Sound Synthesis Methods

Matti Vihola, mvihola@cs.tut.fi

23rd August 2001

1 Objectives

The objective of sound synthesis is to create sounds that are

- Musically interesting
- Preferably realistic (sounds like some instrument)
- Produced in real time

A taxonomy of digital sound synthesis

1. Abstract algorithms
2. Processed recordings (sampling)
3. Spectral models
4. Physical models

2 Abstract algorithms

Typical features of abstract algorithms are the simplicity and ease of implementation. The sound is more or less artificial compared to more sophisticated methods[1].

2.1 Frequency Modulation (FM) (‘70s)

Early digital synthesizers and sound card synthesizer chips are based on FM.[2]

- Very simple, easy to implement, only couple of voltage controlled oscillators (VCOs)
- Time-variant structure like in natural sounds, fast vibrato
- Bell-like and metallic timbres
- Harmonic sound when integer ratio between carrier and modulator
- Feedback systems add stability to frequency behaviour

[1] Which may not be always a drawback…
[2] Take a look at `http://www.obsolete.com/120_years/`

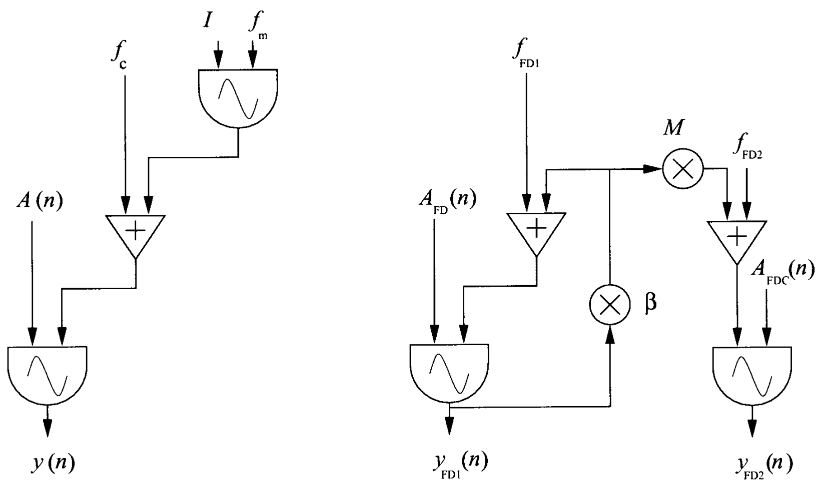

Figure 1: Left: simple FM synthesis, right: one-oscillator feedback system (y_{FD1}) and two-oscillator feedback system (y_{FD1})

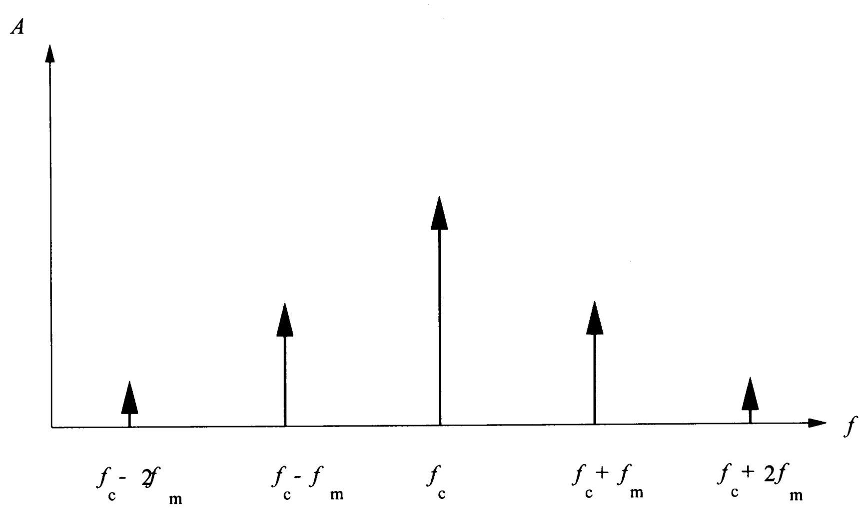

Figure 2: Frequency-domain presentation of FM synthesis.

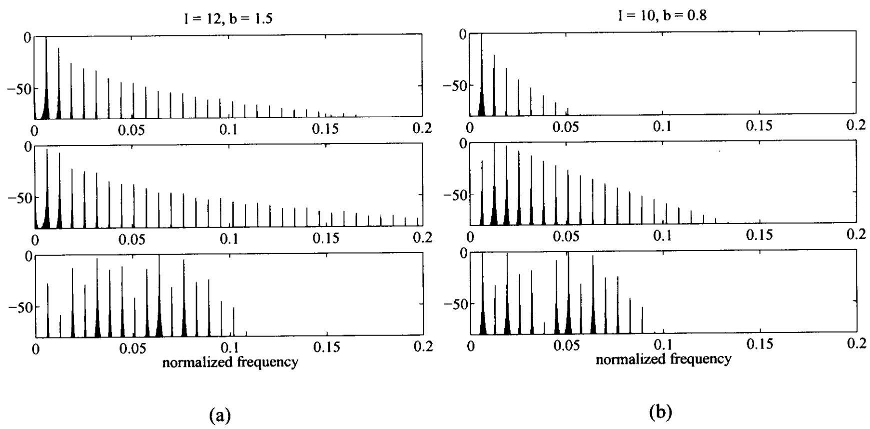

Figure 3: Spectra of (top to down) one-oscillator feedback, two-oscillator feedback and simple FM with different modulation indices and feedback gains.

2.2 Waveshaping Synthesis (Nonlinear Distortion) (late ‘60s)

- Nonlinear shaping function applied to the input (excitation) signal
- Most fundamental case: excitation signal sinusoidal
- By using a linear combination of Chebyshev polynomials the ratios of harmonics can be controlled $\Longrightarrow$ can be made to correspond harmonic structure of a real instrument
- Post-processing can be applied, e.g. amplitude modulation (AM)
- Exitation signal can be something else than sinusoid

2.3 Karplus-Strong algorithm (‘83)

- Very simple and computationally effective algorithm
- A short sound buffer, initialized with a random data
- Looping of the buffer, which is filtered with a simple low-pass filter[3] after every read
- Plucked string tones or percussion tones
- Can be implemented using only shift and add operations

3 Processed recordings (sampling)

Manipulation of recorded sounds dates back to 1920s. The memory requirements has been a problem with digital sampling synthesis. The idea is in recording relatively short samples of sounds, which are then played back.

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

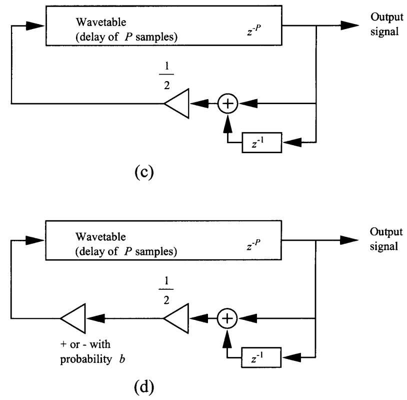

Figure 4: Karplus-Strong algorithm. (c) Simple filtering operation, results in a plucked string tone. (d) Flip sign of every filtered bit with certain probability, produces sound resembling percussion tone.

3.1 Digital Wavetable Synthesis

Looping Typical parts of instrument sound: attack, steady state, release

- Waveform of steady-state sound of most of the instruments approx. periodical $\Longrightarrow$ short sample of the steady state looped
- Loop points determined: length of the looped sample should correspond to fundamental frequency

Pitch shifting every 3th or 4th semitone stored, pitch shifting applied to nearest to obtain the rest

Data reduction Lossless or lossy compression. Alteration of sampling frequency, quantization step or more sophisticated method.

3.2 Multiple Wavetable Synthesis

More than one wavetable, or sample played at once.

Wavetable Cross-Fading Several sections of tone stored, samples are cross-faded from one to another multiplying with amplitude envelope.

Wavetable Stacking Several (arbitrary) sound signals multiplied with amplitude envelope and summed together $\Longrightarrow$ synthetic sound signal. Problem: Find single wavetables (spectra) and amplitude envelopes to produce natural sound.

3.3 Granular Synthesis (‘40s)

Idea is representing sound signals with ‘sound atoms’ or ‘grains’. Waveform shape of grains and their temporal distribution determine the type of sound.

3.3.1 Asynchronous (AGS) (‘90s)

- Scatter sound grains in a statistical manner over a region in time-frequency plane
- Region called a ‘sound cloud’
- Several parameters: Start time, duration, bandwidth of clouds; duration, density, amplitude envelope, waveforms and spatial distribution of grains
- Effective in generating new sound events, simulations of existing instruments hard

3.3.2 Pitch Synchronous (PSGS) (‘90s)

- Better performance in simulation of realistic sounds
- Periodicity corresponding to fundamental frequency
- Analysis grains using STFT or LPC analysis $\Longrightarrow$ impulse response estimation
- Resynthesis using parallel FIRs driven with train of impulses
- Transformations that add variation to the produced signal

All methods that use overlap-add technique can be viewed as granular synthesis.

4 Spectral Models

The idea is that the properties of sound that are perceived are stored (time-varying spectra).

4.1 Additive Synthesis

- Summing sinusoidal components of different (phase,) amplitude and frequency

$$y(n) = \sum_k A_k(n) \sin\left[\frac{2\pi n}{Fs} F_k(n)\right]$$

- The control functions of single sinusoids ($A_k(n)$, $F_k(n)$) are slowly-varying
- Drawback: Large amount of data (control function parameters) and large number of oscillators
- Data reduction that preserves intuitively appealing data and original sound (e.g. line segment approximation of control parameters)

4.2 Phase Vocoder

- Can be viewed as bank of filters or STFT analyzer
- Resynthesis: FT^{-1} and overlap-add
- Time-scaling and pitch transposition easily:
  - Time-scaling accomplished by modifying the hop size (the time difference of consecutive synthesized frames)
  - Pitch modification by modifying time scale, and then changing the sampling rate of the signal
- Works well for harmonic, slowly varying tones, blurs transient type of sounds (the time resolution of the STFT-analysis)

4.3 Source-Filter Synthesis

- Exitation signal filtered with time-varying filter
- Titled also as subtractive synthesis: input signal with harmonic rich spectrum *filtered*
- Not very robust representation of realistic sounds
- Used in analog synthesizers: some exitation waveform generators and filters

4.4 McAulay-Quatieri (MQ) (‘86)

- Original signal decomposed to a set of sinusoids $\{A_k^l, \omega_k^l, \psi_k^l\}$
- Trajectories for all components
- Analysis: STFT, peak detection, sinusoidal trajectory detection and noise thresholding
- Synthesis: Trajectory interpolation, additive synthesis

4.5 Spectral Modeling Synthesis (SMS) (late ‘80s)

- Sinusoidal analysis with MQ $\Longrightarrow$ deterministic part of the signal
- Residual $x_{res}(n) = x(n) - x_{sin}(n)$ modeled as noise (stochastic component)
- Duration (tempo) and frequency (key) modification easily
- Transient sounds a problem

4.6 Transient Modeling Synthesis (TMS) ('97)

- Extension of SMS
- The residual part of SMS presented as transients and noise, $x_{res}(n) = x_{tra}(n) + x_{noi}(n)$
- The 'noise' part is then only the steady noisy components
- Time $\leftrightarrow$ frequency domain duality: impulsive signals in time domain $\leftrightarrow$ sinusoidal in frequency domain
- DCT produces real-valued sinusoid when the time domain signal is an impulse
- Transients detected and parametrized, resynthesized $\Longrightarrow$ steady noisy components left

4.7 FFT^{-1} Synthesis ('92)

- Can be viewed as additive synthesis in frequency domain
- All the signal components added together as spectral envelopes
- Synthesized using overlap-add of consecutive frames
- Complex sounds possible

4.8 Formant Synthesis

- Formant is a concentration of energy in energy/power spectrum envelope
- Originally in speech processing
- Used in speech synthesis (and singing)

5 Exercise Work

1. Implement the simple FM synthesis. Search suitable values for f_c and f_m in order to make the signal sound funny. Test with different values of modulation index I.
2. Implement the Karplus-Strong algorithm. Find suitable buffer size, and try to create both plucked string and percussion type of sounds

You can find the more detailed description of the methods e.g. in pages 4–10 of [1].

References

[1] Tolonen T., Välimäki V. and Karjalainen M., "*Evaluation of Modern Sound Synthesis Methods*", Report 48, HUT, Department of Electrical and Communications Engineering, Laboratory of Acoustics and Audio Signal Processing, Espoo, Mar 1998. Available at `http://www.acoustics.hut.fi:7077/~ttolonen/sound_synth_report.html`

[2] Välimäki V. *“Lecture notes on Digital Sound Synthesis”*, HUT, Laboratory of Acoustics and Audio Signal Processing. Available at `/share/argh/klap/HUT_ASP/asp-8-synthesis.pdf`