Developing a Music Sequencer/Synthesizer

Victor Shepardson
Computer Systems Lab 2009-2010

Abstract

A non real-time sequencer/ synthesizer is being developed in Python, capable of producing audio signals made up of superposed periodic functions. Smooth pitch transition, vibrato, tremolo, envelope shaping, and polyphony have been implemented. The program performs as intended so far.

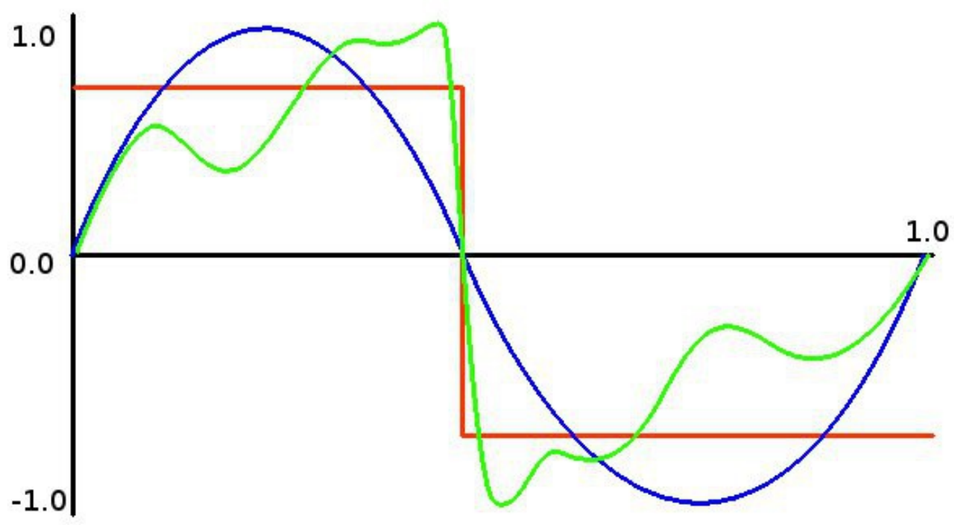

Fig 1: waveforms

Background and Introduction

Currently the only language in use is Python. Knowledge of Python was preexisting, as was basic knowledge of the digital representation of sound and of synthesis by superposition. To date, areas of study have been sound file formats, scientific pitch notation, and methods of synthesis. Articles discussing granular synthesis and synthesis by cross-coupled oscillators were examined; similar methods may be implemented at some point. An interface to communicate with a composition program called MuseScore might also be developed.

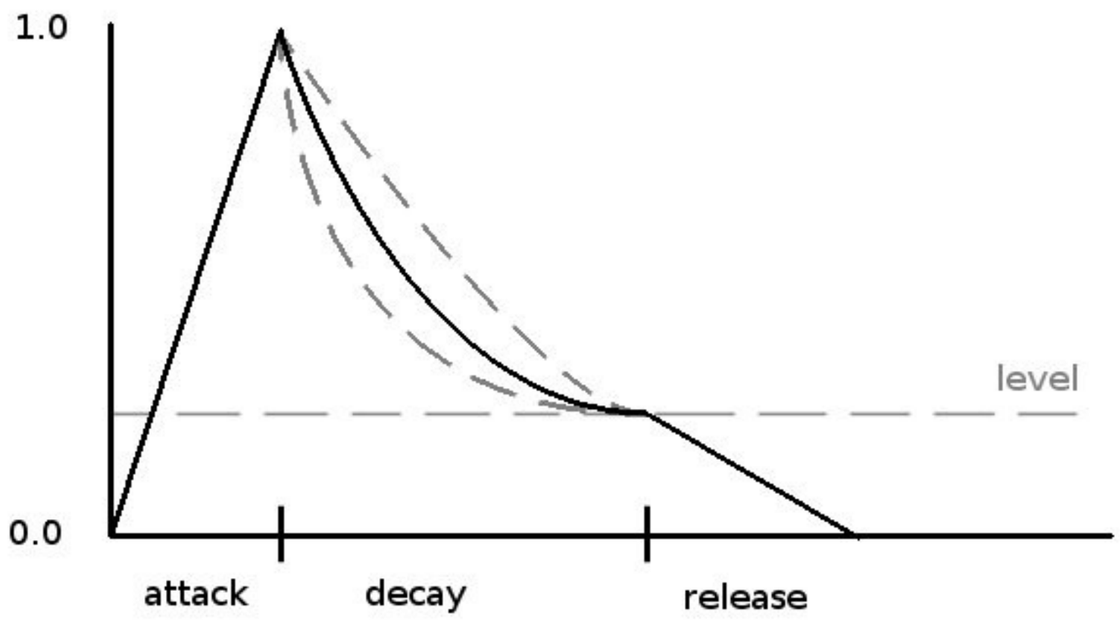

Fig 2: The envelope function

Discussion

At the core of my synthesizer is an oscillator. It works not as a function of time, but as a function of a parameter p which varies from 0.0 to 1.0. This function can be any specified to be any shape; a sine wave, a square wave, a harmonic series. Different pitches are achieved by incrementing p based on a frequency function. The result is that the audio signal will be continuous (as long as the wave form is) whether or not the frequency function is. A consequence is that any voice must be monophonic; polyphony can be achieved only by using multiple voices. Multiple frequency functions can be specified per voice; they will be multiplied together to produce a single value. In this way it is possible to achieve vibrato, whammy and ring modulation effects. The output of the oscillator is multiplied by each amplitude function; this enables envelope control, tremolo, and longer term crescendos and fade outs.

Music can be input via the note matrix: note objects in a list have properties of pitch, position in time, duration, and envelope parameters. I have written frequency and amplitude functions which will build themselves based on these parameters to match the note matrix. Once a sequence of notes is programmed in, the harmonic content and envelope shape can be changed; notes can be played legato or staccato, can swell in or decay exponentially like a plucked string. Any of the previously mentioned tremolo, vibrato and ring mod effects can be applied.

Results and Conclusions

The synthesizer is producing wav files with the intended content; smooth pitch transitions and properly shaped envelopes are evident. Polyphony is functioning but may still exhibit issues related to phase of voices in unison. Noise is not excessive and seems to be properly linked to sample depth of output files. Currently, speed lags behind real-time, and is probably nowhere close to optimal.

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