Developing a Music Sequencer/Synthesizer

Victor Shepardson
Computer Systems Lab 2009-2010

Abstract

A software audio synthesizer is being implemented in Python, capable of taking musical and nonmusical input information and using additive and FM methods of synthesis to achieve rich spectra, vibrato, tremolo, and smooth pitch change effects.

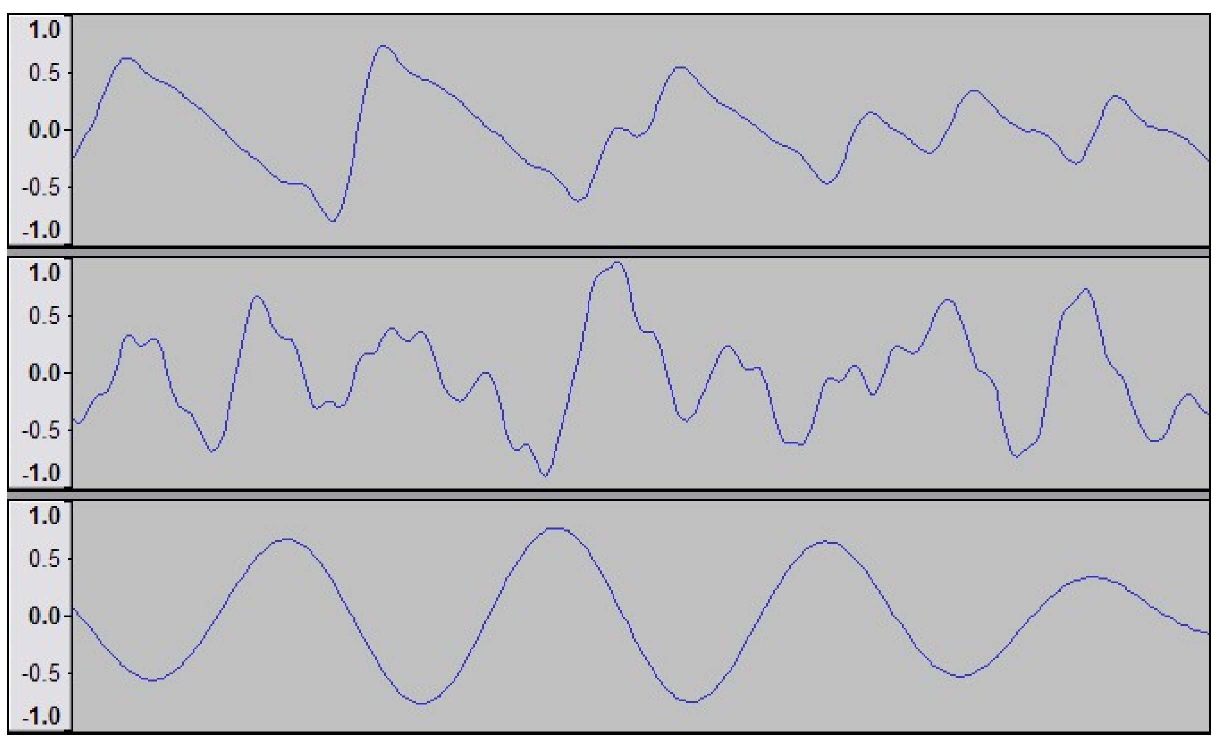

Fig 1: waveforms

Background and Introduction

Electronic sound synthesis has been of interest to musicians, electrical engineers and computer scientists for as long as it has been practical. Synthesizers have evolved from primitive analog machines to sophisticated computer programs.

An audio signal can be represented by a collection of sine waves with different phase, frequency and amplitude called a spectrum. Additive synthesis creates audio signals by summing together sine waves or by using Fourier Transforms to turn spectra in audio signals. One implementation of additive synthesis--the one used in this project--is to use multiple digital oscillators to generate harmonic tones, and to sum their outputs.

Frequency Modulation (FM) synthesis can produce a rich spectrum from just one tone by using it to modulate the the frequency of a second oscillator operating at a carrier frequency. By varying the harmonic relationship between the modulating frequency and carrier frequency, and the amplitude of the modulating signal, output signals which are spectrally rich and dynamic over time can be produced.

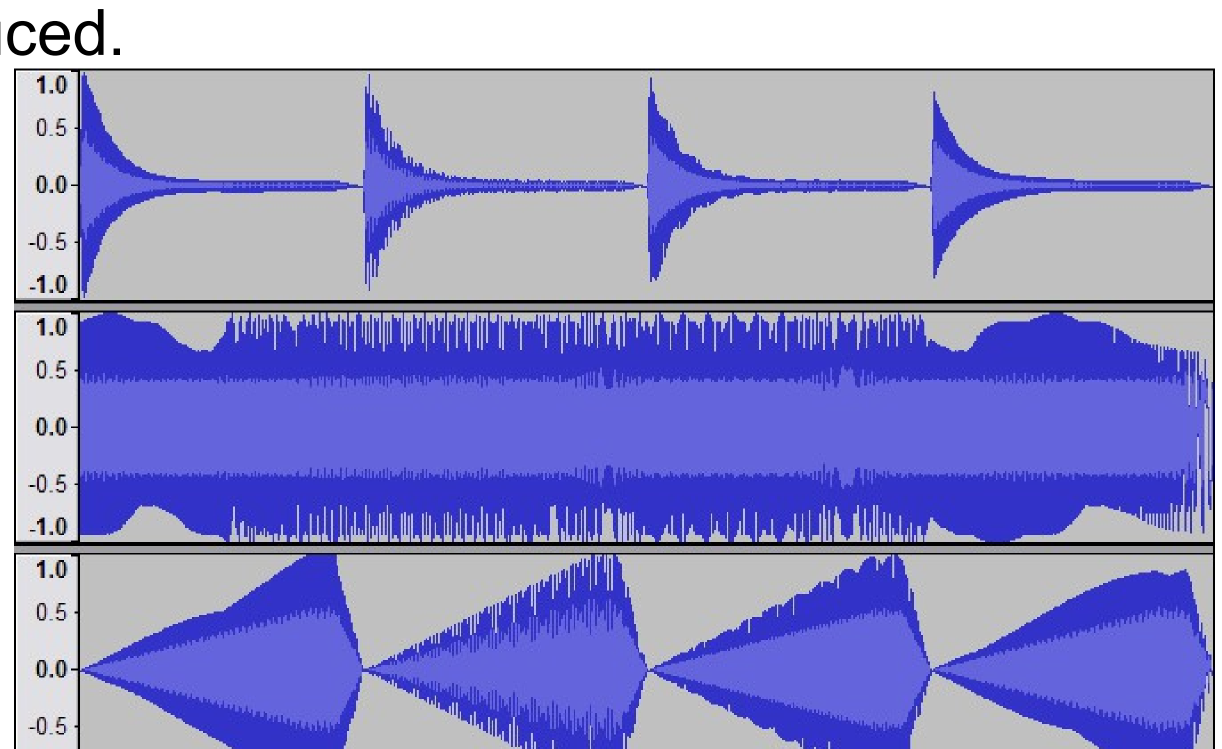

Fig 2: envelopes

Discussion

A final version will consist of a GUI allowing the creation and interconnection of several elements: oscillators, waveforms, amplitude/frequency functions, mixing blocks, and note matrices. Oscillators repeat waveforms periodically given frequency inputs, and then apply amplitude inputs. Mixing blocks can take two or more input signals and a control signal and output some function of those signals, for example, an average of two inputs weighted by the control signal, or one input scaled by a control signal and added to a second input. Note matrices are representations of musical notation which can be used to generate functions. Using a simple notation developed for this project, a user can input musical information; a note matrix can then produce piecewise frequency functions corresponding to the specified pitches, and piecewise amplitude functions corresponding to information about envelope shape and note durations.

Any number of oscillators can be created, allowing additive synthesis by the addition of pure tones, or implicitly by the use of different waveforms. Sub-audio band frequency and amplitude functions can create vibrato and tremolo effects; note matrices allow notes to be played legato or staccato, or to swell in or fade out. By linking the outputs of oscillators to the frequency or amplitude inputs of other oscillators FM and AM can be implemented.

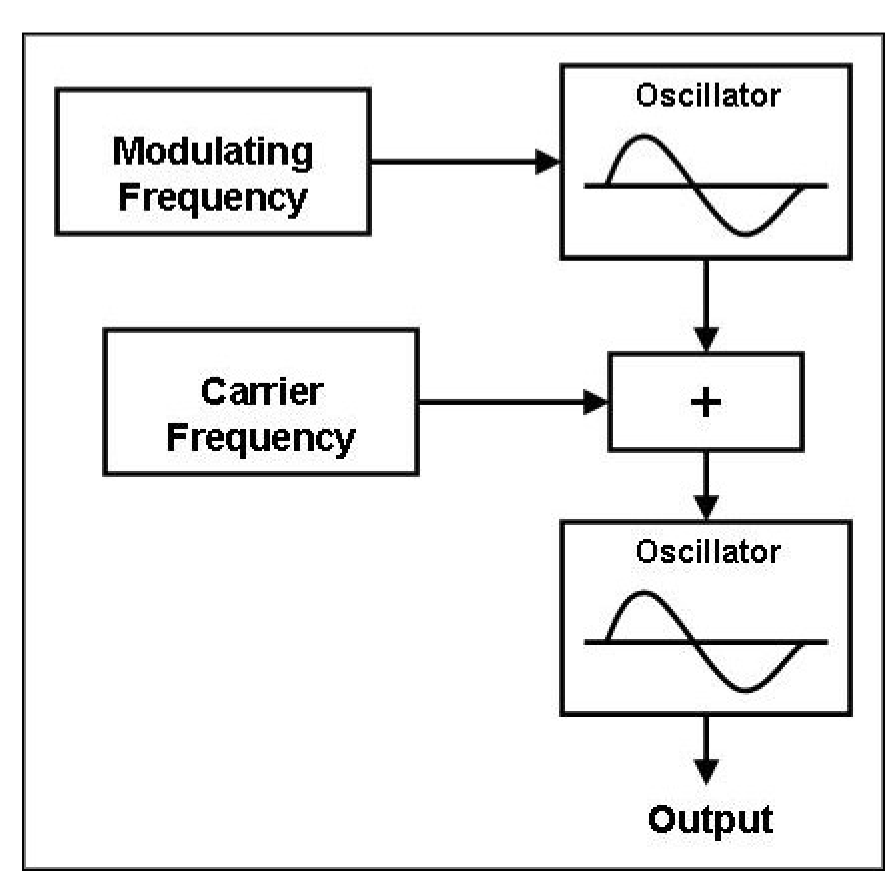

Fig 3: FM implementation

Results and Conclusions

The synthesizer is producing wav files with the intended content; smooth pitch transitions and properly shaped envelopes are evident. Feedback from oscillator outputs to other elements has yet to be implemented; only additive synthesis is operational.

The employed methods of sound synthesis are capable of producing a wide range of subjectively interesting tones.