'Continuous-Time' Synthesis and the Context of Timelab'

March 16, 2014

1 Introduction

1.1 why timelab?

1.1.1 issues with existing systems

Music-N/CSound (Max Mathews, Barry Vercoe, John ffitch, and others 1957-present) A classic programming environment, stemming from the pioneering work of Max Mathews, it is widely used, and highly developed.

- rigid structure
- difficult to interact with / compile time
- obtuse representation of scores and synths / hard to read (pull up examples)
- score/instrument layout insists on antique perception of music composition/creation – a notion which in many ways goes against the grain of analog electronic music practices

Max/Msp and Pd (David Zicarelli and Miller Puckette, 1987-present) An almost ubiquitous and highly interactive paradigm, these programs the are industry standards (commercial and open source respectively) for experimental audio algorithm design.

- patch chords and guis!!!
- difficulty of iterating loops
- difficulty in doing things in mass
- flow diagram structure (borrowing from analog synthesizers) also insists on a way of thinking about music composition – denying the user a 'code-ish' mindset

SuperCollider (James McCartney, 1996-present) SC provides the interactiveness of Pd with the 'codiness' of a text based programming environment such as CSound.

- server/client structure has benefits but designates a .2s default latency between server time and client time
- non-standard editor and gui environments

ChucK (Ge Wang and Perry Cook) ChucK is another text-based audio programming environment that emphasizes a live-coding ethos.

- admittedly emphasizes ease of programming over performance (ChucK is slow)
- unintuitive approach to time by treating 'now' as a variable that we manipulate by hand: `http://en.wikipedia.org/wiki/ChucK`

1.1.2 Fall 2012 206

1.1.3 Control timing

All of the above metnioned systems have specific solutions to problems of rectifying the timing between control input and precisely when control is enacted in the DSP engine. Cf. my poster from last year's ICMC (TIMELAB: YET, YET ANOTHER REAL-TIME AUDIO PROGRAMMING SYSTEM – not yet available online) for a quick and dirty on this subject.

Also of relevance are these talks by Miller Puckette and James McCartney: `http://repmus.ircam.fr/mutant/rtmseminars`

- Puckette talk: 9:30 - 15:00 : DAGs and mutual exclusion, determinism
- McCartney talk: 29:00 - 37:45 : control rate issues

1.1.4 embedded audio programming

Just the other day (Friday, March 14) someone posted to the Pd-list asking if it was possible to crunch a Pd patch down into a guitar pedal (the answer was 'no, not really'). Clearly, there is a market for throwing these kinds of experimental algorithms into a guitar pedal packages. One solution is to utilize micro computers such as the Raspberry Pi or UDOO. Another is to have an audio programming API in C with a build environment that can target truly embedable hardware, such as ST's discovery board featuring the ARM Cortex processor (or any of the open source solutions built around these chips).

2 Continuous-Time Synthesis

What is meant by 'continuous-time synthesis'

- as distinct from traditional computer music DSP structure which consists of UGens and signal flow
- use differential equations to model synthesis behaviors

This affords us advantages that are beyond the scope of UGen/signal flow structures. The programmer may create complex synthesis algorithms by arranging equations in a network with interleaved numerical solver stages so that time does not pass between nodes in the network. In other words, 'delay free loops', or 'instantaneous feedback' is acheivable. The mathematical necessity of unit delay is currently a limitation of the sate of the art in computer music systems and synthesis theory.

2.1 sync functions

Delay free feedback in a network can fascilitate (among other things) accurate oscillator sync. In the parlance of analog synthesizers, a master oscillator may slave another one to it by resetting the phase of the slave at every other zero crossing (each period) of the master.
red function slaves blue function
misaligned reciprocal sync

Sync functionality is but one benefit of delay free loops. In general having instantaneous state updates from stage to stage in a synthesis algorithms allows for a general solution to numerous problems associated with Virtual Analog.

2.2 virtual analog

Virtual analog (VA) is the practice of emulating the sound of analog synthesizers with DSP algorithms. This is desirable because:

- analog gear is heavy, and expensive
- analog gear is sensitive to heat and requires constant tuning and maintainance
- polyphony is 'expensive' in analog and 'cheap' in digital
- digitization means patches and states can be saved and easily re-accessed

But, analog 'sounds better'. It is precisely the inexactness of analog and the nature of circuit components that give analog equipment its richness. Challenges in VA include:

- aliasing due to the Nyquist theorem

- parameter (state) change quantization
- unit delay fudge-factors

Although it has been around since Synergy (1981), and the term was brought into commercial use nearly 20 years ago with the introduction of the Nord Lead 1 synthesizer (1996), VA is still a very active area of computer music research.

There are numerous papers and book chapters that propose novel algorithms and improvemnets to existing algorithms in VA. Many address problems in problem-specific ways. There are a also a few general solutions to these problems.

2.2.1 ngspice

Needs be mentioned: ngspice is the current open source edition of Berkeley's spice (Simulation Program with Integrated Circuit Emphasis – Nagel, 1971). Spice and its descendants allow users to construct virtual integrated circuits and the algorithm will simulate the behavior of the circuit by using numerical solvers. It is not real-time, nor is it intended for audio synthesis.

2.2.2 wave-digital filters

WDFs are an approach to circuit emulations that are attractive because:

- they are modular
- circuit component behavior is easily discretized with the bilinear transform
- they are computationally cheap, thus networks of very many of them are readily available

However, the trick of designing WDF networks often lies in dealing with delay free loops (to which often unit delay is the given solution) and non-linearities. The continuous time synthesis model comletely does away with these complications, but at the price of increased complexity in the domain of physical model design.

3 Timelab

3.1 Brief Overview

3.2 Examples

3.2.1 quadrature oscillator

3.2.2 non-linear oscillators

These are often good as LFOs due to their controlled chaotic behavior:

- van der Pol oscillators

- Duffing oscillators
- other noisey oscillators

3.2.3 moog filter example

3.2.4 larger synth example

3.3 new models for new sounds

3.3.1 noisey oscillators

3.3.2 non-linear filters

3.3.3 a new way to view synthesis