The Digital Oscillator

Good Vibrations

- The oscillator is one of the most basic building blocks in synthesis
- An oscillator generates periodic waveforms
- Basic controls include amplitude, frequency, waveform type and phase
- The output of an oscillator is a sequence of samples which forms a digital signal representing the waveform (Dogde - Jerse)
- Two methods for digital oscillator implementation:
  - 1. <u>Direct Evaluation</u>: Calculating the value of the mathematical function for every sample. Slower, still.
  - 2. <u>Wavetables</u>: A stored sequence of numbers corresponding to the the successive points of the waveform. Usually a power of 2 in size. Much more efficient

Table-lookup synthesis.

- The oscillator maintains a numerical value - the phase (aka index), indicating the address of the value currently in use.
- On every sample the oscillator algorithm obtains the current value of the phase and adds to it an amount that is proportional to the frequency of the oscillation (Dodge - Jerse)
- The frequency produced by the table-lookup oscillator depends on the length of the wavetable and the sampling frequency
- If the sampling frequency is 1000 samples/sec and the table has 1000 entries the result is 1000/1000 = 1 Hz

Table-lookup synthesis.

- In order to produce a different frequency the sampling frequency can be adjusted:
- If sampling frequency = 100 000 hz, frequency = 100 000/1000 = 100
- Frequency = **Sampling frequency / table size**

- A better method is to scan the wavetable at different rates, skipping some samples, in effect changing the size of the wavetable in order to generate different frequencies
- In order to raise the pitch by an octave we can skip every other sample
- The increment determines the number of samples to be skipped, and is added to the current phase location in order to find the next location
- **Phaser Increment = (table size * frequency) / Sample Rate**

Table-lookup synthesis.

- In effect, the oscillator re-samples the wavetable in order to generate the desired frequency
  (CMT p 93) Dodge p67
- Often the oscillator will end up with a phase index with a fractional part while the entries in the wavetable are all perfect integers:

i.e.: for a wavetable of 512 entries, at 40KHz SR, a 440 Hz tone would require a sample increment of 5.632 (Dodge-Jerse)

- There are three ways of dealing with the fractional part:
  1. Truncation (worse)
  2. Rounding (better)
  3. Interpolation (best)

Table-lookup synthesis.

1. Truncation simply ignores the decimal part of the sample increment. (fast, but not very accurate)

2. Rounding simply rounds up the value of the sample increment to the nearest integer. (slower, and still not that accurate)

3. Interpolation gives the best results. The ideal value of the wavetable is calculated and then approximated as a weighted average between the two entries in the wavetable where it falls.

   "…this process can be thought of as taking the waveform value on a straight line that connects the values of successive wavetable entries."

   (Dodge-Jerse p68)

Table-lookup synthesis.

- The errors introduced by the fractional parts manifest themselves as distortion in the signal (quantization noise)
- Interpolating oscillators have the greatest signal to noise ratio, but a non interpolating oscillator can also generate good results with a large table size. Using a non interpolating oscillator with a large table size is more efficient than using an interpolating oscillator
- Csound offers a variety of interpolating oscillators, which have the letter i added to their names: oscili, foscili etc…

Phasors and Tables

- We can create our own custom oscillators in csound with the phasor and table opcodes:

A function table is essentially the same as what other audio programming languages call a buffer, a table, a list or an array. It is a place where data can be stored in an ordered way. Each function table has a **size**: how much data (in Csound just numbers) can be stored in it. Each value in the table can be accessed by an **index**, counting from 0 to size-1. For instance, if you have a function table with a size of 10, and the numbers [1.1 2.2 3.3 5.5 8.8 13.13 21.21 34.34 55.55 89.89] in it, this is the relation of value and index:

VALUE	1.1	2.2	3.3	5.5	8.8	13.13	21.21	34.34	55.55	89.89
INDEX	0	1	2	3	4	5	6	7	8	9

So, if you want to retrieve the value 13.13, you must point to the value stored under index 5.

The use of function tables is manifold. A function table can contain pitch values to which you may refer using the input of a MIDI keyboard. A function table can contain a model of a waveform which is read periodically by an oscillator. You can record live audio input in a function table, and then play it back. There are many more applications, all using the fast access (because a function table is part of the RAM) and flexible use of function tables.

From the Floss Csound Manual: http://en.flossmanuals.net/_booki/csound/csound.pdf

Phasors and Tables

- We can create our own custom oscillators in csound with the phasor and table opcodes

- phasor — Produce a normalized moving phase value.
- ares **phasor** xcps [, iphs]

- An internal phase is successively accumulated in accordance with the *kcps* or *xcps* frequency to produce a moving phase value, normalized to lie in the range 0 <= phs < 1.

- When used as the index to a *table* unit, this phase (multiplied by the desired function table length) will cause it to behave like an oscillator.

Phasors and Tables

```
instr 1

        idur  = p3
        iamp  = p4
        ipch  = cpspch(p5)
        irate = p6

        aphasor phasor irate
        atable  table  aphasor * 8192, 2
        aosc    oscil  iamp, ipch, 1, -1
        aosc    =      aosc * atable
        out aosc
Endin

f1 0 8192 10 1 0 0 0 0 0 .5 0 0 .3 0 0 0 .2 0 0 .1
f2 0 8192 -7 0 192 1 200 .6 7800 0

i1 0  2 10000 7.00  4
i1 +  .       7.05  3
```

Csound Oscillators

• Simple table oscillators:

Oscil- A simple oscillator, no interpolation.

Oscil3 - A simple oscillator with cubic interpolation.

Oscili - A simple oscillator with linear interpolation

• Precision Oscillators:

poscil — uses floating-point table indexing, instead of integer math, like *oscil* and *oscili*

poscil3 — High precision oscillator with cubic interpolation

Csound Oscillators

- Flexible oscillators:

oscilikt — A linearly interpolated oscillator that allows changing the table number at k-rate. (similar to oscili)

osciliktp — A linearly interpolated oscillator that allows allows phase modulation.

- Others…

pulse — Generates a set of impulses.

vco — Implementation of a band limited, analog modeled oscillator.

vco2 — Implementation of a band-limited oscillator using pre-calculated tables.

Notes: Oscillator Sync

When an oscillator is sync'ed, it follows a master oscillator (the slave oscillator may or may not be tuned to the master's frequency)

Every time the master oscillator's cycle repeats the slave oscillator is re-triggered regardless of its position.

If the slave is tuned to a lower frequency than the master, it will re-trigger before a full cycle has been completed

If the slave oscillator is tuned to a higher frequency than the master it will re-trigger partway through a cycle.

Sync'ing oscillators forces them both to the same frequency but the irregular slave cycles create harmony and aliasing.