MUS420/EE367A Lecture 1
Overview, Outline, Demos, and Administrative Info

Julius O. Smith III (jos@ccrma.stanford.edu)
Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
Stanford, California 94305

March 31, 2008

Music 420 Overview

Computational acoustic modeling for digital audio effects and sound synthesis

Topics:

- sampled traveling waves
- acoustic simulation using delay lines, digital filters, and nonlinear elements
- comb filters, allpass filters, and state-space models
- delay-line interpolation and sampling-rate conversion
- phasing, flanging, and chorus effects
- finite difference schemes and associated filter design
- computational models of selected musical instruments, such as
  - plucked, struck (piano), and bowed strings
  - woodwinds (primarily the clarinet)
  - flute and organ pipes
  - brasses

and, time permitting,

- waveguide meshes for simulating vibrating membranes and acoustic volumes
- wave digital filters
- virtual analog

(This overhead presentation is adapted from Handout #1: MUS420 Administrative Info[1].)

[1] http://ccrma.stanford.edu/~jos/intro420/

Applications

- Audio signal processing (filter design, structures)
- Virtual musical instruments (computer music)
- Music synthesizers (hardware and software based)
- Physical models for digital control
- Model-based audio coding (MPEG-4 / SAOL)
- Musical acoustics (put theory to the test!)
- Numerical methods for solving differential equations

Physical Modeling Synthesis Sound Examples

- 2006 AES Masterclass presentation entitled "Physical Modeling Sound Synthsis" (with live sound-example links in the PDF): `http://ccrma.stanford.edu/~jos/pdf/AES-Masterclass.pdf`
- Sound-examples page from class textbook: `http://ccrma.stanford.edu/~jos/pasp/Sound_Examples.html` (mostly same as the masterclass examples, but more conveniently downloaded, bookmarked, etc.)

Administrative Information

Web site:
http://ccrma.stanford.edu/CCRMA/Courses/420/

Units

- Tuesday-Thursday lectures + assignments = 3 units
- Final project = 1 unit (optional 4th unit)
- Grading

Text

The text for this course is

Physical Audio Signal Processing[2]

- Available online in HTML format
- Hardcopies available from the TA
- Reading assignments specified in the *Course Schedule and Outline*[3].

[2] `http://ccrma.stanford.edu/~jos/pasp/`
[3] `http://ccrma.stanford.edu/~jos/intro420/Schedule_Assignments.html`

Prerequisites

- Elementary physics (Newton's laws and associated analysis of dynamic systems)
- Elementary signal processing, as discussed in, for example,
  - **Mathematics of the Discrete-Time Fourier Transform (DFT)**[4] — prerequisite material pertaining to the DFT (Music 320 text)
  - **Introduction to Digital Filters**[5] — prerequisite material in the area of digital filtering and linear systems theory (Music 320 text)
  - McLellan, et al., **DSP First: A Multimedia Approach**, Prentice Hall, 1998 (TK5102.M388). (Former Music 320 text — the CD-ROM is installed at CCRMA in the directory `/usr/ccrma/courses/320/dspfirst/`)
  - Oppenheim, Schafer and Buck, **Discrete-Time Signal Processing, 2nd ed.**, Prentice-Hall, 1999. (Text for EE 264 = "Digital Filtering")

[4] `http://ccrma.stanford.edu/~jos/mdft/`
[5] `http://ccrma.stanford.edu/~jos/filters/`

Final Project (4th Unit)

- Allows *you* control over some course emphasis
- Plan to spend at least three hours per week
- Must be related to lectures or labs
- Based on individual *research* projects or expanded lab assignments
- One-page project spec due by fourth class
  - Email to jos (at ccrma) or hardcopy fine
  - JOS and TA are your "research advisors"
- Final project report is due at 5pm on the last day of finals (hardcopy preferred).
- *Project presentations encouraged during last class*

Project Types

1. Reading and report
2. Programming project and report
3. Mixture of reading and programming ("real research")
   (a) Library research, reading
   (b) Implementation, test, results
   (c) Write-up

Project Ideas

- Study of related fundamental topic such as
  - Acoustics
  - Waves
  - Mechanics
  - Algorithms for string synthesis
  - Algorithms for wind synthesis
  - Digital filter design
  - Wave digital filters
  - Modal analysis and synthesis
  - Finite-element modeling (and the like)
- Working problems in a related textbook (especially one addressing holes in your background)
- Literature survey on topics in musical acoustics such as horns, reeds, pianos, etc.
- Programming project in the Synthesis Tool Kit (STK) (see `ccrma.stanford.edu` for download)
- Programming project in `pd` ("Pure Data")
- Make your own DSSI / LADSPA plugin

- Programming project using `sfront` (MPEG-4 Structured Audio development tool)
- Study of numerical requirements for high-quality musical instrument simulation
- Sound synthesis algorithm development
  - Karplus-Strong algorithm and variations
  - Waveguide synthesis
  - Commuted waveguide synthesis
  - Frequency-domain string simulation
  - Waveguide mesh simulation of membranes and acoustic spaces
- Effects algorithms
  - Reverberation
  - New delay effects
- Comparison of string simulation by *difference-equation* and *digital-waveguide* methods
- Analysis
  - Modal analysis of a guitar or violin body
  - Calibration software for waveguide synthesis
  - Identify tube-amplifier transfer characteristics

Making your own DSSI / LADSPA Plugins

Recommended free software:

- Planet CCRMA conveniently installs
  - `jack` sound server
  - `rosegarden` music sequencer (nice DSSI host)
  - DSSI plugins (`/usr/lib/dssi/`)
- `dssi-0.9.1` — Contains DSSI spec, examples:
  - `trivial_synth.c` — good starting point for writing a new plugin
  - `less_trivial_synth.c` — adds an editor GUI
  - `trivial_sampler.c` — sampling synthesis
  - `jack-dssi-host` — example plugin host (see `ghostess` below)
- WhySynth — Sean Bolton's latest DSSI synth plugin
- `ghostess` — Sean Bolton's extension of `jack-dssi-host` to provide GUI editor support — nice simple host for testing and debugging of a new plugin in `gdb`
- See jos sysadmin blog[6] for more things to consider installing beyond Fedora Core 4 and Planet CCRMA.

[6] http://ccrma.stanford.edu/~jos/mypc/FC4_Things_Do_Installing.html

Course Website

http://ccrma.stanford.edu/CCRMA/Courses/420/Welcome.html

- Weekly Schedule[7] (including *all assignments*)
- Pointers to lecture notes (overheads used in class)
- Pointers to all course readering assignments (generally a superset of lecture overheads)
- Sound examples
- Related items of interest

(This presentation adapted from Handout #1: MUS420 Administrative Info[8].)

[7] http://ccrma.stanford.edu/~jos/intro420/Schedule_Assignments.html
[8] http://ccrma.stanford.edu/~jos/intro420/