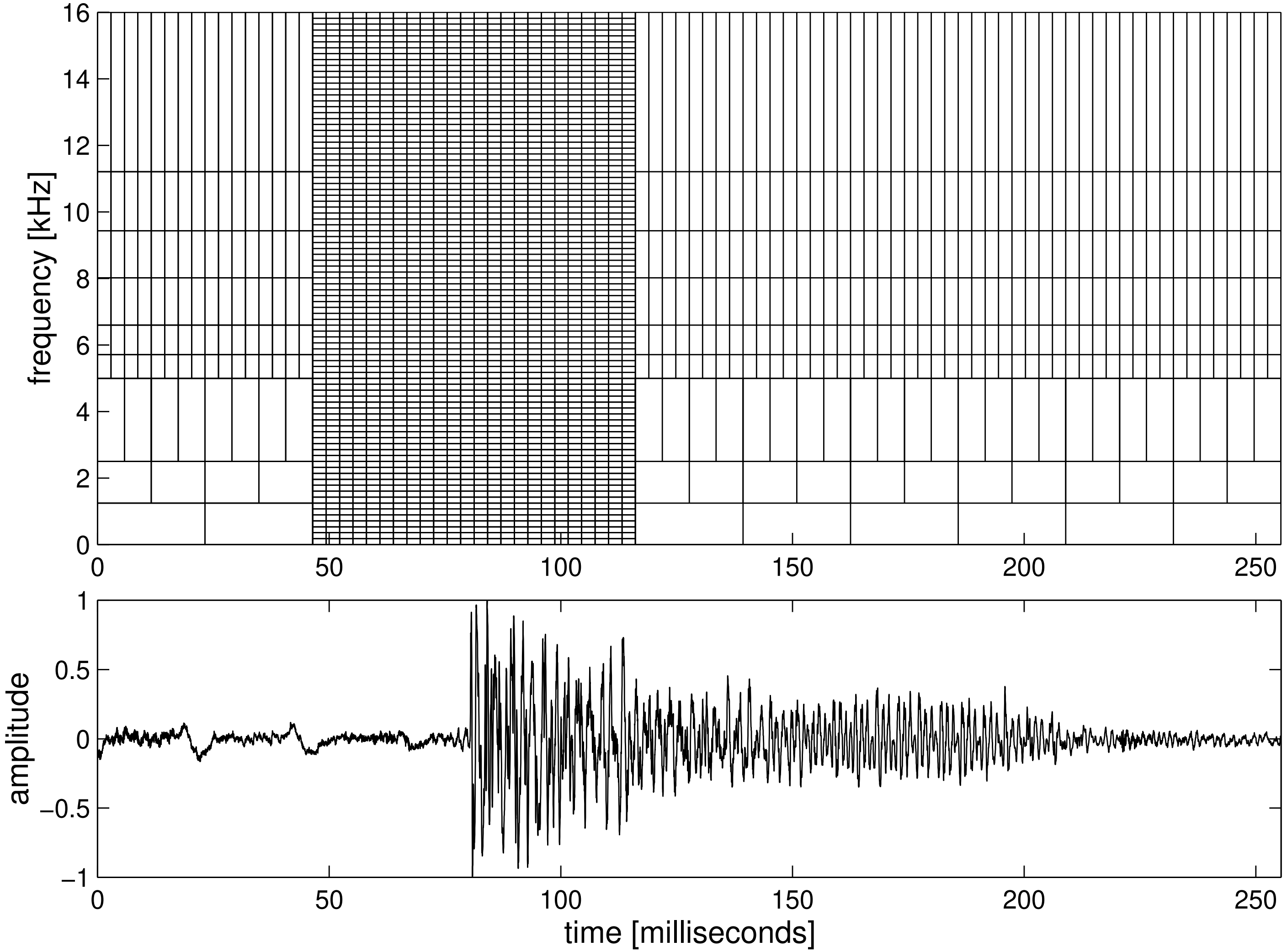
Time Scale Modification of Sines + Noise + Transients Models



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

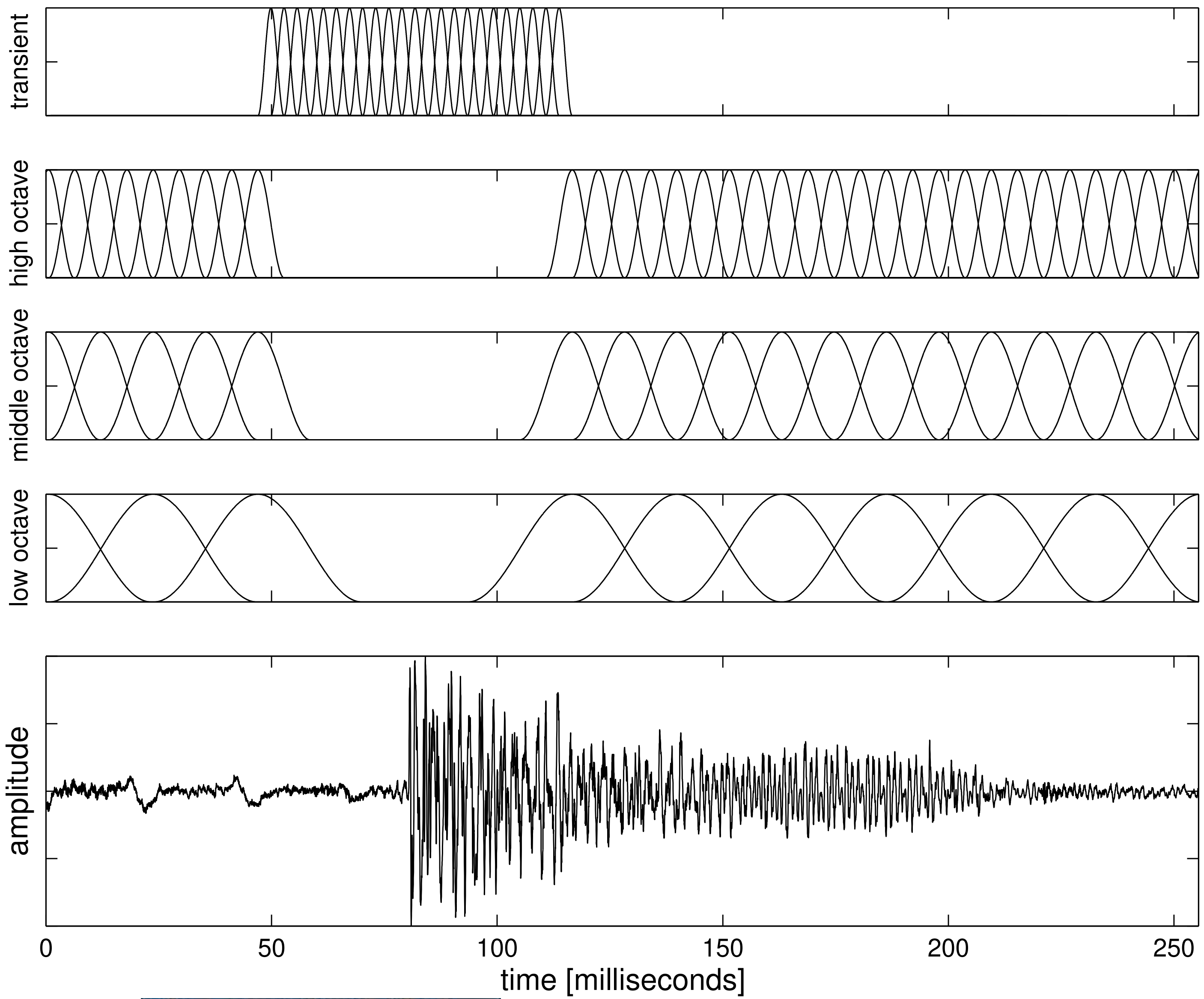
- Transients are *translated* in time
- Sinusoidal envelopes are *scaled* in time
- Noise-filter envelopes also *scaled* in time
- Dual of TSM is *frequency scaling*

Sines + Noise + Transients Time-Frequency Map

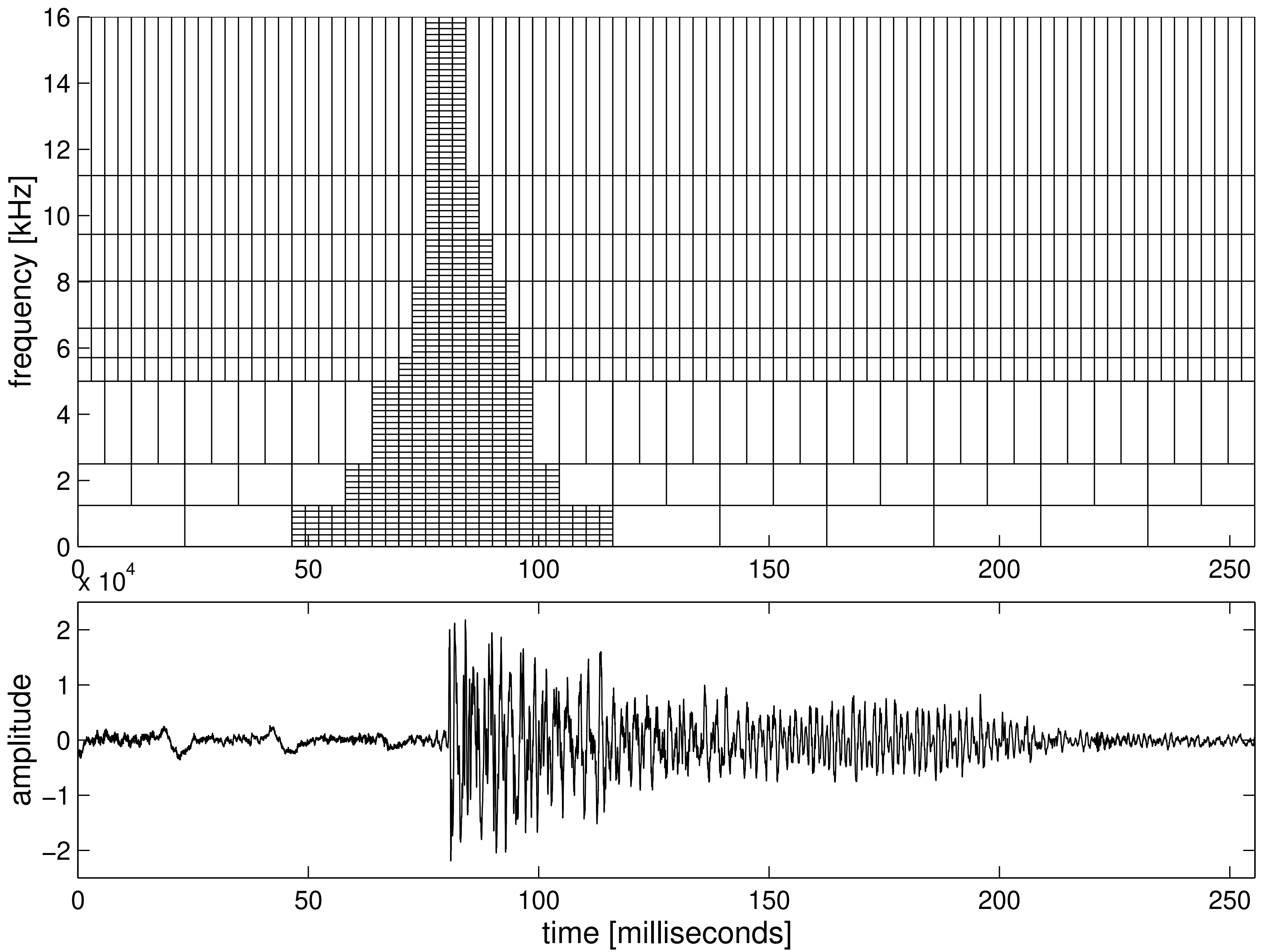


(Levine 1998)

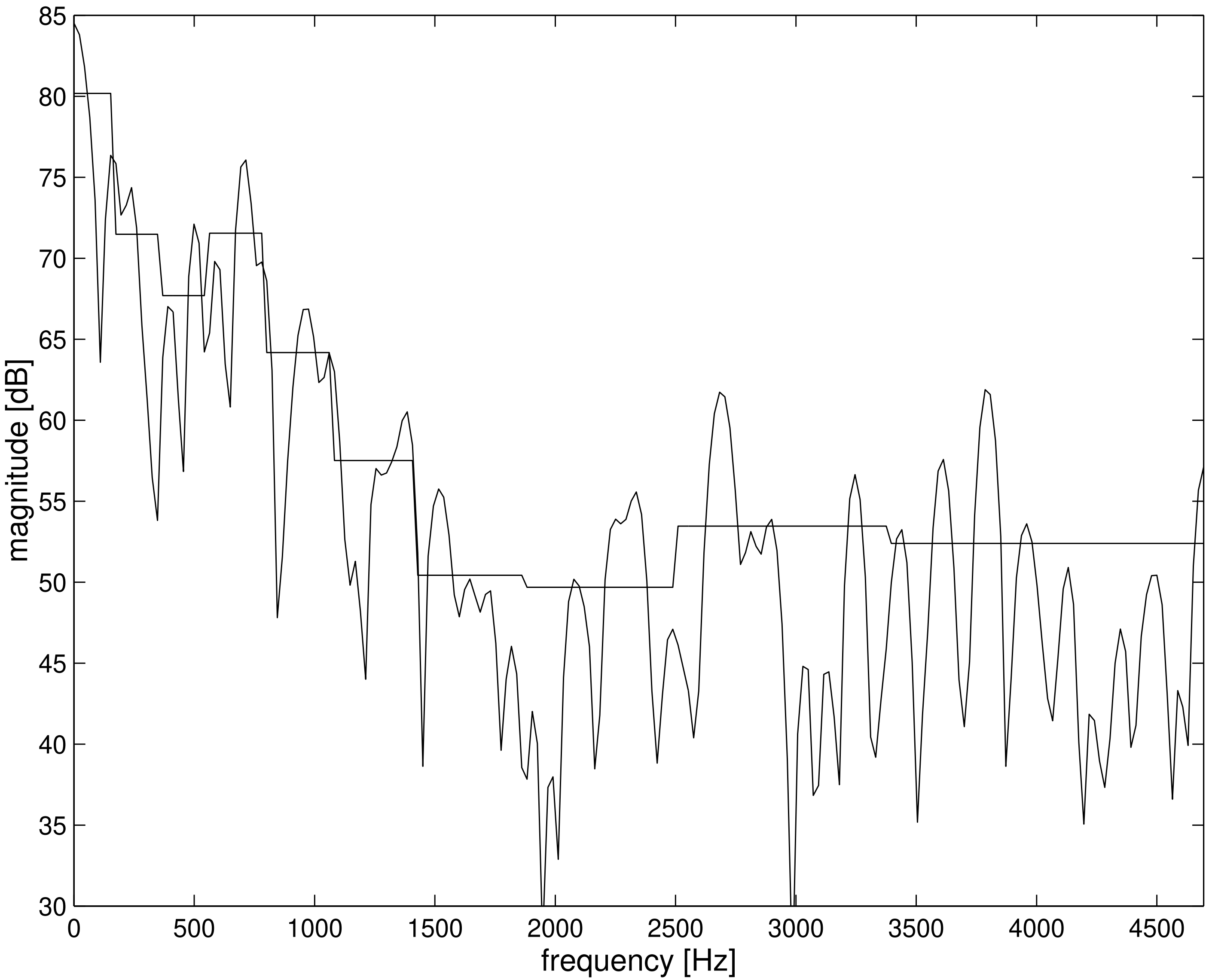
Corresponding Analysis Windows



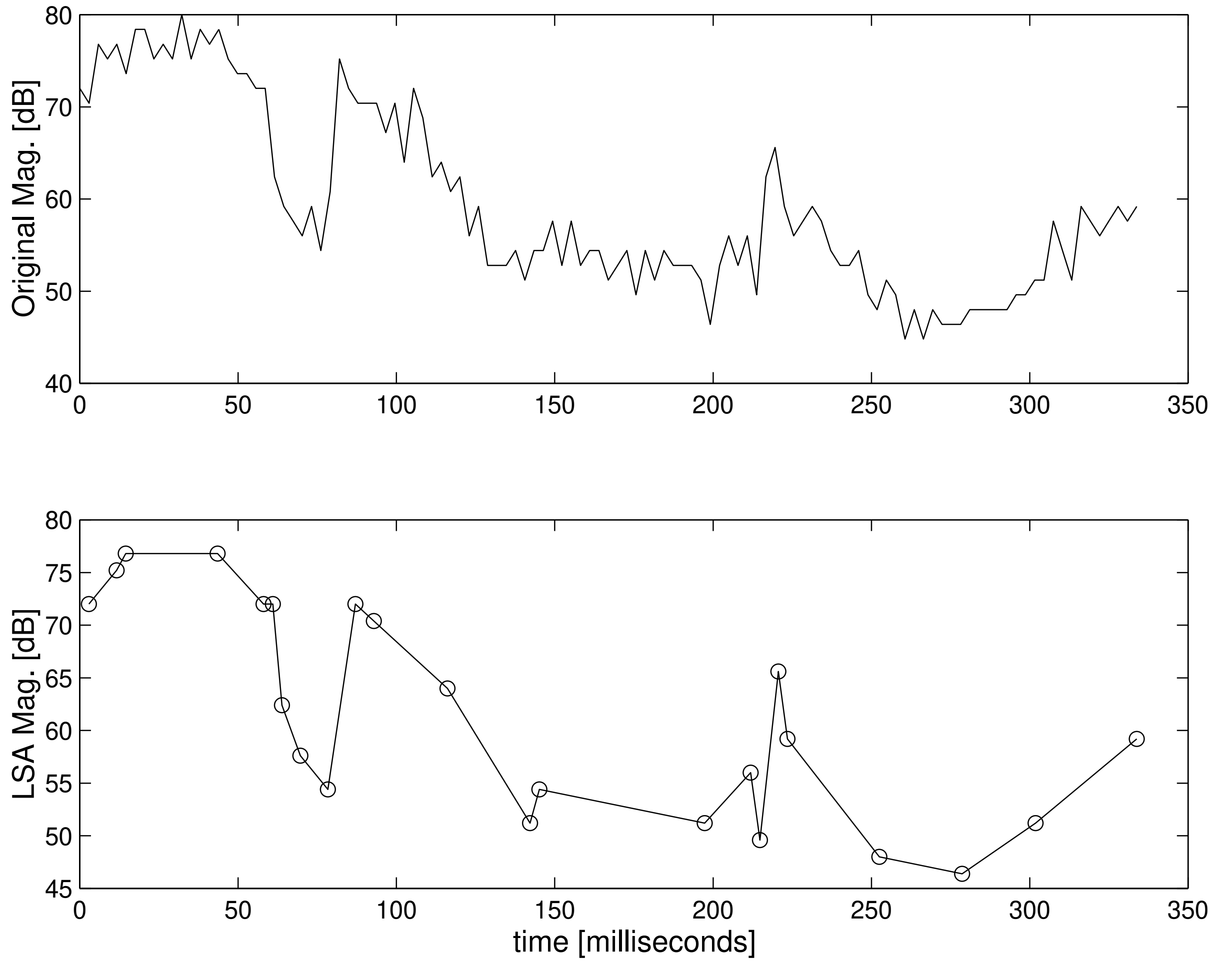
Quasi-Constant-Q (Wavelet) Time-Frequency Map



Bark-Band Noise Modeling (Levine 1998)



Amplitude Envelope for One Noise Band



For more information, see Scott Levine's thesis.[1]

[1] `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`)

"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)

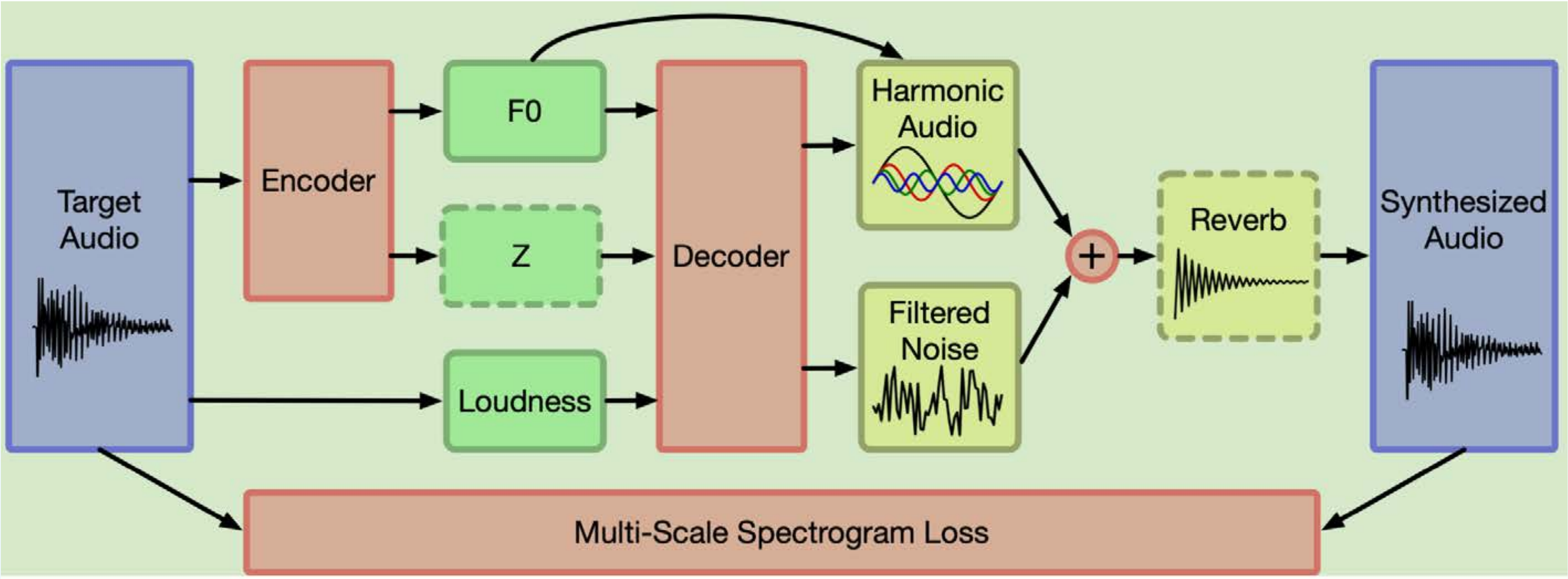
- S+N+T time-scale factors [2.0, 1.6, 1.2, 1.0, 0.8, 0.6, 0.5]

S+N+T Pitch Shifting (timing unchanged)

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

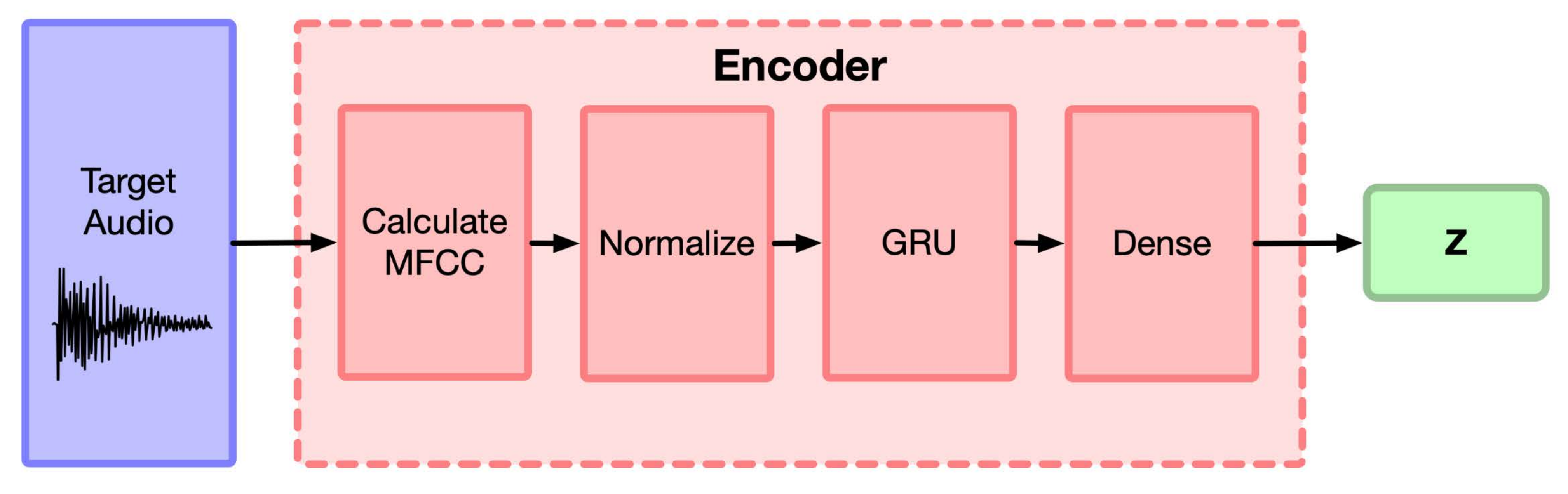
Differentiable DSP (2019)

Differentiable DSP (AI Meets Sines+Noise Synthesis



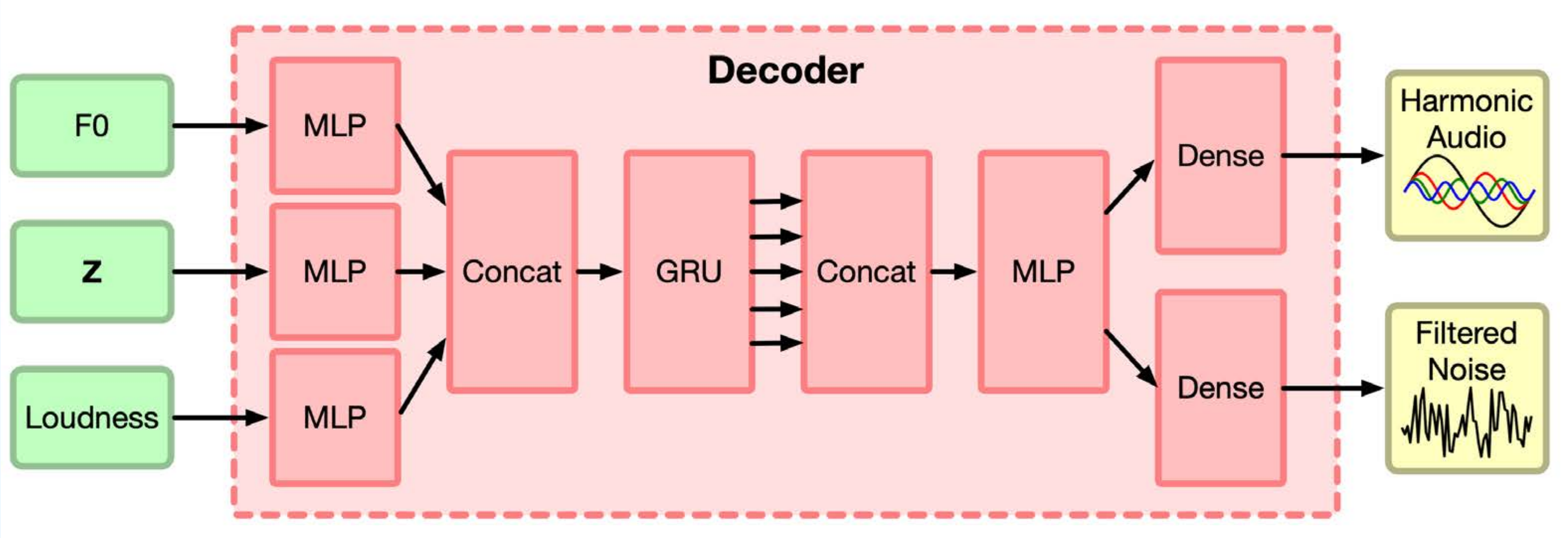
- Jesse Engel et al. at Google Magenta Group
- Neural network analysis/synthesis for *differentiable signal models*
- *Additive Synthesis* example:
  - Loudness normalized by A-weighted log-power spectrum
  - Fundamental Frequency F0 from pretrained CREPE pitch detector
  - Timbre vector Z from *autoencoder*
  - Timbre vector decodes to sinusoidal amplitude trajectories

DDSP Encoder



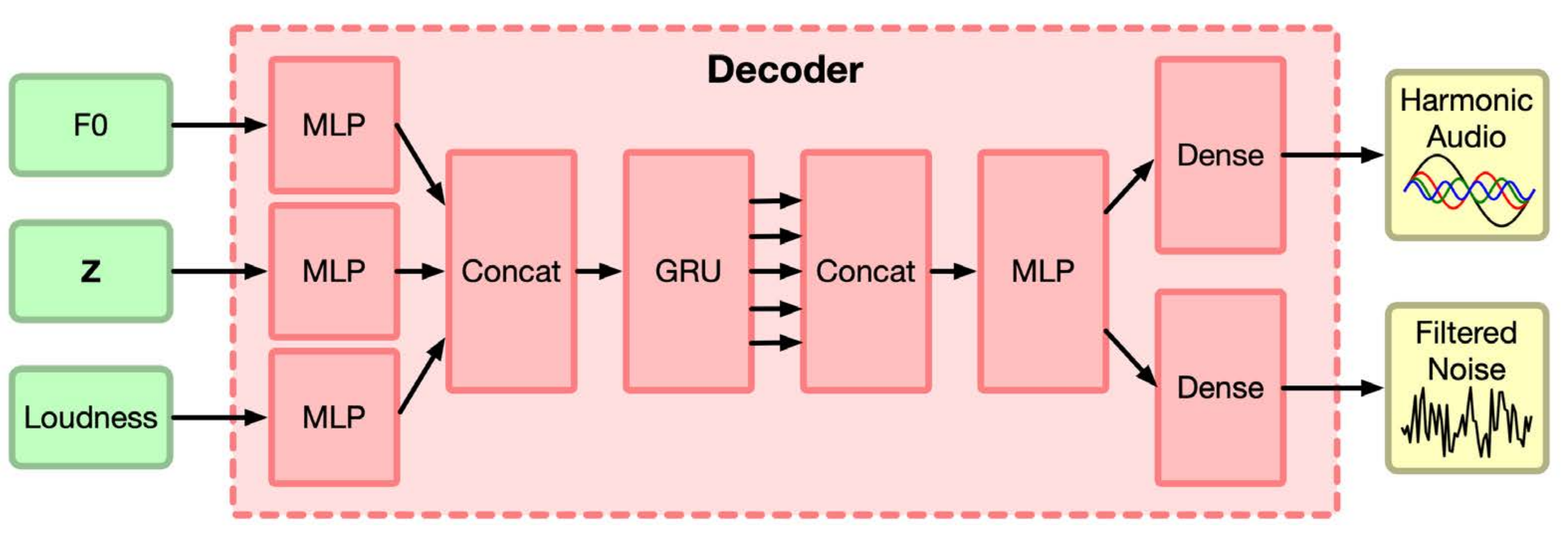
- Loudness and F0 of Target Audio have been normalized away
- MFCC = Mel Frequency Cepstral Coefficients
- GRU = Gated Recurrent Unit (Cho 2014) - similar to LSTM = Long/Short-Term Memory
- Dense = Fully Connected Linear Deep Neural Net (512-to-16 compression step)
- F0 and Loudness normalization leave only *timbre* to be encoded

DDSP Decoder



- MLP = Multi-Layer Perceptron (classical neural network)
- 250 time steps (frames) included
- Output is additive synthesis parameters (sines + filtered noise)

DDSP MLP



- RELU = Rectified Linear Unit (half-wave rectifier)
- 3 layers and 512 Units
- Entire model is differentiable end to end, so back-propagation can optimize everything together (ADAM optimizer used)
- Optimization is generally Stochastic Gradient Descent

Audio in Generative AI

Recent AI Music Audio Generation Timeline

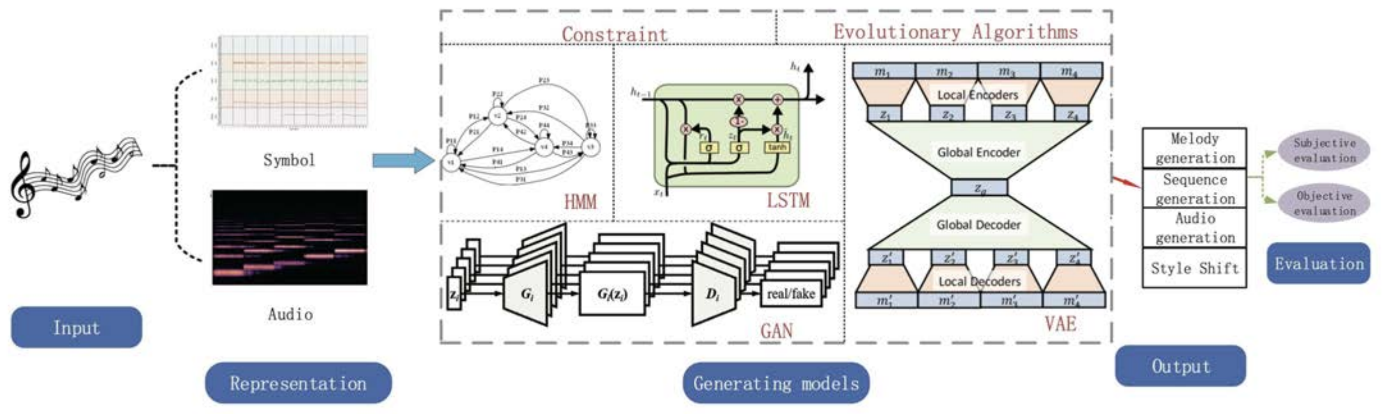
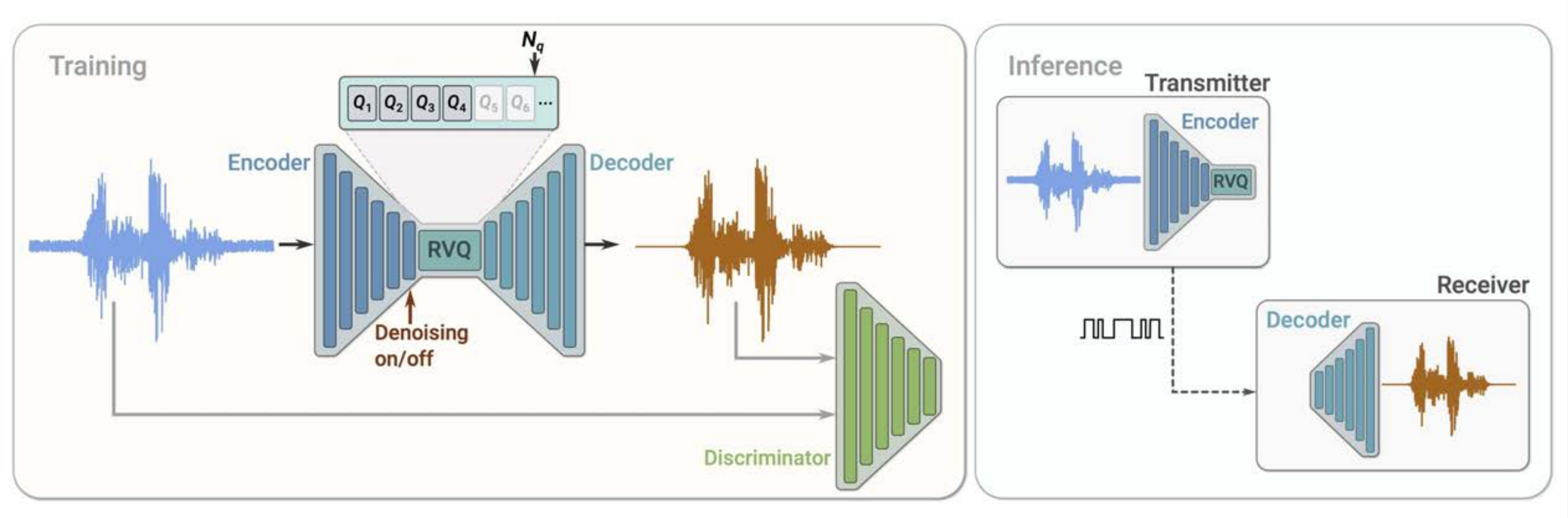


Fig. 1: Overview of music generation system

From [Review] below

- 2016 - WaveNet: “A Generative Model for Raw Audio”
- 2017 - SampleRNN: “Generating Albums with SampleRNN . . . ” [Dadabots]
- 2019 - MusicVAE: “A Hier. Latent Vector Model for Learning L.T. Structure in Music”
- 2020 - Jukebox: “A Generative Model for Music”
- 2022 - [Review]: “A Review of Intelligent Music Generation Systems”
- 2022 - “AudioLM: a Language Modeling Approach to Audio Generation”
- 2023 - “MusicLM: Generating Music From Text” [Jan]
- 2023 - MusicGen: “Simple and Controllable Music Generation” [Jun]

AI Audio Codec Timeline



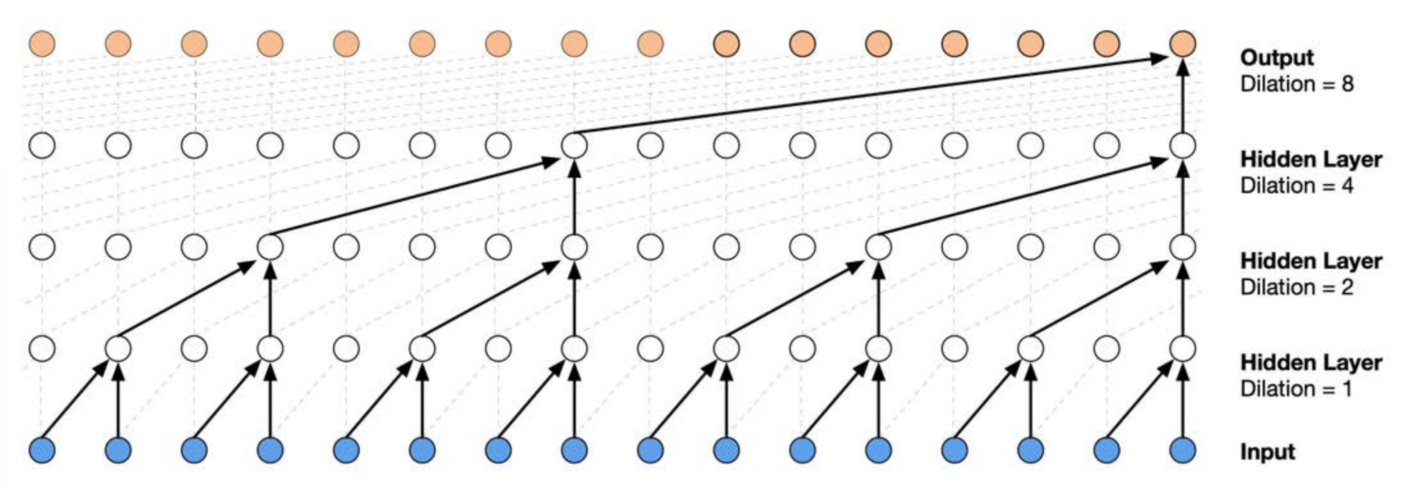
SoundStream: https://arxiv.org/abs/2107.03312

- 1984 - Vector Quantization (VQ)
- 2013 - Variational AutoEncoder (VAE)
- 2018 - VQ-VAE: "Neural Discrete Representation Learning"
- 2021 - "SoundStream: An End-to-End Neural Audio Codec"
- 2022 - EnCodec: "High Fidelity Neural Audio Compression"
- 2023 - "SoundStorm: Efficient Parallel Audio Generation" [May]
- 2023 - Descript: "High-Fidelity Audio Compression with Improved RVQGAN" [Jun]

Will Spectral Modeling Continue?

WaveNet (van den Oord et al., 2016) “A Generative Model for Raw Audio”

WaveNet achieved superior text-to-speech (TTS) by predicting *time samples* autoregressively (no spectral representations)

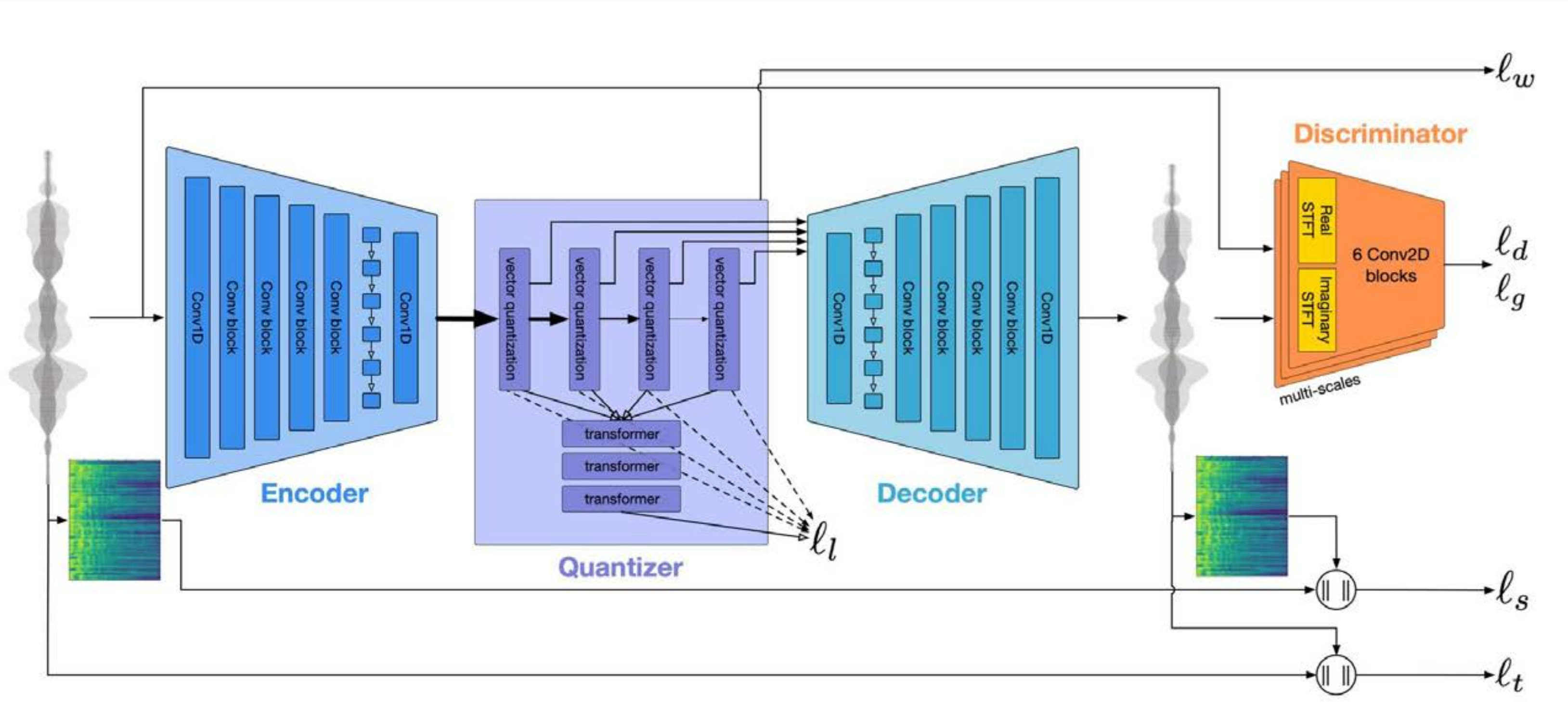


Training WaveNet’s nonlinear (mu-law) sample predictor.

“Traditionally, speech recognition research has largely focused on using log mel-filterbank energies or mel-frequency cepstral coefficients (MFCCs), but has been moving to **raw audio** recently.”

Descript (2023) "High-Fidelity Audio Compression ..."

Based on **SoundStream** (2021), **EnCodec** (2022), and others,
Descript (2023) achieves state-of-the-art performance on benchmarks for AI audio codecs:



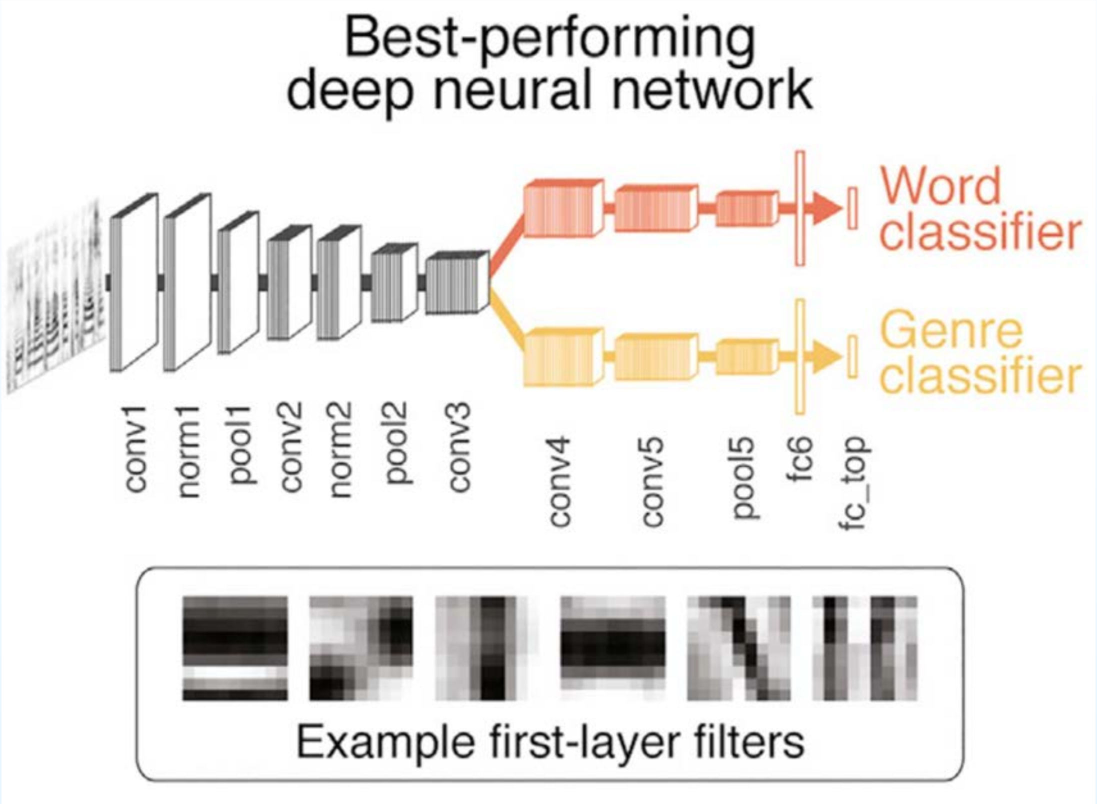
EnCodec

- Time-domain *convolutional encoder-decoder* architecture
- Time-domain *multi-period discriminator* loss
- Frequency-domain discriminator-loss based on *complex multiresolution STFT*
- Frequency-domain *multiresolution mel-spectrogram reconstruction loss* (L1 norm)

Spectral Methods in Recent AI Audio Systems

- **SoundStream** (2021), **EnCodec** (2022), and **Descript** (2023) show a progression from end-to-end learning (**WaveNet** 2016) to the use of *multiresolution spectral reconstruction and adversarial losses*
  - Learning such auditory loss functions end-to-end is *much* more work
  - The audio coder/decoder itself is learned end-to-end: "neural concatenative synthesis"
- Spectra are also pushing back into the *decoder:*
  - **ISTFTNet** (2023): Replaces last two upsample blocks of **HiFi-GAN** with *iSTFT*
  - **Vocos** (2023): Replaces *whole decoder* by iSTFT ⇒ model = STFT coefficients ⇒ Matches SOTA using an *order of magnitude less computing*

McDermott Lab Example: Kell et al., Neuron 2018



- *Cochleagram* presented as an image input to the convolutional neural network (CNN)
- For word recognition and music genre classification, this 12-layer network emerged as best (out of 180 tried)
- The 1st 7 *shared* layers were found to perform comparably with separate 12-layer networks
- Performs at human level, with similar errors
- Predicts fMRI responses in the audio cortex
- Outperforms previous spectrotemporal filter models of the audio cortex
- Cochleagram required for predicting human responses
- Perhaps the cochleagram can be replaced by vastly more natural data and training (evolution-equivalent)

F0 Estimation Reinvents Auditory Filter Bank

Verma and Schafer, Interspeech 2016

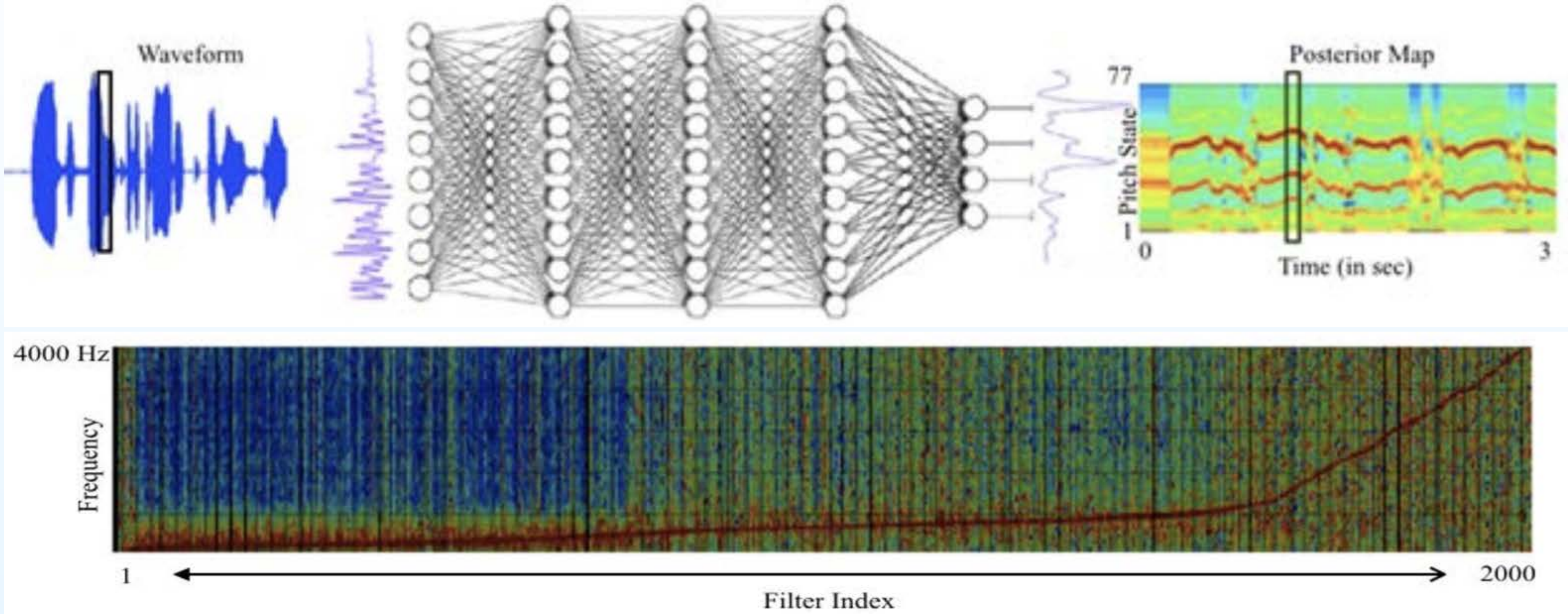


Figure 4: *Frequency response of the learned filters in the first layer sorted according to the highest peak for TIMIT. Red denotes high values whereas blue represents smaller value.*

- Audio filter bank *learned* in the first layer for the F0-estimation task
- Filter bands more dense in the F0 range
- To do: Replace or prepend first layer with a *pre-structured auditory filter bank* having a *differentiable and convex* parameterization, allowing it to be *data* and *task adaptive*

Spectral Representations Continue to be Useful

- Models of hearing (e.g., *Josh McDermott*'s Computational Cognitive Neuroscience lab)
- High-quality audio compression (e.g., **MPEG AAC** (`.m4a`))
- *Reconstruction loss* for audio models and tasks (e.g., **Descript 2023**)
- *Adversarial loss* for audio perception (e.g., **Descript 2023**)
- Faster vocoders (e.g., **Vocos** 2023)
- Intuitive and far more efficient targets for *reverse diffusion* (e.g., **Riffusion** 2022)
- Audio generation *feature-based controls* (e.g., **SingSong** 2023)
- *Human-editable* outputs (e.g., **Anticipatory Music Transformer** 2023)

End-to-End $\rightarrow$ **Embedded Known DSP** is a natural progression, especially for smaller systems

> Huge *end-to-end systems* trained on "infinite" data are great for establishing state-of-the-art performance, but then we can try to *win back territory* from opaque neural nets using *far more efficient and precise processing* (ideally differentiable), while maintaining comparable quality.

Advantages of Spectral Representations

Spectral representations provide

- “Structural Priors” (Inductive Bias) aligned to **human hearing** (forcing synthesis models to the distribution of natural sounds for humans)
- “Human-Centered Loss Functions” (minimize the errors that humans hear most)

These contributions give reduced

- model size,
- training time,
- data requirements,
- computation,
- fossil fuel consumption $(\ddot\smile)$

Common theme: *Keeping It Human*

Maybe spectral representations no longer needed after the AI Apocolypse?

Key Takeaways

- Spectrum analysis “hardware” evolved in the human ear to extend its frequency response and sensitivity
- Time-frequency distributions are fundamental to human audio/music perception and appreciation
- Musical instruments evolved accordingly
- Audio compression evolved accordingly
- AI can reinvent spectral processing, or we can provide it, saving training
  - “Neural concatenative synthesis” needs frequent phase matching, so time-domain and frequency-domain models are not that different
  - Audio loss functions still employ spectral methods
  - Differentiable DSP adds trainable known structure that will surely keep growing
  - Constraining AI to human ears (e.g., cochleagram front ends) helps to model human audio perception