Electronic musical instruments

COMPUTER MUSIC TOOLS

Computer music tools

- Software for creation of computer music (both single sounds and whole musical pieces).
- We have already learned about:
  - samplers – musical instruments and software for creating instrument banks,
  - MIDI sequencers – software for recording, editing and playback of MIDI codes, controlling other musical instruments (hardware and software).

VST

Virtual Studio Technology (VST) – a standard from Steinberg.

- VST plugins:
  - VST effects: sound effects; they receive digital sounds and they output processed sounds,
  - VST instruments (VSTi): they receive MIDI codes and generate digital sounds (by synthesis, sampling, etc.) and they send them to the output,
  - VST MIDI effects: they process MIDI codes (rarely used).
- VST host: a software that sends data to plugins and collects the results.

VST instruments

The task of a VSTi programmer is to:

- write an algorithm that generates digital sounds (using any method) according to the received parameters,
- create a user interface,
- define MIDI Control Change parameters and their interpretation (how they affect the generated sound).

A programmer does not need to worry about input and output functions – this is the task of the host program.

VST host - DAW

Modern VST hosts are digital audio workstations (DAW)

- audio tracks:
  - recorded digital sound (instruments, vocals),
  - use VST effects;
- MIDI tracks:
  - only recorded MIDI codes,
  - control VSTi or hardware instruments,
  - allow for sequencer functions (MIDI code editing),
  - a digital sound is created only during the mastering.

Advantages of VSTi

Why should we use VSTi and MIDI tracks instead of simply recording sounds on audio tracks:

- we can easily edit MIDI codes (individual notes),
- we can modify the VSTi sound by changing its parameters,
- we can change the VSTi leaving the same MIDI sequences,
- we can use many instruments at the time, the processing power of a PC is the limit.

Running VSTi

How to run a single VSTi without installing a complex DAW?

- Download free *SAVIHost*

  (http://www.hermannseib.com/english/savihost.htm)
- Put savihost.exe in the same location as the DLL file of the VSTi.
- Rename the file so that EXE and DLL names match, e.g.:
  - the plugin file is *Synth1 VST.dll*
  - rename *SAVIHost.exe* to *Synth1 VST.exe*
- Run the executable file. The plugin is loaded, and we can use it as an instrument.

Computer music programming - CSound

- General purpose programming languages (C++, Java, Python, etc.) do not have easy procedures for computer music creation.
- There are special programming languages developed for creating digital sounds and music.
- CSound – one of the most popular music programming languages. A programmer writes procedures for:
  - *orchestra* – code that creates sounds (instruments),
  - *score* – code that creates music from these sounds.
- Many functions, but relatively hard to learn.
- Documentation and many examples are available.

Csound - example

Csound script
- simple FM synthesis

http://booki.flossmanuals.net/csound/d-frequency-modulation/

```
<CsoundSynthesizer>
<CsInstruments>
sr = 48000
ksmps = 32
nchnls = 2
0dbfs = 1
instr 1

kCarFreq = 440     ; carrier frequency
kModFreq = 440     ; modulation frequency
kIndex = 10        ; modulation index

kIndexM = 0
kMaxDev = kIndex*kModFreq
kMinDev = kIndexM*kModFreq
kVarDev = kMaxDev-kMinDev
kModAmp = kMinDev+kVarDev

; oscillators
aModulator poscil kModAmp, kModFreq, 1
aCarrier poscil 0.3, kCarFreq+aModulator, 1

outs aCarrier, aCarrier
endin
</CsInstruments>

<CsScore>
f 1 0 1024 10 1     ; Sine wave for table 1
i 1 0 15
</CsScore>

</CsoundSynthesizer>
; written by Alex Hofmann (Mar. 2011)
```

SuperCollider

- Modern computer music programming language.
- Code written as text (scripts).
- Used for:
  - real time sound synthesis,
  - algorithmic composition.
- Uses a client-server architecture.
- Less examples than Csound.
- Available in many operating systems.
- Ability to create custom instruments.

SuperCollider - example

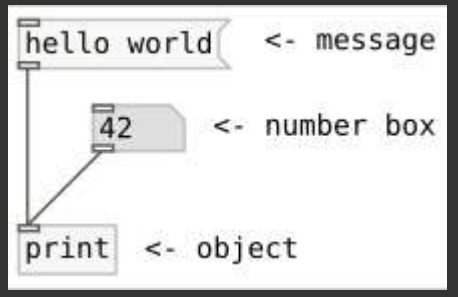
SuperCollider script to create a simple FM synthesis

http://danielnouri.org/docs/SuperColliderHelp/Tutorials/Mark_Polishook_tutorial/Synthesis/14_Frequency_modulation.html

```
(
SynthDef("fm1", { arg bus = 0, freq = 440, carPartial = 1, modPartial = 1, index = 3, mul = 0.05;
            // carPartial :: modPartial => car/mod ratio

            var mod;
            var car;

            mod = SinOsc.ar(freq * modPartial, 0,
                        freq * index * LFNoise1.kr(5.reciprocal).abs);

            car = SinOsc.ar( (freq * carPartial) + mod, 0, mul);

            Out.ar(bus, car)
}).load(s);
)

(
Synth("fm1", [\bus, 0, \freq, 440, \carPartial, 1, \modPartial, 1, \index, 10]);
)

(
s.queryAllNodes;
)
```

Pure Data (pd) / Max/MSP

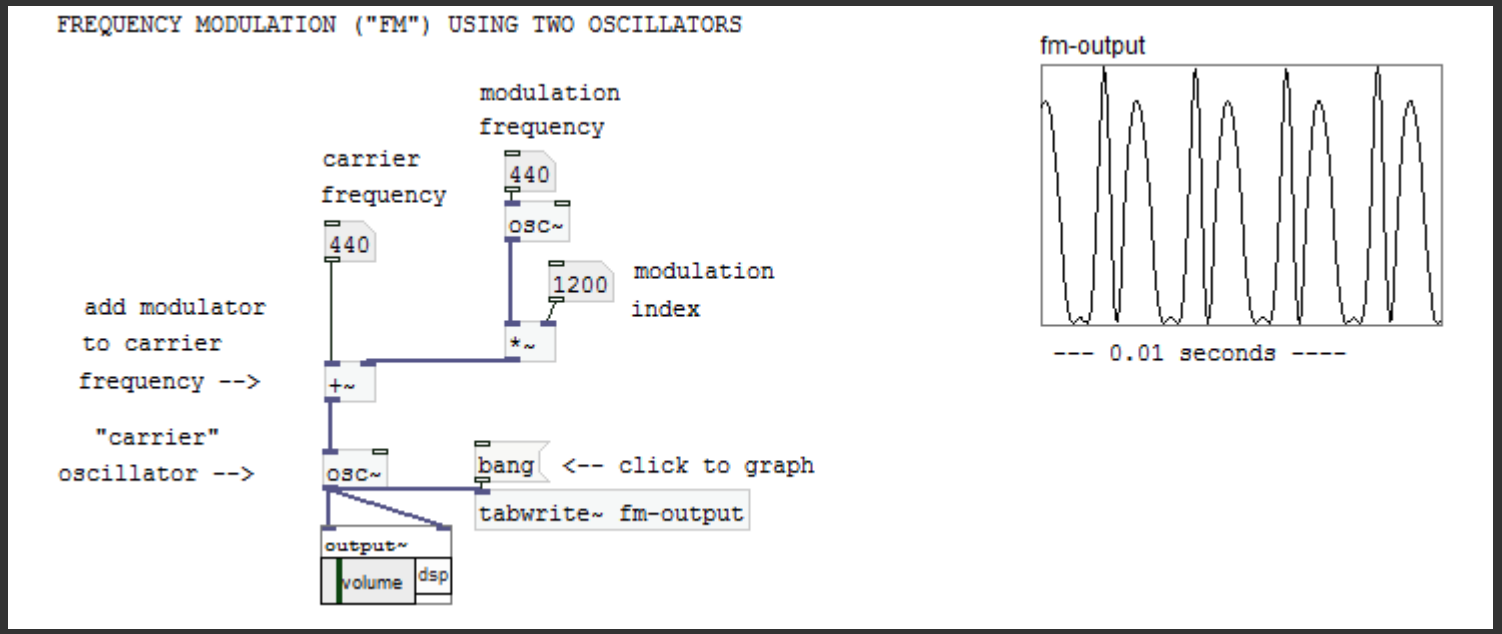
- An alternative approach: visual creation of algorithms in GUI instead of writing a code.
- Implementations:
  - Max/MSP – commercial, MacOS,
  - *Pure Data* (pd) – open source (Linux, Windows, Mac).
- Sound processing, synthesis, sampling, MIDI.
- The user interface is rather ascetic.
- More appealing visually, but complex algorithms may look “tangled”.
- Ability to create custom blocks.
- Audio and control signals.



Pure Data - example

A simple FM synthesis in pd

an example included in the installed program (*A09.frequency.mod.pd*)

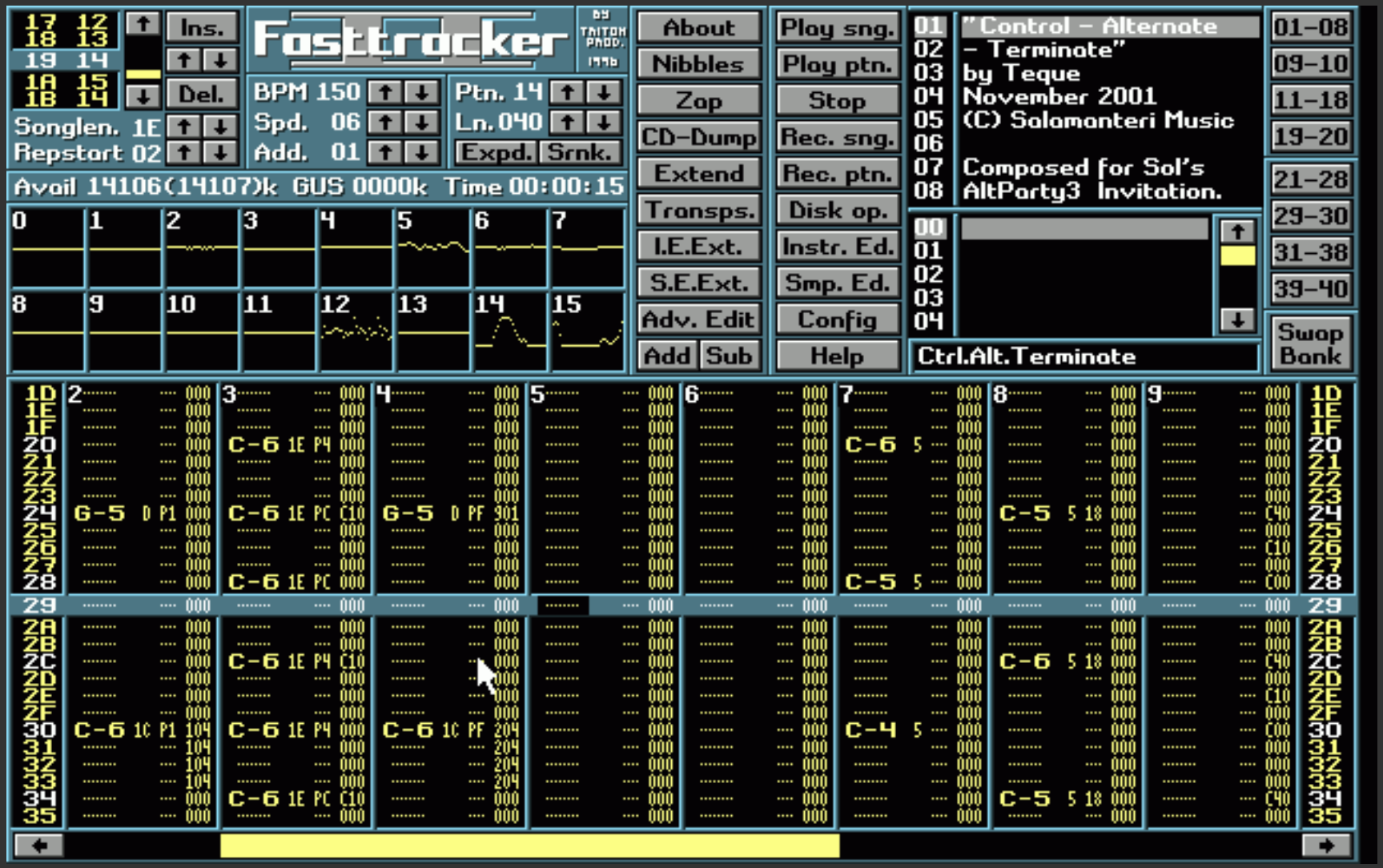


Trackers

Trackers were popular in 1980s on 16-bit computers (Amiga).

- Combine functions of a sampler and a sequencer.
- Samples are very short, they are usually looped.
- Instruments – samples with envelope and effects.
- Patterns – control sequences that generate music.
- Order – defines how the patterns are played.
- Module (MOD) – a single file that contains all the data, very small (typically, below 100 KB).

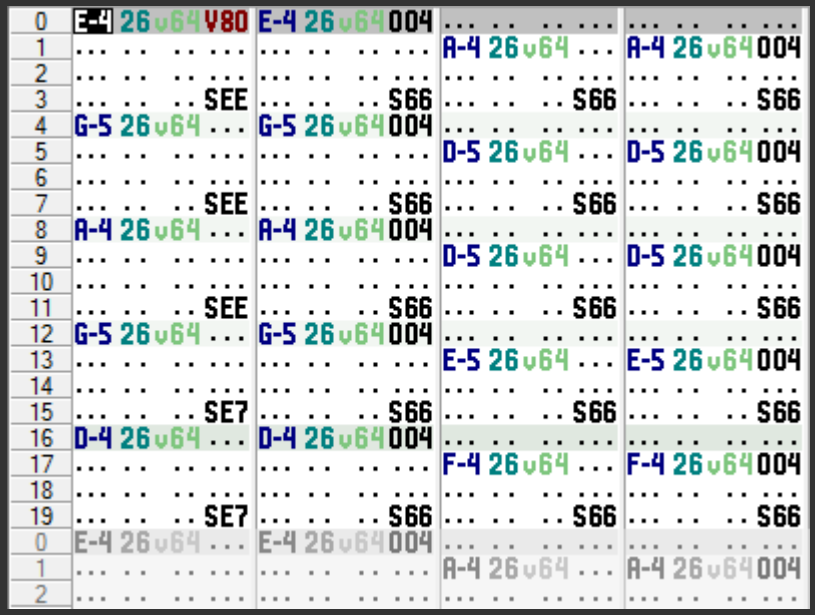
Fasttracker (Amiga)



Patterns

Pattern describes how the sounds are generated.
A pattern has a form of a table.

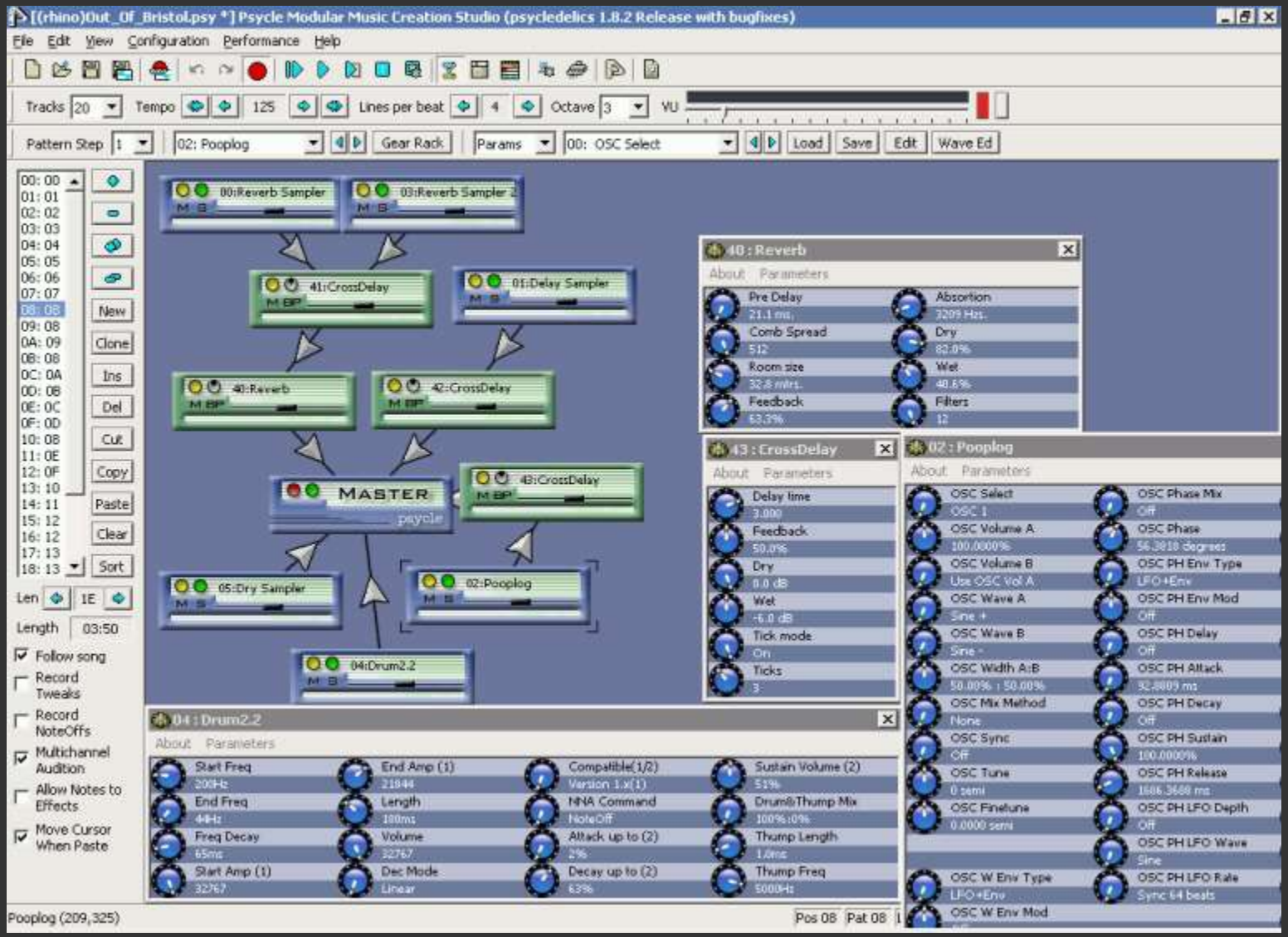
- Columns – tracks, voices.
- Rows – lines, notes on a timeline.
- Cells – notes:
  - pitch,
  - instrument number,
  - loudness,
  - sound effect.



Modular trackers

- Modern trackers use modules (machines) instead of recorded samples:
  - generators – synthesizers and samplers,
  - effects – modify the sound,
  - amplifiers, mixers, etc.
- The machines are controlled with patterns, just like in the original trackers.
- Much more capabilities of music creation.
- Programmers can create custom machines.

Psycle - example of a modular tracker



Examples of trackers

Modern trackers:

- *OpenMPT* – a free “traditional” (MOD) tracker, Windows
- *Psycle*, *Buzz* – free modular trackers,
- *Renoise* – a commercial DAW based on the tracker concept.

Some software allow for creation of modular systems that are controlled with MIDI instead of patterns. An example: *NI Reaktor* (commercial).

Algorithmic composition

Algorithmic composition uses software to create musical pieces (“songs”). A computer becomes a composer.

Methods of algorithmic composition (can be used simultaneously):

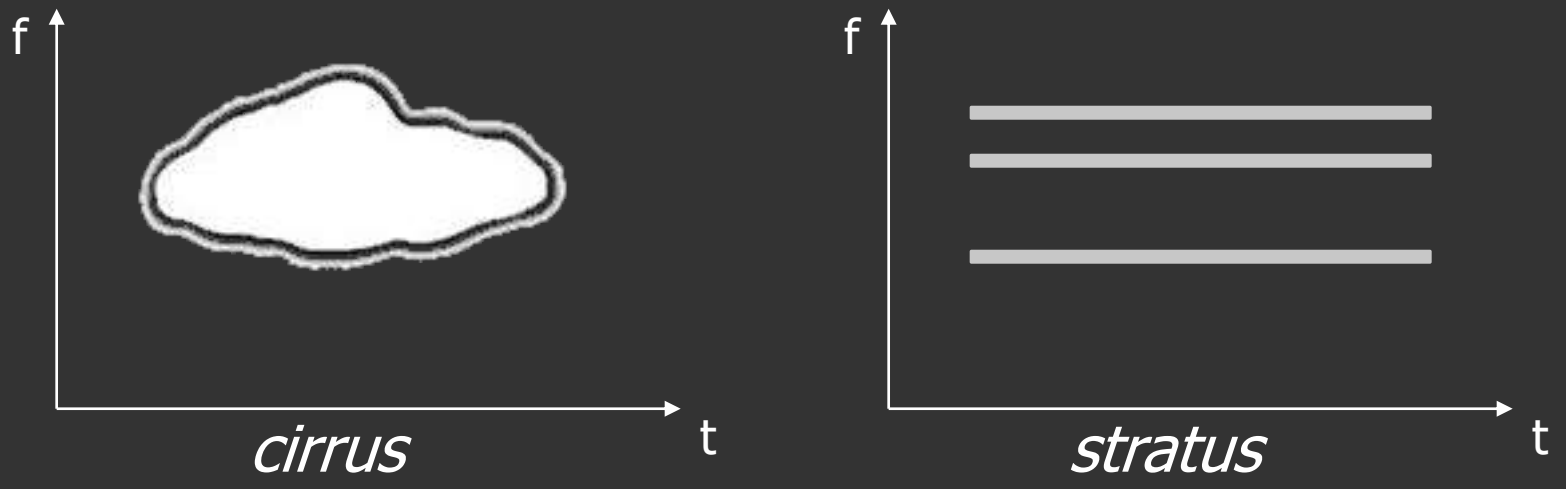
- grammatic algorithms based on music theory,
- algorithms based on statistic models,
- randomized algorithms,
- fractal algorithms,
- artificial intelligence methods, deep learning.

Example software (try it!): *cgMusic*
(http://codeminion.com/blogs/maciek/2008/05/cgmusic-computers-create-music/)

Granular synthesis

The granular synthesis method combines sound synthesis with algorithmic composition.

- Granules are very short sound fragments (below 50 ms).
- Each granule has pitch and loudness.
- A large set of granules is organized in clouds (soundscapes) on a time-pitch plot.



Granular synthesis

- We can control pitch, loudness and tempo of the clouds.
- Organized clouds create sounds.
- Clouds can be created by a human or by an algorithm.
- The created sounds are “psychedelic” in nature, not like real sounds. It can be used e.g. to create an original soundtrack.
- Wavelet synthesis is similar to the granular synthesis. It uses wavelet decomposition to create granules with desired parameters.

Bibliography

- VST SDK (Steinberg), for VST programmers: http://ygrabit.steinberg.de/~ygrabit/public_html/
- SAVIHost (free VST host): http://www.hermannseib.com/english/savihost.htm
- CSound: http://www.csounds.com/
- SuperCollider: http://supercollider.github.io/
- Pure Data (pd): https://puredata.info/
- Tracker *OpenMPT*: https://openmpt.org/
- *Psycle*: psycle.pastnotecut.org
- Wikipedia – Algorithmic composition: https://en.wikipedia.org/wiki/Algorithmic_composition