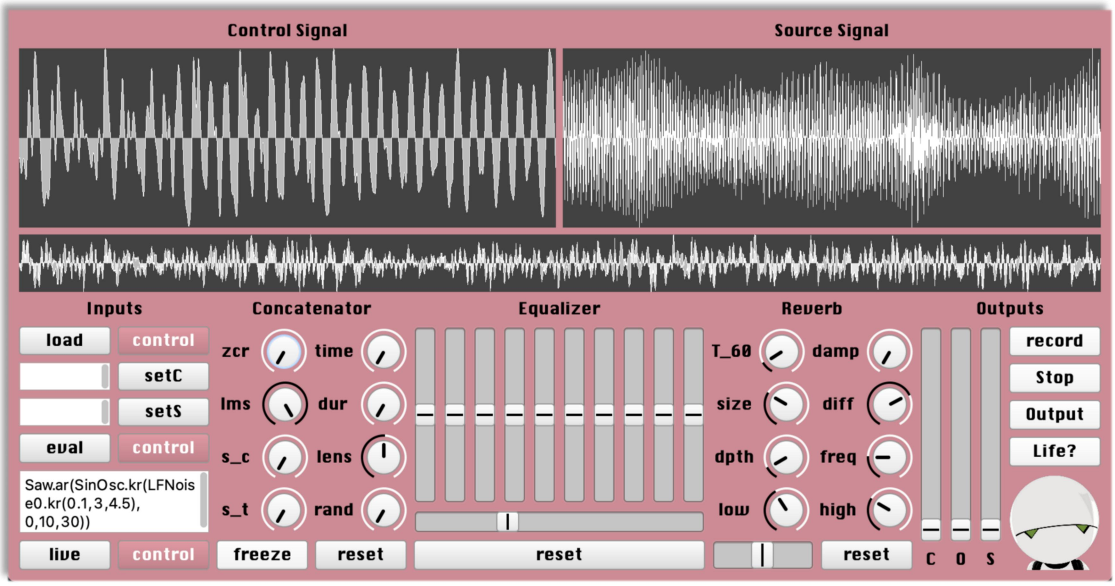
User Manual for
Concatenative Sound Synthesizer
Version 0.1



zeyuyang42

15-01-2022

Q&A

What’s this?

It is a synthesizer by its name, obviously. If you don’t know what is a synthesizer just go and see the Wikipedia. Not like well-known synthesizers from Moog or YAMAHA which could be regarded as standard music instruments nowadays, It is a Concatenative Sound Synthesizer (CSS) that is much more like a “noise” generator for the difficulty of getting melodies or chords from it. But you could synthesis some crazy sounds by it. And for sure it is impossible to find all those unbelievable sounds by any other kinds of synthesizers. The algorithm is simple and determined but the outputs are complex and uncontrollable. There are no predictions for the outputs caused by different Input signals and parameter setting. As a chaotic system it could bring much surprise.

What could I use it for?

For its properties as mentioned above, CSS is good at making glitch or ambient music. It is an amazing tool for the pad sounds or adding some “error-like” vibe.

What should I do if I want to use this synthesizer?

Just download it and use it to write your music pieces or to achieve some random crazier ideas. It’s fully open source:). Of course I will be grateful if you could mention me or cite the link of my Github.

1. Environment

This software is based on Supercollider and can only run in the sclang develop environment. It could have a VST or standalone version in the future if I want to do more practice.

1.1 Setup

1) Download Supercollider

Go to the offical website of Supercollider and download the current version for your system. This project was created under SuperCollider 3.12.1 for Mac and not tested in Linux or Windows environments.

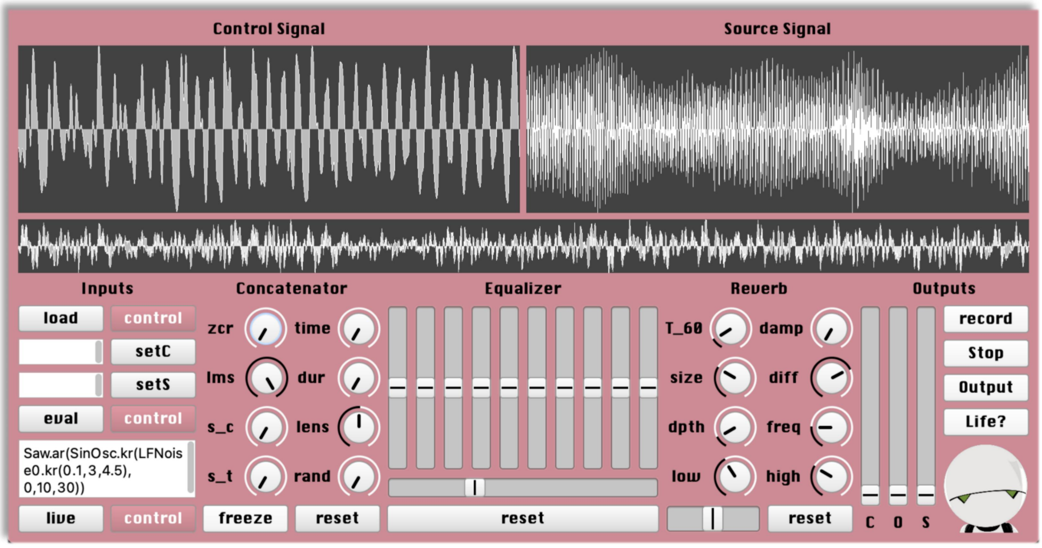
2) Preparation

Run[1] the first three lines to boot the Supercollider engine and set basic parameters

```
s.boot;
s.options.sampleRate = 44100;  // in case of sample rate mismatchi, mostly caused by Zoom's default 48000Hz setting
s.options.memSize_(65536 * 4); // according to the Note in JPverb
```

3) Run the synthesizer

Run the main codes and you can see the GUI and hear the sound now. If something goes wrong, for example the audio scopes are horizontal lines, just close the old one and try it again. :<



[1] In case of someone don't know how to run codes.

Put the cursor to the line you want to run and click CMD+Enter for Mac or Ctrl+Enter for Windows. To run the main codes just put cursor to an arbitrary line inside the brackets

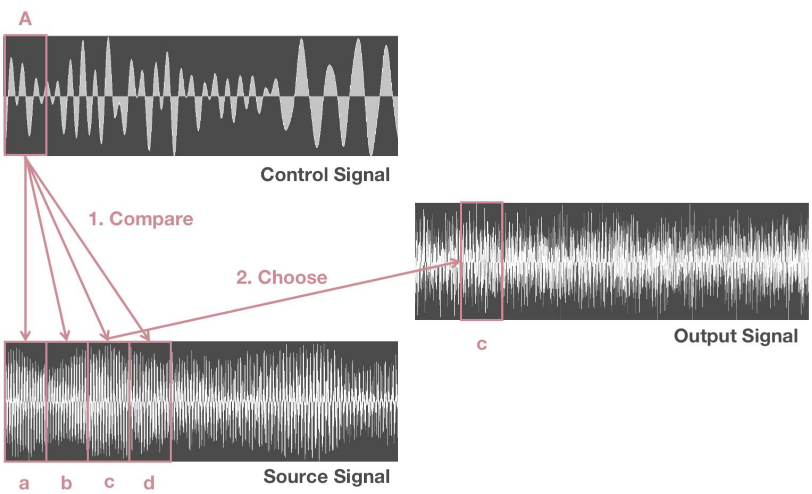
2. Concatenative Synthesis Algorithm

"Implementation of concatenative sound synthesis based on matching four features from a control input to the database generated from a source input. The control determines which frames of the source get played back, and you can change the weighting of features to refine your matching criteria (ie, make rms the most heavily weighted to have amplitude as the most important criteria). You can also modulate the match length, being the amount of source played back for a match, the feature weightings, and freeze the database collection to work with a collected sample from the source."

Hereby is a citation of comments from the core extension used in this synthesizer. Metaphorically speaking it is musical mosaicing or "musaicing". The basic idea behind is simply replace each small clips of audio signal with the signal from others according to some predefined criteria. Under the same Idea comes different implementation strategies. The store and replace behavior could be real-time or static. The criteria could be acoustic descriptors or unexplainable embedding. Check this Neural Granular Sound Synthesis from IRCAM.

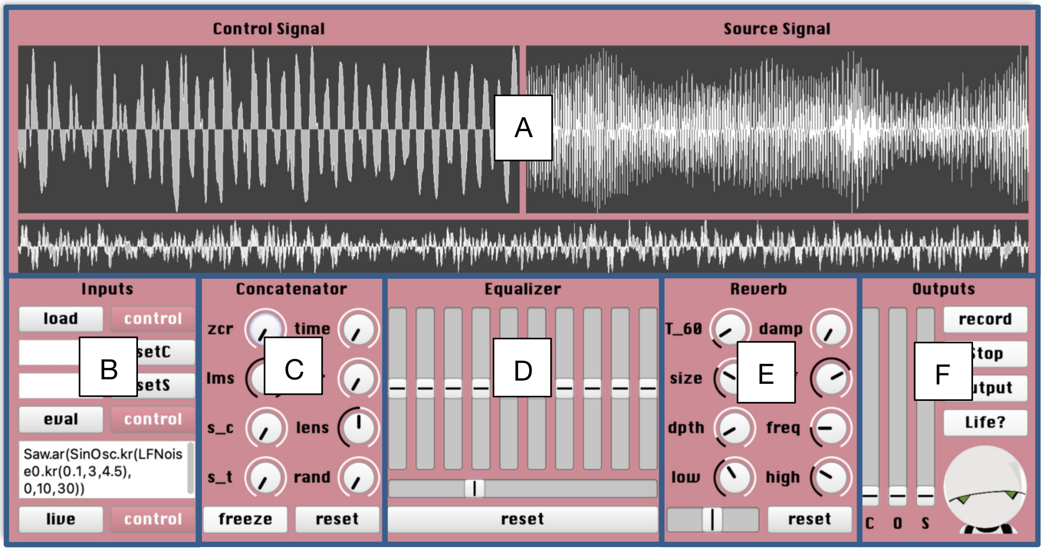
2.1 Oversimplified demonstration

There are two kinds of signals, namely control signal and source signal. Both Signals are sliced into small pieces and marked by Alphabets. What we want to do now is to find a small clip in source signal (a to d) which is the most similar one to A in control signal. Then choose it as part of the output signal. Keep this process for the following clip in control signal until the entire output signal is generated.
For the real-time rendering version used in CSS, the only different is that not the whole Signal but only signal with a certain length inside a window will be processed. The control and source signal inside the window keep changing so that makes the output unpredictable.



3. GUI

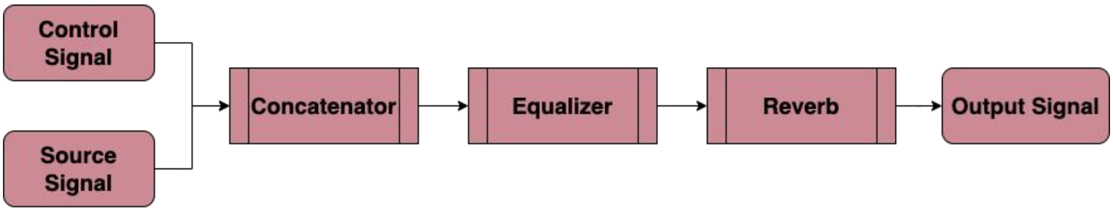
The graphic user interface(GUI) of CSS can be separated into 6 components.
A: Oscilloscopes where waveforms of the control signal (top left), the source signal (top right) and the output signal (below) are plotted.
B: Input control panel where the control and source signals are loaded or generated and manipulated.
C: Concatenator where two input signals are processed by algorithm in chapter 2 and output signal is generated in real-time.
D: 10-band Equalizer with adjustable Q factor.
E: High quality reverb with inherent chorus effect and dry/wet control.
F: Output control panel with utilities (also the Marvin is watching you~).



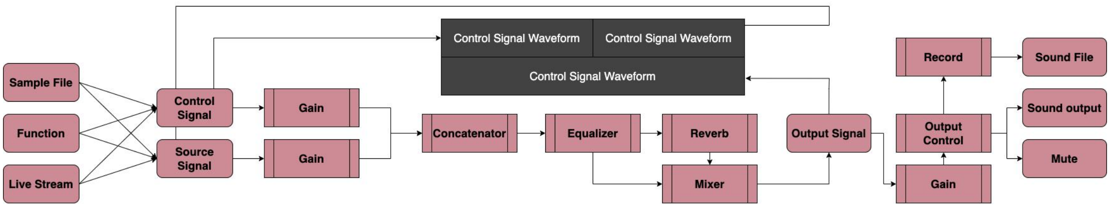
The meaning of every buttons and sliders and the connection between each panels will be explained in following chapters.

4. Signal Chain

A brief figure that shows the overall signal chain of CSS. The control and source signals are sent to the concatenator and processed by the Algorithm in Chapter 2 to get the raw concatenated signal. The raw signal is reshaped by Equalizer. Then adjustable amount of reverb effect is added to it and the final output signal is generated. All the signals are monophonic.

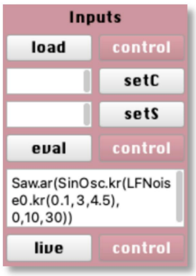


Combined with all utilities here presents the signal chain of CSS system with whole details, which will be introduced in following chapters.



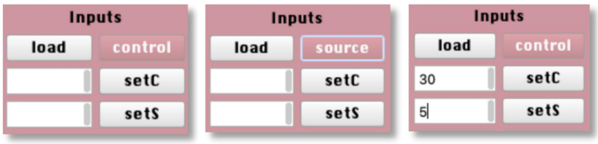
5. Input Control

The original control and source signals are loaded recorded or manipulated by the input control panel. There are three types of signal that can be used as control or source.



5.1 Sample File:

The control signal and source signal can be loaded from sample. As showed in the default setting that signals are loaded from audios in .wav format into buffer and keep looping. The control\source switch button is used to indicated which buffer the audio is loaded to. The replay time (how many seconds needed to play through the entire sample) for both signals can be assigned by typing number into test-viewer and click setC or setS



5.2 Supercollider Function:

Signals can also be generated by supercollider's sound engine for whom has the experience for sclang. Synth function in the test-viewer will be interpreted and run on the sound engine after choosing the signal type and hitting the eval button.

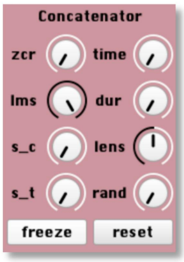
5.3 Live Audio stream:

This system can also read audio stream from your computer's microphone or soundcard. Use live button to switch to audio stream mode for control or source signal.

6. Concatenator

Concatenator panel provides parameters and functions for the controlling of concatenative algorithm. The left four knots controls the four acoustic descriptors implemented in this system as match-criteria. The right four knots set the parameters of windowing and matching strategies. Freeze can stop collecting novel source input, keep store fixed. So the output sound will become stable and keep looping. Reset is used to put every parameters back to initial value.

Please note that it is the ratio between four descriptors not the absolute value that dominates the importance of criteria in matching. Exp. All 1 for zcr, lms, s_c, s_t perform exactly same as all 0.

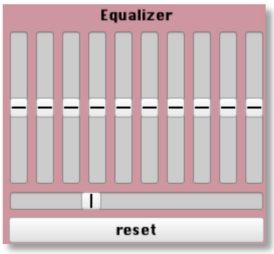


Name	Range	Description
zcr	(0.0 - 1.0)	Weight for zero crossing rate feature
lms	(0.0 - 1.0)	Weight for log mean square amplitude feature
s_c	(0.0 - 1.0)	Weight for spectral centroid feature
s_t	(0.0 - 1.0)	Weight for spectral tilt feature
time	(1.0 - 5.0)	Time in seconds into the past to start searching for matches
dur	(1.0 - 5.0)	Time in seconds from seektime towards the present to test matches
lens	(0.0 - 0.1)	Match length in seconds (this will be rounded to the nearest FFT frame)
rand	(0.0 - 0.8)	size of source store sample buffer in seconds

For details and demo codes please check the concat2 document.

7. Equalizer

Equalizer panel provides a 10-band EQ effect with central frequencies placed in octave (36Hz, 75Hz, 157Hz, 329Hz, 688Hz, 1440Hz, 3013Hz, 6303Hz, 13184Hz, 18000Hz) and a unite Q-factor control slider for adjusting the bandwidth of every band-pass filter. The gain and attenuation level for each band is from -24dB to 24dB.



8. Reverb

Reverb panel is a high-quality algorithmic reverb effect with 8 adjustable parameters. It can also be used as a chorus-like effect by adjust dpth and freq parameters. Dry/wet control and reset function are implemented for using convenience.

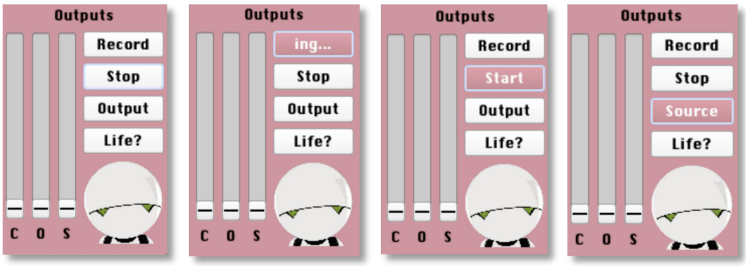
Name	Range	Description
T_60	(0.1 - 10)	approximate reverberation time in seconds
damp	(0.0 - 1.0)	controls damping of high-frequencies as the reverb decays.
size	(0.5 - 5.0)	scales size of delay-lines within the reverberator, producing the impression of a larger or smaller space. Values below 1 can sound metallic.
diff	(0.0 - 1.0)	controls shape of early reflections. Values of 0.707 or more produce smooth exponential decay. Lower values produce a slower build-up of echoes.
dpth	(0.0 - 1.0)	Depth of delay-line modulation. Use in combination with **freq**to set amount of chorusing within the structure.
freq	(0.0 - 1.0)	Frequency of delay-line modulation. Use in combination with dpth to set amount of chorusing within the structure.
lowcut	(100 - 6k)	Frequency at which the crossover between the low and mid bands of the reverb occurs.
highcut	(1k - 10k)	frequency at which the crossover between the mid and high bands of the reverb occurs.

For details and demo codes please check [the JPverb document].

9. Output Control

Output Control panel gives methods to control audio stream and save generated sounds. The left three sliders with labels C,O and S are the gain-controllers of control, output and source signals[2]. The processed audio can be recorded into file in wav format in recordings folder inside the project by Record. Also the sound can be muted by Stop/Start switch-button. Or the user can listen to all three kinds of signal by switching the status of Output/Control/Source button.

The most important part: remember to read what Marvin said~



[2] The Output Gain is connected to the final output audio stream not the output signal self. So If you are listening the C or S sound the output sound level is control by O slider. C and S serve only the original signal. I know it is little bit weird.

10. Demos

Maybe I will write tracks or build a sample packet by this in the future.
Or just send me something you created that inspired by this tool :).

End