11 September 2020

A little back ground...

- A modulular synth simulation is one of the earliest ideas in computer music.
- Bell Labs 1950's (including Max Mathews)
recognised that analog electronic circuits could be understood
performing mathematical operations that can be modelled digitally.

- Previous programs BLODI, Music I-V....acoustic compilers.

Basic design

Concept:

- A basic method of analog synthesis is to patch together self contained modules that generate or modify electric signals.
- Csound uses this paradigm, instead of modules csound has unit generators and patch cords are signal variables.

So, there are are two parts to this investigation:

1. Write a "patch specification language"(PSL). A simple "language"(program) to transtate patch specifications to csound orchestra.

2. Develop it into a programme will algorithmically generate patch specifications, automating the sound design process.

Why?...

...PSL instead of writing Csound code? It should encourage creation of more complex patches, as PSL is more convenient.

Which analogue modules shall we model?

- Oscillator
- Filter
- Sample and hold unit
- Noise

Unit Generators (Modules) / interface design

Oscillator.

Patch Specification:
- Adjust Waveform.
- Adjust Frequency.
- Frequency Modulate oscillator.
- Amplitude Modulate oscillator.
- Specifiy oscillator range.
- Output name (patch cord).

Intereface:
- output variable.
- frequency.
- waveform.
- AM signal.
- FM signal.

Csound Code:

OSC a1 400 SINE NONE NONE -1 1

Format Alternative

- to patch by hand instead of automation you could use:

OSC OUT=a1 FREQ=440

- filter
- sample hold

Building the Code

C needs to.
-> Read in patch specifications.
-> Turn read patch specifications into Csound code.
-> Embed the Csound code patch specificaions into a working instrument.
-> Write Csound orchestra that can synthesize the patch.

Code can be found as:
- synthmodule.h
- synthmodule.c

The Command-Line Interface

Use the synthmodule (executable file) in the shell. A data file is the module patch to configure our synth. This can be done on the command line using:

λsynthmodule < modulepatch1

Tightening the Structure and Making It Safer (note: xtremeprgramming.com)

At the moment there are a maxium of 256 OSCS.

print*When*we

Next I think is the algorithm for the Frequency and ... If the minimum frequency is !=-1.0 and if maximum frequency is !=1.0

Rescale the frequency with maxium frequency minus the minimumfrequency then divid by 2.0.

We then ask for the output signal to be: the minmum frequency plus (rescaled frequency * output signal + rescaled signal)