Ricchiuti Mauro

Technical Specifications

The requirements and set-up

The system is built mainly in 2 software: TouchDesigner and VCVRack. However, it makes uses of different modules and drivers to get around some constrains. Here you find a list of what you need to run the system:

- TouchDesigner
- VCVRack
- MIDI Virtual cable (I have used LoopMIDI, an alternative could be LoopBe).
- 2 Audio Virtual Cables (I have VB-Audio and VAC on windows 10. For macOS, you could try soundflower, JACK etc.)

In order to run my VCVRack patch you would need the following libraries:

- Basic VCV modules
- Unless modules
- Valley
- Vult
- Befaco
- NYSTHI Fixed Voltage Source

Additionally, if you want to train your own Markov Chain you could need ML Modules and some rewiring from the MIDI-CV, mr. Chanikov and the OctaFlop.

Here I post all the links to download the required softwares/drivers:

https://derivative.ca/download

https://vcvrack.com/manual/Installing

https://www.tobias-erichsen.de/software/loopmidi.html

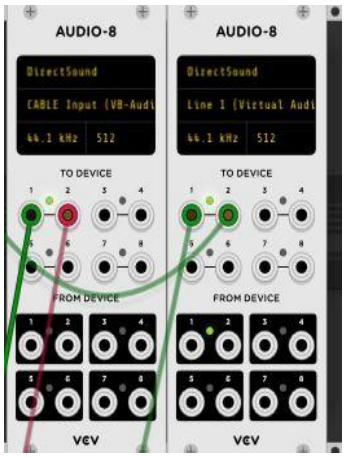
https://vb-audio.com/Cable/

https://vac.muzychenko.net/en/download.htm

Then to install the VCV Libraries you should:

- Register for a VCV account and log in using Rack's "Library" menu.
- Subscribe to a plugin, relaunch Rack, and click "Library > Update all" in the menu bar.

After this setup, make sure to have the virtual cables set properly. This means that loopMIDI should be your device 0 in your computer, the 2 Audio Modules in VCVRack should be output, through your audio driver on CABLE(VB-Audio), the left one, and Line1(VAC), the right one. In the following pictures, you can see what you need to check.



If you install CABLE and VAC properly, they should work by themselves without changing anything. However, I am writing this because there could be some problems. If you run the systems on a macOS for example, and some drivers are not available, there would be needed some changes. The important thing is that the left audio device in VCVRack (here right picture) is connected with the same audio cable as the audiodevin1 operator in TD. The same is valid for the right Audio device in VCVRack which should be sending output to Line1, while the audiodevin2 should receive from Line1. So, left Audio Device in VCVRack sends to audiodevin1 through CABLE, while right VCVRack Audio decive sends audio to audiodevin1 through Line 1 cable. It is really important to keep these cables or everything would be messed up.

How it works

In the figure 1 below, a block diagram is used to describe the system.

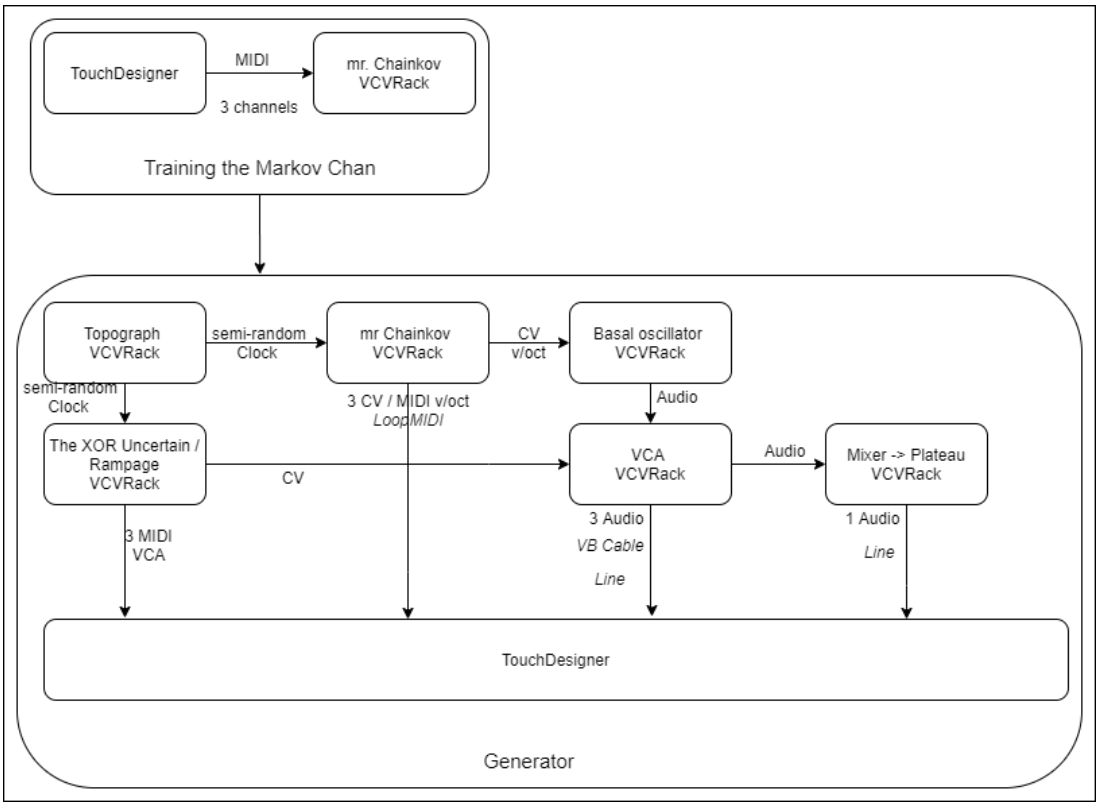


Figure 1 Block Diagram

The system works in 2 steps. First MIDI data are sent to mr. Chainkov in VCVRack to learn the distribution probability of the notes. Then, a rewiring is needed, and the notes are played following the probabilities.

I provide 2 VCVRack. In the file “Learn”, it is provided the wiring to teach the probability distribution to the mr. Chainkov module of VCVRack. When a new MIDI input is used so, the patching should look as in this file. In TD, is needed to activate the MIDI sending, going inside the “MIDI_sender” component, activating every operator, and waiting till the whole song is sent. Some adjustment on the beat frequency would be needed to match the tempo of the sent MIDI track. Make sure to have untoggled all the MIDI send/receive in both TD and VCVRack apart from the MIDI_sender in TD and the 3 MIDI-CV modules on the left side of VCVRack screen. If no signals would be sent to the 2 top inputs of the mr Chainkov module (learn and forget), the module would learn all the probability. However, this could be changed to selectively teach notes, for example if a live input is provided.

When the probability is learnt, we could play it. Have a look at the “Play” vcv file. Here the mr. Chainkov modules, are triggered by the Topograph, a semi-random clock, which has 3 knobs that would control the speed of the clock signals. Two other knobs, swing and the one with the numbers could be used to play the clock signals. The CV output from the Chainkov, would follow the probability distribution and would be used as V/Oct value for 3 simple oscillators from Vult named Basal. The output audio from these oscillators would be sent to a VCA which is controlled by another pseudo-random CV singal. The outputs of the VCA are mixed together and sent to TD. Also, the 3 CV-MIDI modules on the right side of the screen, would send the 3 MIDI notes from the mr. Chainkov, as well as the CV which is used to modulate the VCA. Additionally, the 2 Audio-8 modules would send each of the 3 audio tracks separately to TD and a mixed track too.

When the data are sent to TD, they are analyzed and different features are extracted. The note played in each channel is detected. Is done an FFT of each separate audio channel, and the frequency with the highest amplitude is taken. The CV from the VCA is selected. These values are then used to change the color, scale and translate the different visualizations. The provided TD file would have a first interface with a slider named “Switch” with which should be possible to switch between the different visualizations. With the button “Play”, the mixed audio coming from VCVRack could be played and listened. The button named “?” could be used to record all the visuals and the audio together, however here is de-activated. Finally, in the main page of TD, the one which is shown when the file is opened, is shown the “inspiring note-color mapping” from Scriabin’s model. Below it, is shown my reproduction of said colors, and on its left the table holding the note-color mapping RGBA values. On the right side of it, a component called MIDI_sender is de-activated, and it should be activated if a new MIDI file is used to teach new Markov Chain probabilities.

I’m handing-in, together with this file, a video tutorial explaining the system in more details.