Software Modular Synthesizer

Computer Sound Production Laboratory Assignment Topic Submission

Gregor Kovač
Faculty of Computer And Information Science, University of Ljubljana
Email: gk1854@student.uni-lj.si

I. Introduction

Synthesizers are electronic musical instruments that generate audio signals [3]. They do this by producing a waveform, like a sine wave, and then shape it using different techniques. The first proper analog synthesizers were created in the 1960s [2], but they were replaced by digital synthesizers in the 1980s. As of then, they have been widely spread in the computer sound production world. Here we will focus on *modular synthesis* - a popular technique in the analog domain, where one takes a collection of different smaller synthesizers called *modules* and connects them together with cables to route the audio signal. This allows for a lot of creativity and freedom in creating new interesting sounds. There exists many different modules ([1] presents some of them) which makes this field of sound production complex. On top of that, building such a system can be rather expensive. We will be designing a simpler all-in-one software alternative to a modular synthesizer.

II. Goals

The goal of this project is to build a standalone application that simulates modular synthesis. Some software alternatives like this already exist, like N*ative Instruments Reaktor*[6] and *VCV Rack*[7], but they seem complex and intimidating at first, just like usual modular synthesizers. The purpose of our app will be exploration, creativity and learning, therefore we will strive to develop a beginner-friendly learning curve.

III. Technologies

For developing the application we intend to use the programming language *Python* with the help of the libraries *PYO* [4], a library for digital sound production, and *MatPlotLib* [5], a library for visualizations. We may change our toolkit during the development process according to our needs.

IV. Implementation

Here we provide a list of descriptions of all features we plan our application to have. We may add more in the future.

1) **User interface.** The user will be presented with a blank workspace. There will be a menu to choose modules from. The user will be able to add a module to the workspace and move it around. Modules will consist of input and output pins for controlling the signal flow and buttons and knobs for controlling different parameters. The user will be able to connect pins with "cables" to route the signal. The application wil also feature optional tooltips to guide a new user towards creating their first sounds.
2) **Modules**. Synthesizer modules will be developed using object oriented programming to allow for easy implementation of new ones. The signal flow will be simulated live and we will also be able to change parameters during simulation. Here is a basic list of modules that we plan to implement, but more may be added later:
   a) *Voltage Controlled Oscillator* (*VCO*) produces basic waveforms (sine, square, sawtooth, triangle).
   b) *Noise* can output white and pink noise.
   c) *Low Frequency Oscillator* (*LFO*) generates waveforms at a lower frequency, used to alter parameters and add dynamics to sound.
   d) *Envelope Generator* (*EG*) generates a "one-shot" signal that is shaped using attack, decay, sustain and release (ADSR) and shapes the beginning and the end of a sound.
   e) *Voltage-Controlled Filter* (*VCF*) simulates standard filters (low-pass, high-pass, band-pass, band-stop).
   f) *Voltage-Controlled Amplifier* (*VCA*) controls the volume of an input signal.
   g) *Sequencer* outputs a rhythmic sequence of frequencies (a melody).
   h) *Mixer* has a few inputs and blends them and sends them to the output like the computer's sound card.
   i) *Visualizer* features visualizations of an input signal, like a frequency graph and an oscilloscope.

V. Evaluation

After testing the visualizer module with basic waveforms, we will use it to visually test other modules. Evaluation will also be done by ear, to hear if some patch makes sense. The intuitiveness of the application may be tested by giving it to a few users and collecting their opinions.

References

[1] James, A., *The Secret World Of Modular Synthesizers*, Sound On Sound, 2013. https://www.soundonsound.com/reviews/secret-world-modular-synthesizers
[2] Analog Synthesizer, Wikipedia, 2024. https://en.wikipedia.org/wiki/Analog_synthesizer
[3] Synthesizer, Wikipedia, 2024. https://en.wikipedia.org/wiki/Synthesizer
[4] PYO, Ajax Sound Studio, 2021. http://ajaxsoundstudio.com/software/pyo/
[5] MatPlotLib, 2024. https://matplotlib.org/
[6] Native Instruments, *Reaktor 6*, 2015. https://www.native-instruments.com/en/products/komplete/synths/reaktor-6/
[7] VCV, *Rack 2*. 2021. https://vcvrack.com/