CTP431- Music and Audio Computing
Sound Synthesis

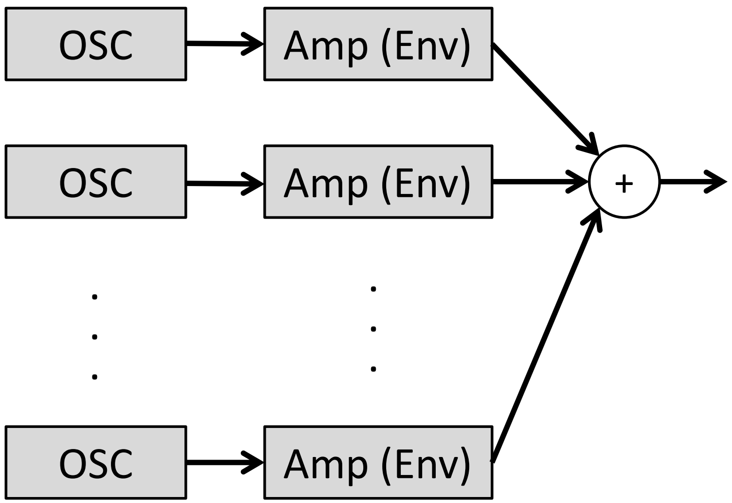
Graduate School of Culture Technology
KAIST
Juhan Nam

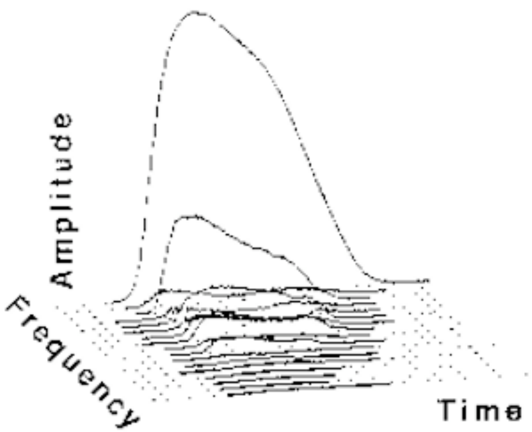
Outlines

- Additive Synthesis
- Subtractive Synthesis
  - Analog synthesizers
  - Singing voice synthesis
- Nonlinear Synthesis
  - Ring modulation / Frequency modulation
  - Wave-shaping
- Physical Modeling
- Sample-based Synthesis

Additive Synthesis

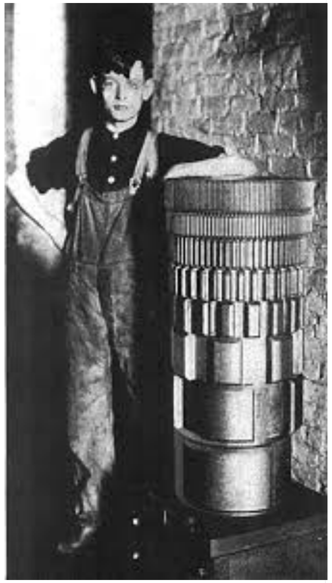
- Synthesize sounds by adding multiple sine oscillators
  - Also called Fourier synthesis

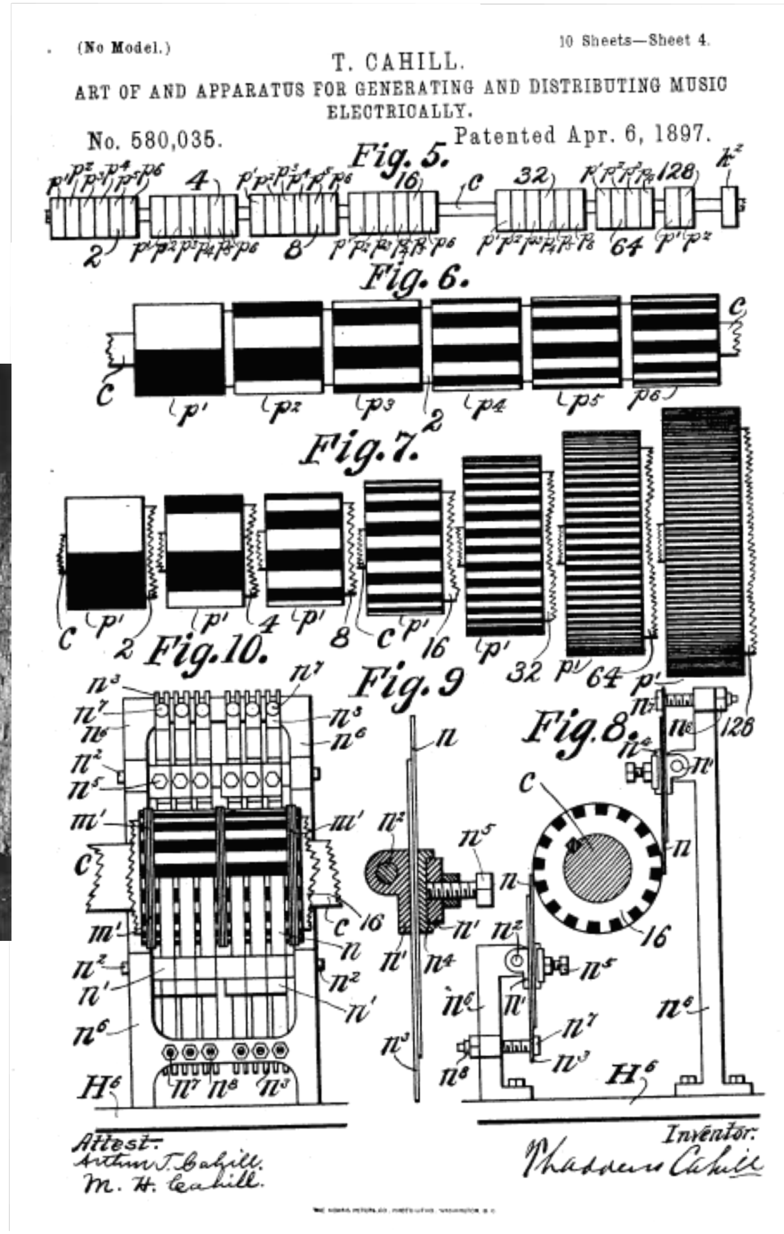




Telharmonium

- The first additive synthesizer using electro-magnetic “tone wheels” (Cahill, 1897)
- Transmitted through telephone lines
  - Subscription only
  - The business failed





Theremin

- A sinusoidal tone generator
- Two antennas are remotely controlled to adjust pitch and volume

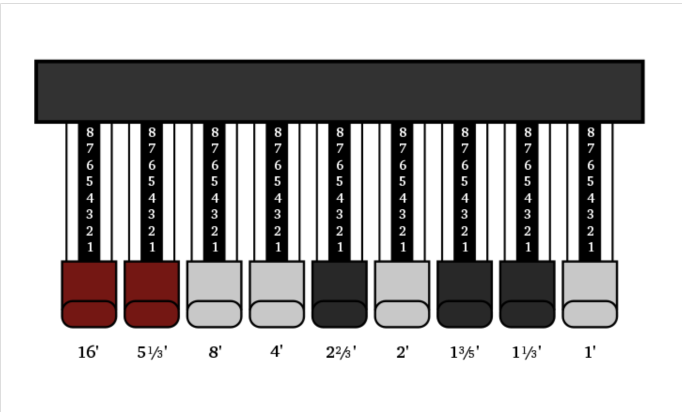
Theremin (by Léon Theremin, 1928)

Theremin (Clara Rockmore)

https://www.youtube.com/watch?v=pSzTPGlNa5U

Hammond Organ

- Drawbars
  - Control the levels of individual tonewheels



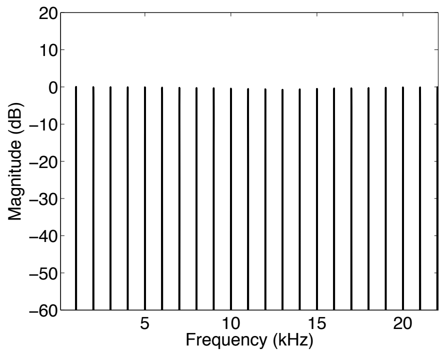
Sound Examples

- Web Audio Demo
  - http://femurdesign.com/theremin/
  - http://www.venlabsla.com/x/additive/additive.html

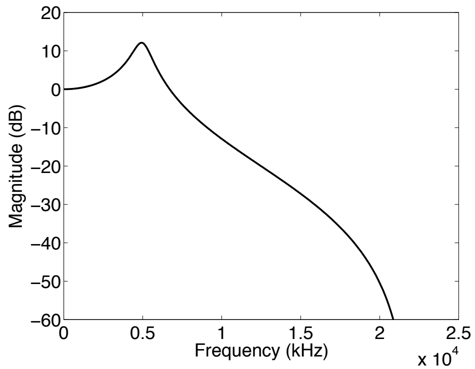
- Examples (instruments)
  - Kurzweil K150
    - https://soundcloud.com/rosst/sets/kurzweil-k150-fs-additive
  - Kawai K5, K5000

Subtractive Synthesis

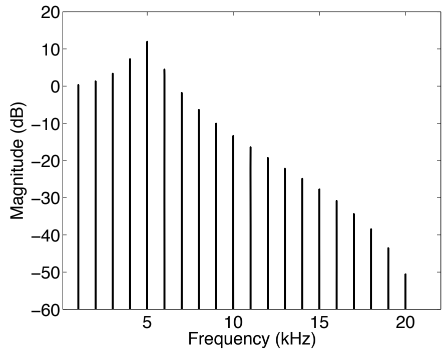
- Synthesize sounds by filtering wide-band oscillators
  - Source-Filter model
  - Examples
    - Analog Synthesizers: oscillators + resonant lowpass filters
    - Voice Synthesizers: glottal pulse train + formant filters



Source

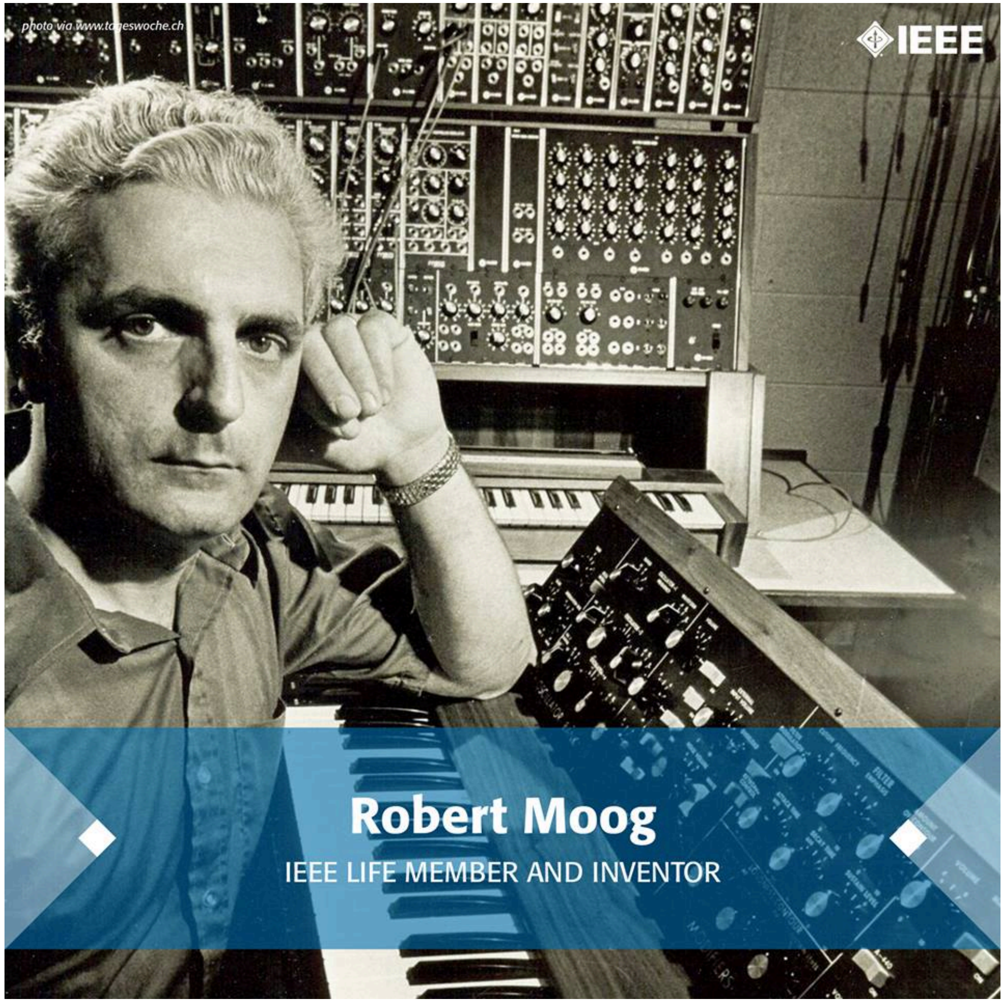


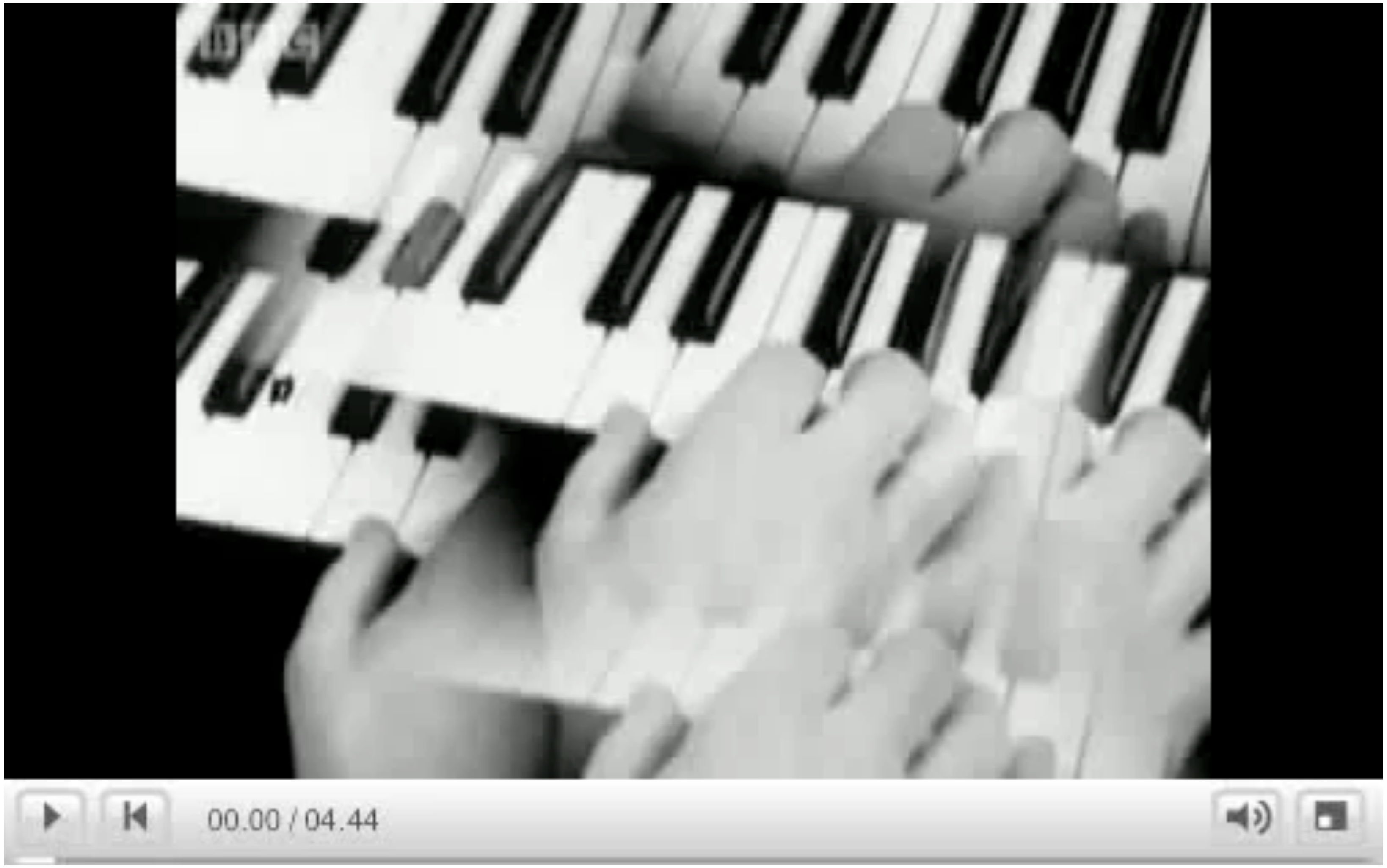
Filter



Filtered Source

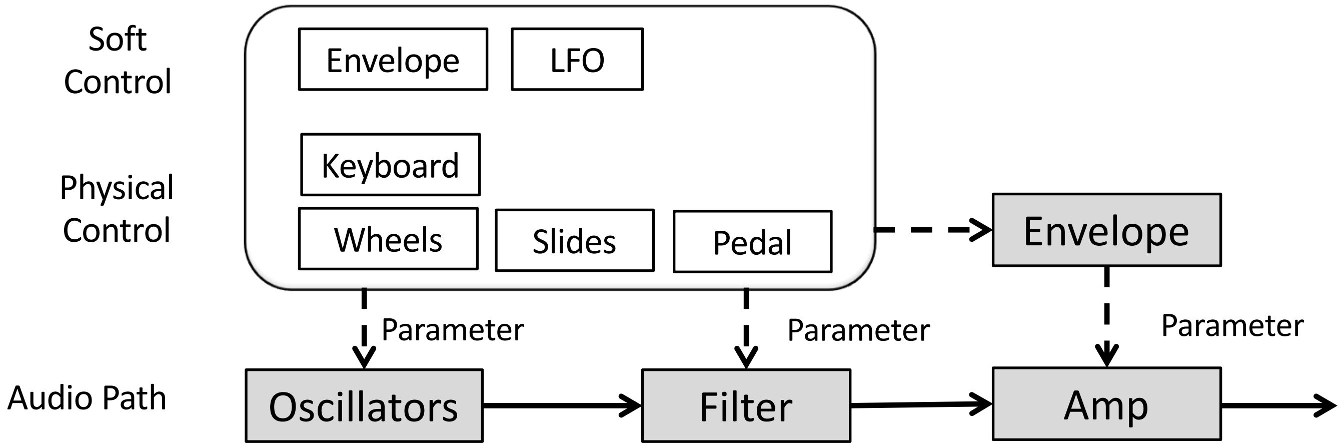
Moog Synthesizers





https://www.youtube.com/watch?v=usl_TvIFtG0

Moog Synthesizers



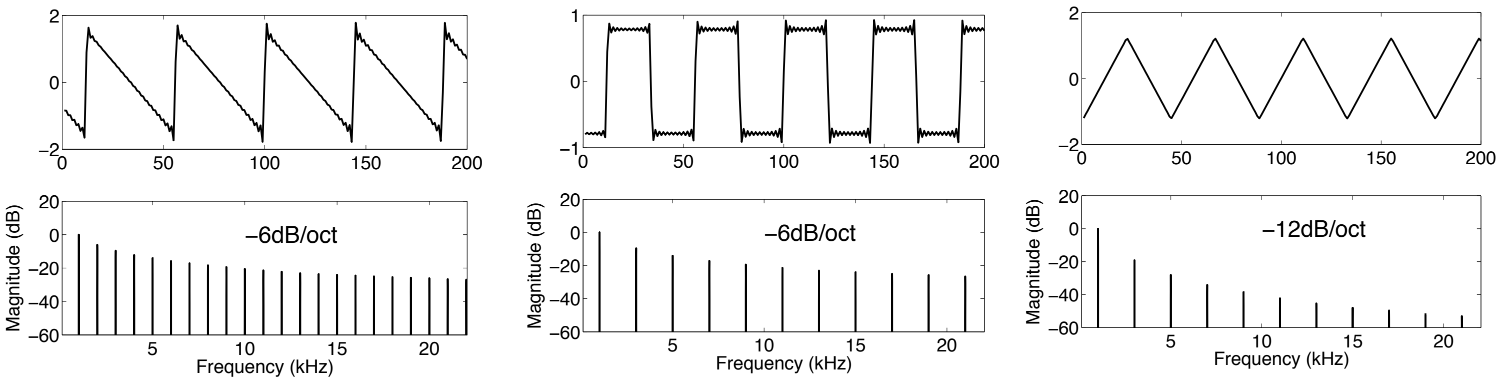
Parameter = $\underline{\text{offset}}$ + $\underline{\text{depth*control}}$

(static value) (dynamic value)

(e.g. filter cut-off frequency)

Oscillators

- Classic waveforms

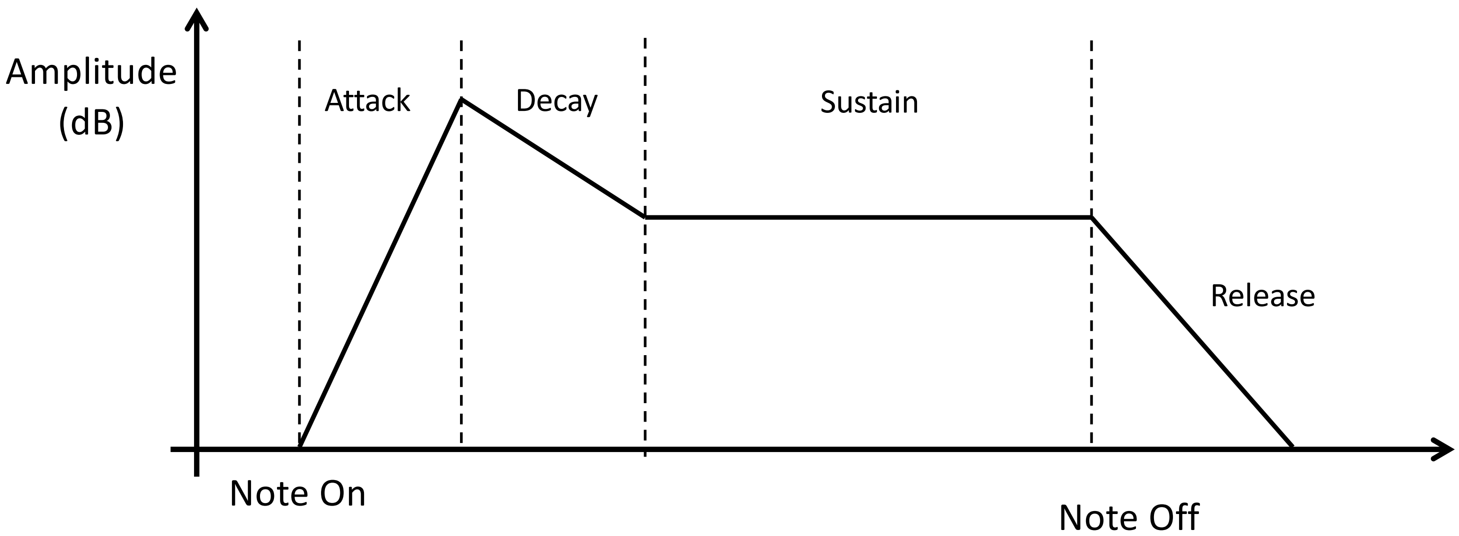


Sawtooth Square Triangular

- Modulation
  - Pulse width modulation
  - Hard-sync
  - More rich harmonics

Amp Envelop Generator

- Amplitude envelope generation
  - ADSR curve: attack, decay, sustain and release
  - Each state has a pair of time and target level



Examples

- Web Audio Demos
  - http://www.google.com/doodles/robert-moogs-78th-birthday
  - http://webaudiodemos.appspot.com/midi-synth/index.html
  - http://aikelab.net/websynth/
  - http://nicroto.github.io/viktor/

- Example Sounds
  - SuperSaw
  - Leads
  - Pad
  - MoogBass
  - 8-Bit sounds: https://www.youtube.com/watch?v=tf0-Rrm9dI0
  - TR-808: https://www.youtube.com/watch?v=YeZZk2czG1c

Modulation Synthesis

- Modulation is originally from communication theory
  - Carrier: channel signal, e.g., radio or TV channel
  - Modulator: information signal, e.g., voice, video

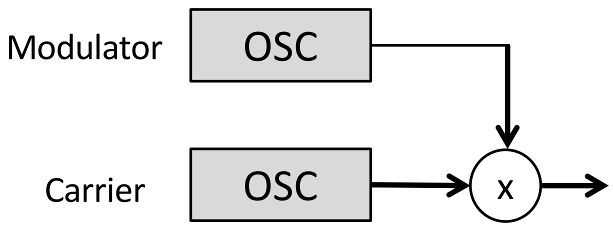
- Decreasing the frequency of carrier to hearing range can be used to synthesize sound

- Types of modulation synthesis
  - Amplitude modulation (or ring modulation)
  - Frequency modulation

- Modulation is non-linear processing
  - Generate new sinusoidal components

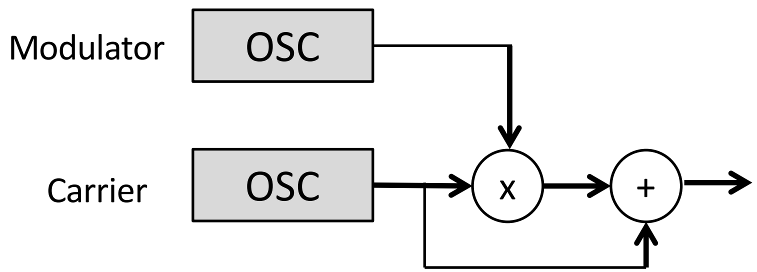
Ring Modulation / Amplitude Modulation

- Change the amplitude of one source with another source
  - Slow change: tremolo
  - Fast change: generate a new tone



$$a_m(t)A_c\cos(2\pi f_c t)$$

Ring Modulation

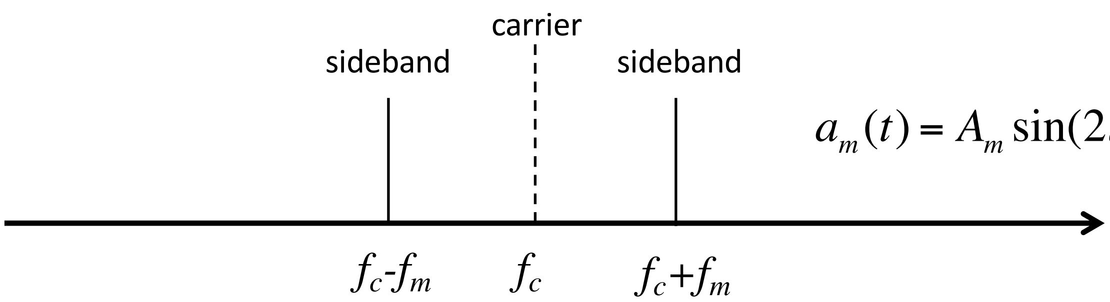


$$(1+a_m(t))A_c\cos(2\pi f_c t)$$

Amplitude Modulation

Ring Modulation / Amplitude Modulation

- Frequency domain
  - Expressed in terms of its sideband frequencies
  - The sum and difference of the two frequencies are obtained according to trigonometric identity
  - If the modulator is a non-sinusoidal tone, a mirrored-spectrum with regard to the carrier frequency is obtained



$$a_m(t) = A_m \sin(2\pi f_m t))$$

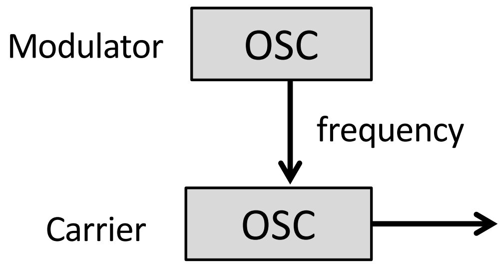
Examples

- Tone generation
  - SawtoothOsc x SineOsc
  - https://www.youtube.com/watch?v=yw7_WQmrzuk

- Ring modulation is often used as an audio effect
  - http://webaudio.prototyping.bbc.co.uk/ring-modulator/

Frequency Modulation

- Change the frequency of one source with another source
  - Slow change: vibrato
  - Fast change: generate a new (and rich) tone
  - Invented by John Chowning in 1973 → Yamaha DX7



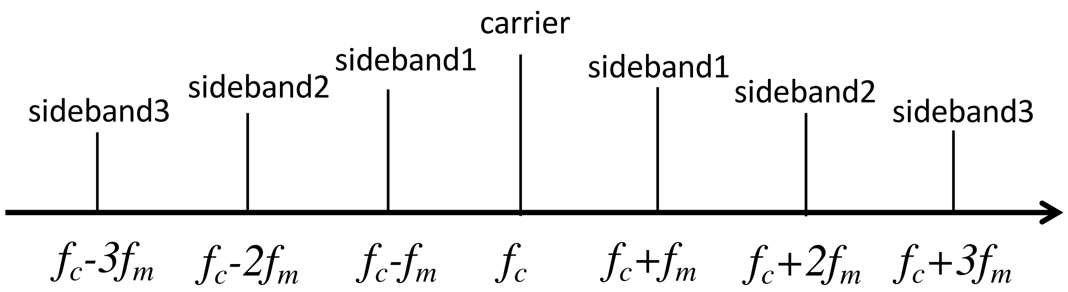
$$A_c \cos(2\pi f_c t + \beta \sin(2\pi f_m t))$$

$$\beta = \frac{A_m}{f_m}$$ Index of modulation

Frequency Modulation

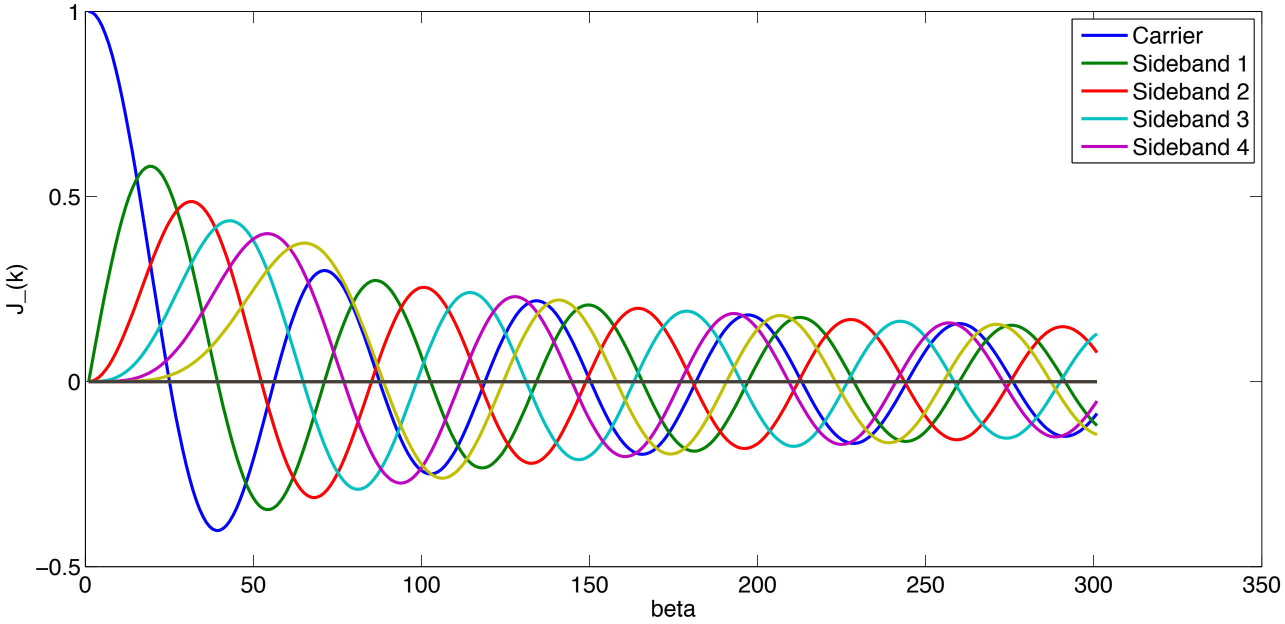
- Frequency Domain
  - Expressed in terms of its sideband frequencies
  - Their amplitudes are determined by the Bessel function
  - The sidebands below 0 Hz or above the Nyquist frequency are folded

$$y(t) = A_c \sum_{k=-\infty}^{k=-\infty} J_k(\beta)\cos(2\pi(f_c + kf_m)t)$$



Bessel Function

$$J_k(\beta) = \sum_{n=0}^{\infty} \frac{(-1)^n (\frac{\beta}{2})^{k+2n}}{n!(n+k)!}$$



Bessel Function

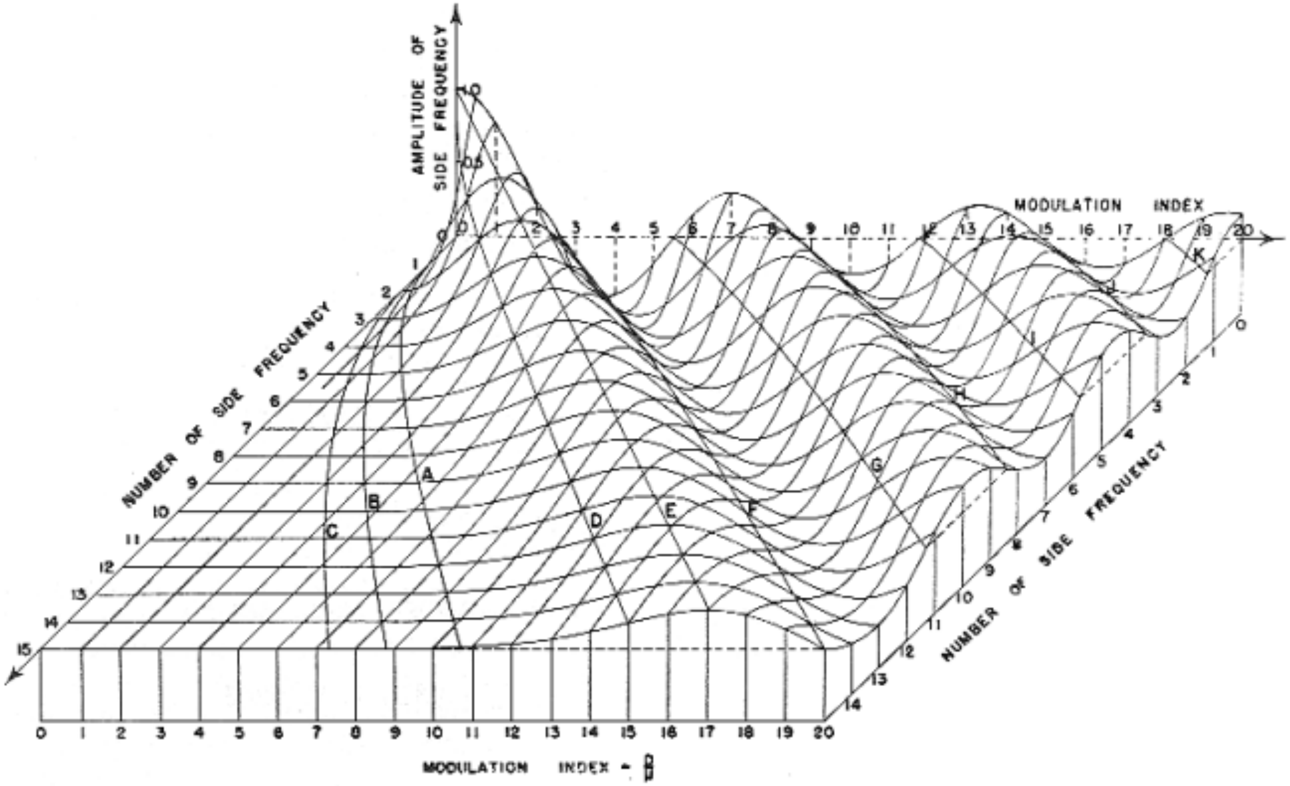
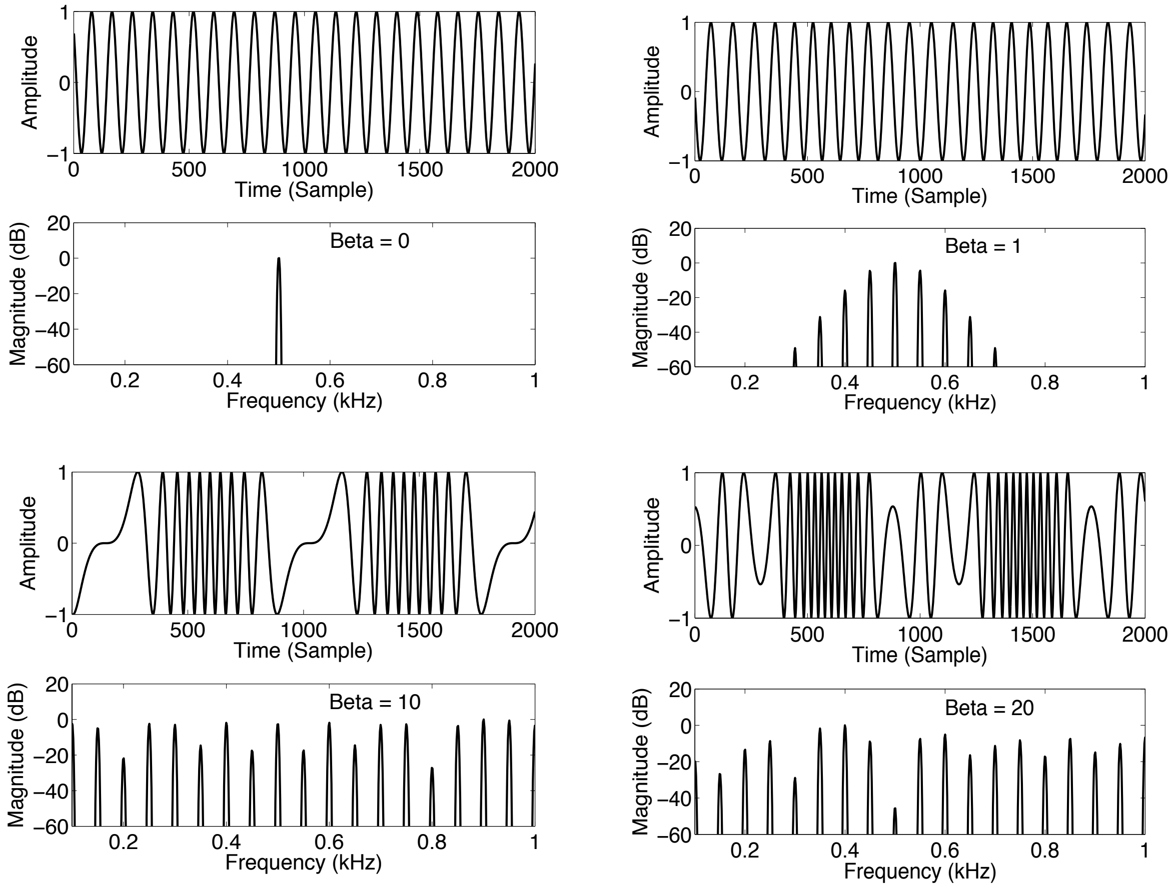
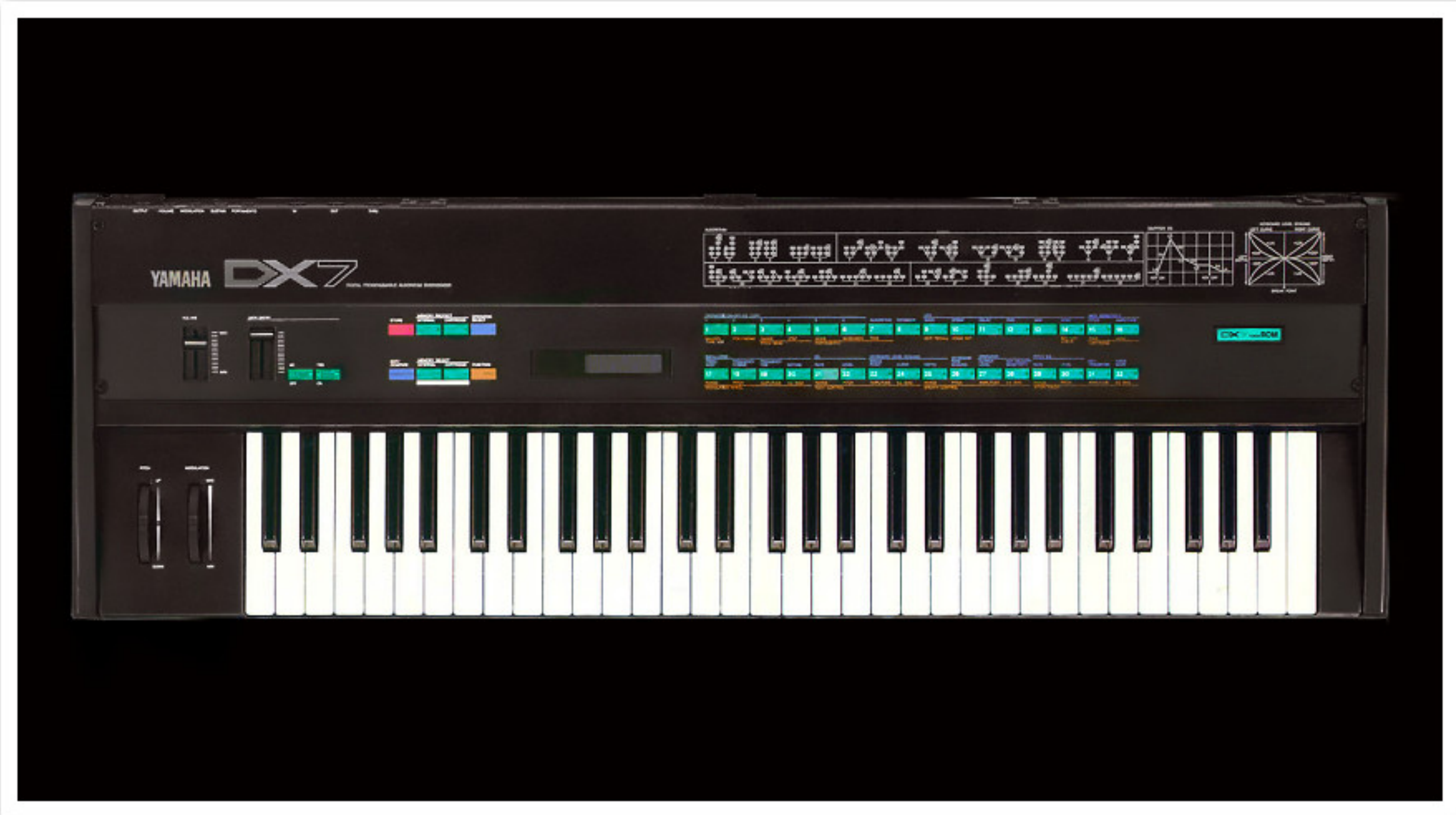


Fig. 4—Side-frequency amplitudes.

The Effect of Modulation Index

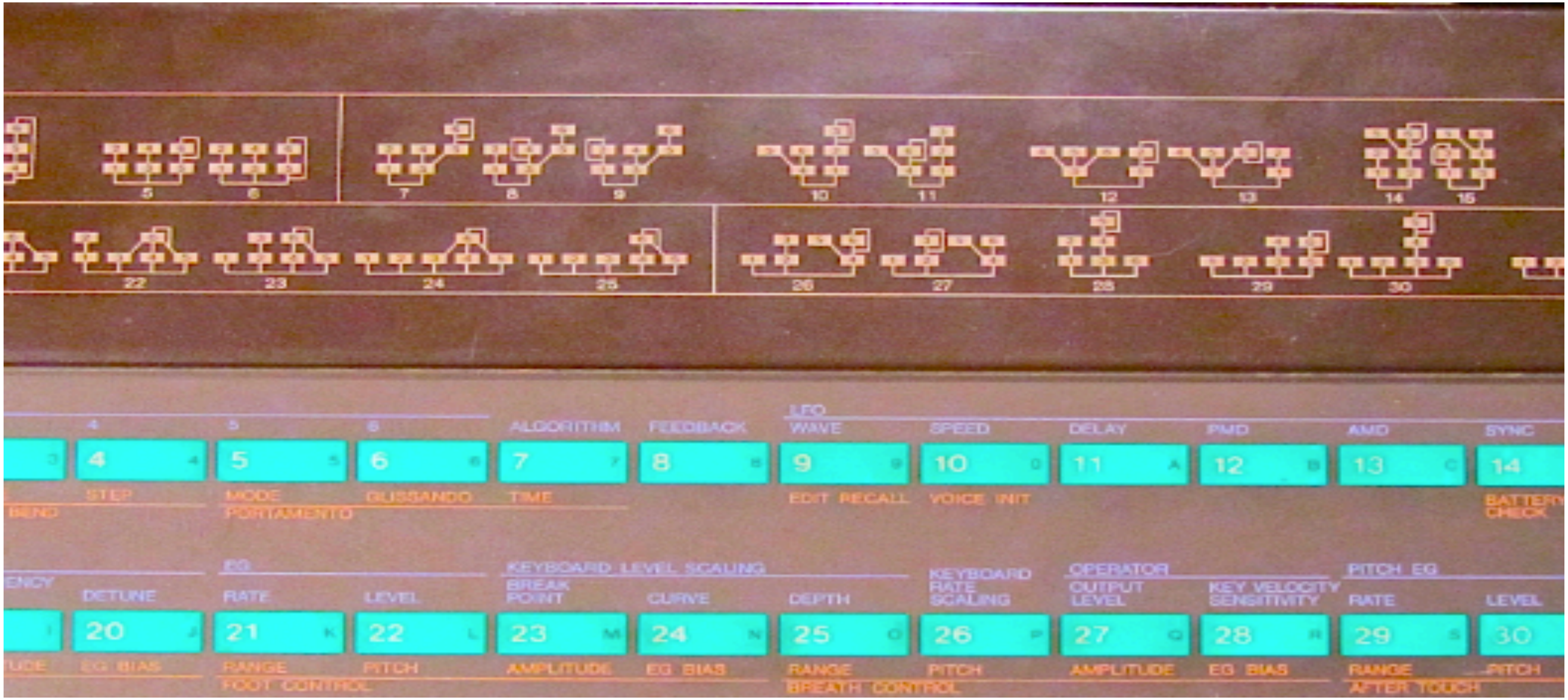


$$f_c = 500,\ \ f_m = 50$$



Yamaha DX7 (1983)

“Algorithms” in DX7



http://www.audiocentralmagazine.com/yamaha-dx-7-riparliamo-di-fm-e-non-solo-seconda-parte/yamaha-dx7-algorithms/

Examples

- Web Audio Demo
  - http://www.taktech.org/takm/WebFMSynth/

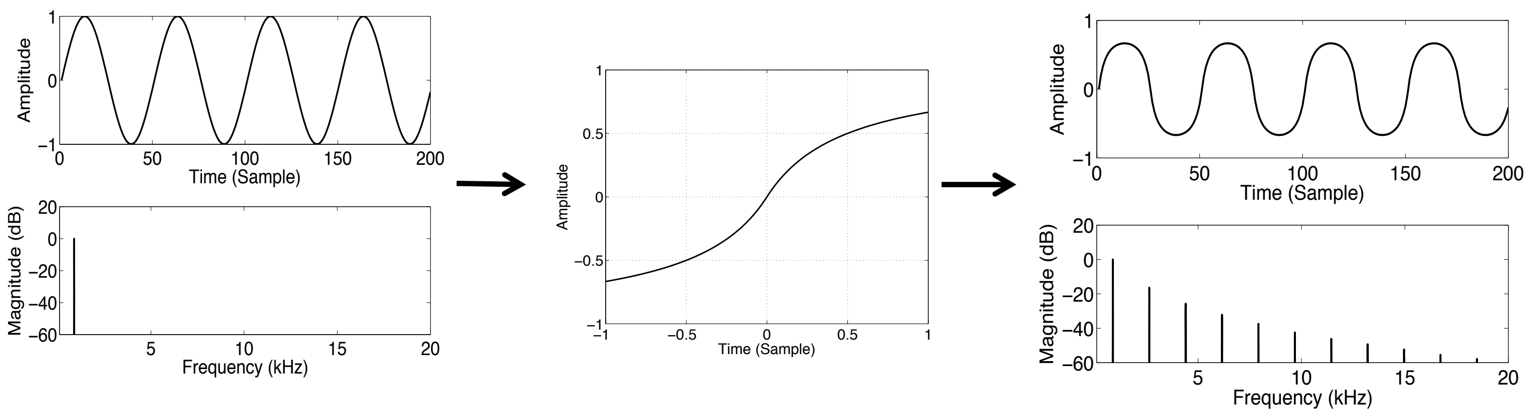
- Sound Examples
  - Bell
  - Wood
  - Brass
  - Electric Piano
  - Vibraphone

Non-linear Synthesis (wave-shaping)

- Generate a rich sound spectrum from a sinusoid using non-linear transfer functions (also called "distortion synthesis")

- Examples of transfer function: y = f(x)
  - $y = 1.5x' - 0.5x'^3$
  - $y = x'/(1+|x'|)$ x'=gx: g correspond to the "gain knob" of the distortion
  - $y = \sin(x')$
  - Chebyshev polynomial: $T_{k+1}(x) = 2xT_k(x)-T_{k-1}(x)$

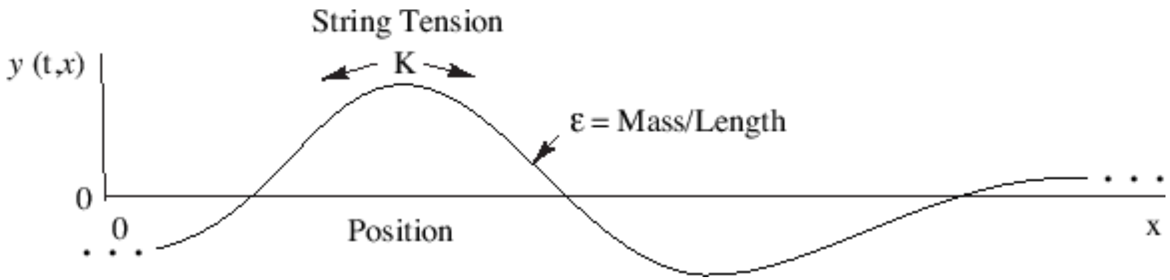
$T_0(x)=1$, $T_1(x)=x$,
$T_2(x)=2x^2-1$, $T_2(x)=4x^3-3x$



Physical Modeling

- Modeling Newton's laws of motion (i.e. $F = ma$) on musical instruments
  – Every instrument have a different model

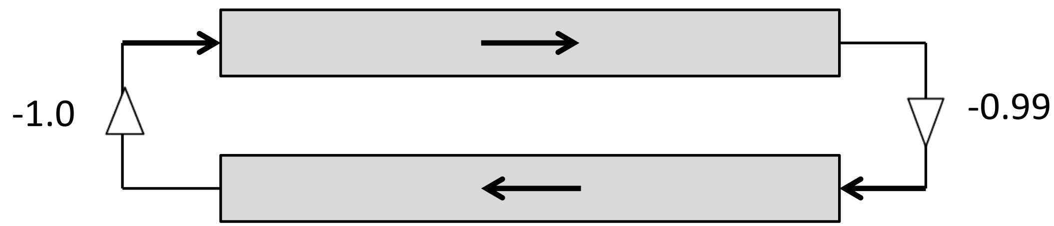
- The ideal string



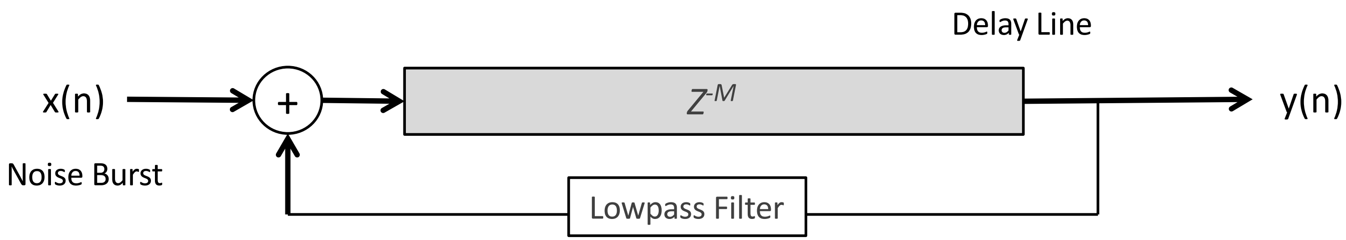
  – Wave equation: $F = ma$ → $K\frac{d^2y}{dx^2} = \varepsilon\frac{d^2y}{dx^2}$ (K: tension, ε: linear mass density)
  – General solution: $y(t,x) = y_L(t - \frac{x}{c}) + y_R(t + \frac{x}{c})$
    →Left-going traveling wave and right-going traveling wave

Physical Modeling

- Waveguide Model
  - With boundary condition (fixed ends)

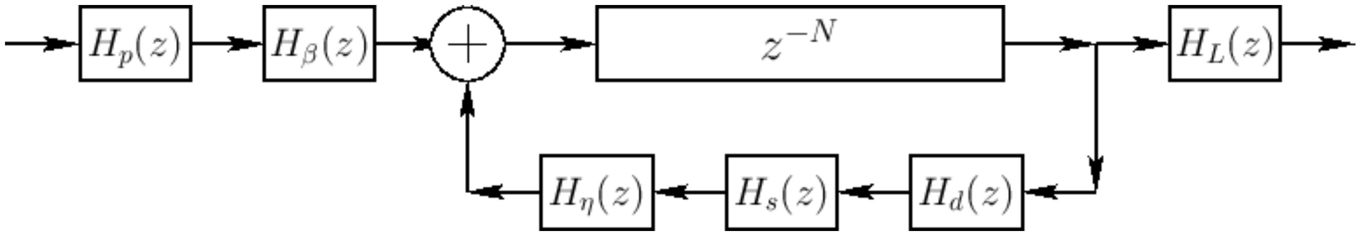


- The Karplus-Strong model



Physical Modeling

- The Extended Karplus -Strong model



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

https://ccrma.stanford.edu/~jos/pasp/Extended_Karplus_Strong_Algorithm.html

Sample-based Synthesis

- The majority of digital sound and music synthesis today is accomplished via the playback of stored waveforms
  - Media production: sound effects, narration, prompts
  - Digital devices: ringtone, sound alert
  - Musical Instruments
    - Native Instrument Kontakt5: 43+ GB (1000+ instruments)
    - Synthogy Ivory II Piano: 77GB+ (Steinway D Grand, ...)

Foley (filmmaking)

Ringtones

Synthogy Ivory II Piano

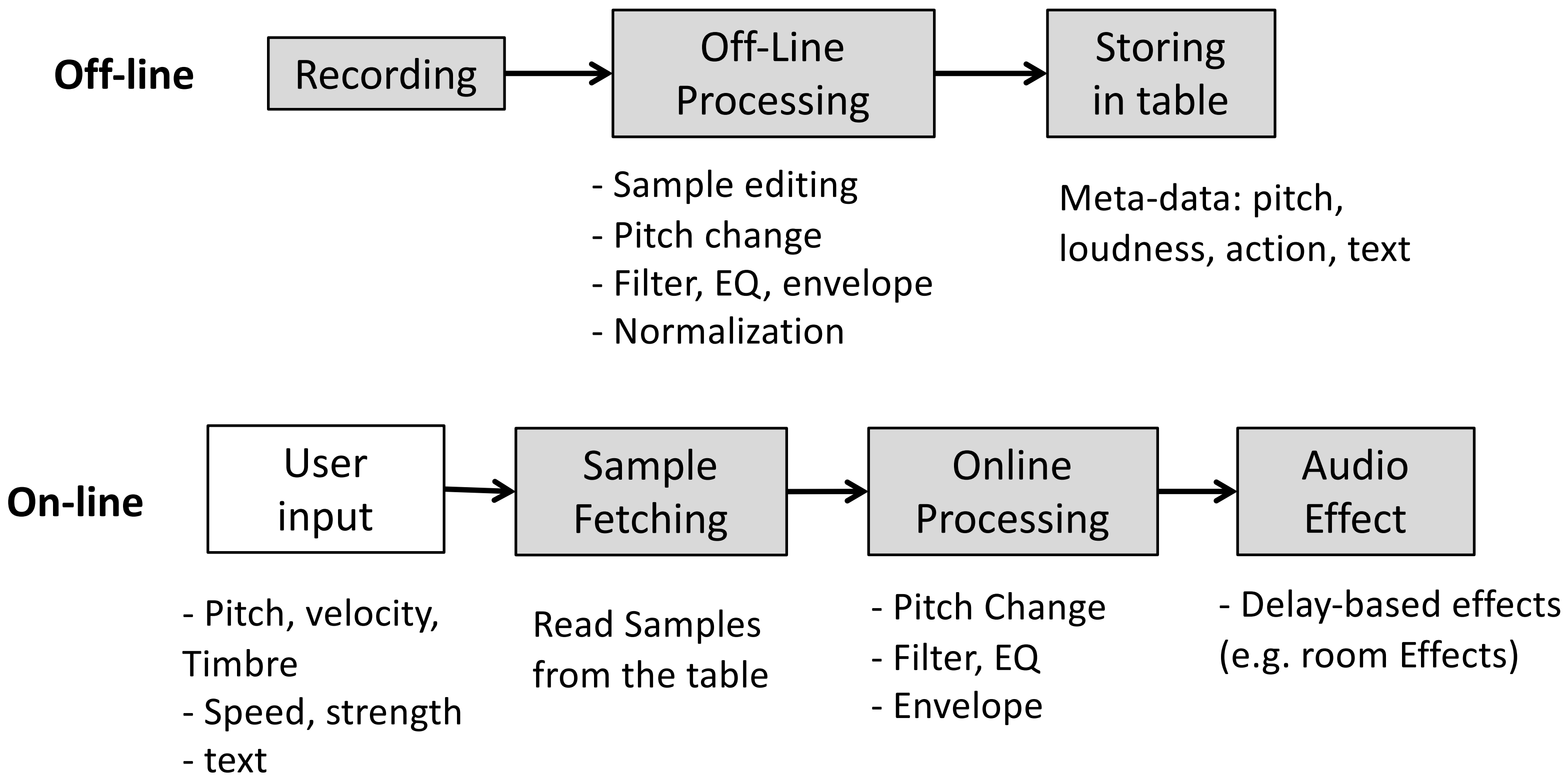
Why Don't We Just Use Samples?

- Advantages
  - Reproduce realistic sounds (needless to say)
  - Less use of CPU

- Limitations
  - Not flexible: repeat the same sound again, not expressive
  - Can require a great deal of storage
  - Need high-quality recording
  - Limited to real-world sounds

- Better ways
  - Modify samples based on existing sound processing techniques
    - Much richer spectrum of sounds
  - Trade-off: CPU, memory and programmability

Sample-based Synthesis



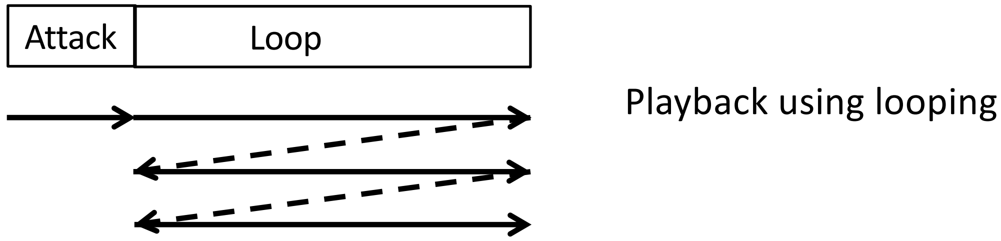
Wavetable Synthesis

- Playback samples stored in tables
  - Multi-sampling: choose different sample tables depending on input conditions such pitch and loudness
    - Velocity switching

- Reducing sample tables in musical synthesizers
  - Sample looping: reduce the size of tables
  - Pitch shifting by re-sampling: avoid sampling every single pitch
  - Filtering: avoid sampling every single loudness
    - e.g. low-pass filtering for soft input

Sample Looping

- Find a periodic segment and repeat it seamlessly during playback
  - Particularly for instruments with forced oscillation (e.g. woodwind)
  - Usually taken from the sustained part of a pitched musical note



- It is not easy to find an exactly clean loop
  - The amplitude envelopes often decays or modulated:
    - e.g. piano, guitar, violin
  - Period in sample is not integer → non-integer-size sample table?

Sample Looping

- Solutions
  - Decaying amplitude: normalize the amplitude
    - Compute the envelope and multiply it inverse
    - Then, multiply the envelope back later
  - Non-integer period in sample
    - Use multiple periods for the loop such that the total period is close to integers
      - * e.g. Period = 100.2 samples → 5*Period = 501 samples
  - Amplitude modulation
    - Crossfade between the end of loop and the beginning of loop meet

- Automatic loop search
  - Pitch detection and zero-crossing detection: c.f. samplers

Concatenative Synthesis

- Splicing sample segments based on input information
  - Typically done in speech synthesis: unit selection

- Sample size depends on applications
  - ARS: limited expression and context-dependent
    - word or phrase level
  - TTS: unlimited expression and context-independent
    - phone or di-phone (phone-to-phone transition) level

Summary

	Memory (Storage)	Programmability (by # of parameters)	Reproducibility of natural sounds	Interpretability of parameters	Computation power
Additive	**	*****	****	****	****
Subtractive	*	***	**	***	**
Non-linear	*	***	**	**	**
Physical model	***	**	****	*****	*** ~ *****
Sample-based	*****	*	*****	N/A	* ~ ***