W18. CS489/689 -- Advanced Topics (Computational Sound)
Undergraduate "topics course". Check Quest, to get correct CS489!

Third offering, W18.
Previous offerings W16, W17 on Piazza.

Richard Mann
https://cs.uwaterloo.ca/~mannr/
mannr@uwaterloo.ca

Audience
- 3B or later (CS370, 371, and/or Matlab/Octave experience)
- sound/ music/ math/ electronics interests
- students who want to do independent project and get course credit
- open to Math/ Physics/ Engineering with similar interests

- all necessary material will be developed in class, lectures + demo code

Grading
Lectures with term work,  50%

Term work: Experiments, homework exercises,
in the language of your choice
Matlab/Octave, Python, C, etc..

Student Project       50%
   Alone or in groups
   Option to present your work.

NO FINAL :)
<table>
<thead>
<tr>
<th>W</th>
<th>Topic</th>
<th>Asst</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to acoustics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Transducers (microphones, speakers) electronics and sound measurement</td>
<td>A0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Analog to digital conversion (ADC)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Time domain processing (mixing, reverb, compression)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Fourier transform (FFT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Spectral processing (filtering, analysis, synthesis)</td>
<td>P0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Sampling and dithering</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Digital filtering (FIR, IIR filters)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Synthesis: AM</td>
<td>P1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Synthesis: FM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Other topics (students' choice)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Student presentations ...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Project topic determined/ negotiated during term,
P0 (one page proposal, week 6),
P1 (three pages, algorithm/ data ready, week 7),
Computing platforms

<table>
<thead>
<tr>
<th>PC/Linux</th>
<th>PC/Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB interface</td>
<td>USB interface</td>
</tr>
</tbody>
</table>
| Mann/Vandkerkooy software (Octave) | National Instruments "myDAQ"
  - function generator
  - oscilloscope
  - frequency analyzer |

Others:
- Raspberry Pi.
- Arduino. Teensy 3.6. ARM processor + ADC
Some past projects

Music synthesis (SW modeling of old school analog synths)
Automatic pitch detection (eg., guitar tuner, note transposer)
Analysis of bird calls ("bio acoustics")
Speech processing/ speech recognition
Music indexing (Shazam algorithm)
Recording and frequency analysis of (student's own) singing voice.
Building, testing and evaluating a new plug in for SW package X.
Time-frequency shifting of audio ("vocoder/ auto tune effects")
Realtime Audio effect programming on Arduino
Design/ measure your own loudspeakers

FAQ: Grading projects

Contributions students can make:
- collect your own data
- analyze others' data
- writing new algorithms/ code,
- evaluating others' code on multiple data sets
- building hardware
FAQ: How much work is required?

Aiming for less than Realtime, Compilers, or Graphics. But the choice is up to you ...

Case study, What is Computer Audio.

Behringer XR12 mixer, web and remote interface.
Interface
GUI interface
Dynamic microphone -->

--> Preamp

--> ADC (analog to digital converter)

--> DSP (digital signal processing)
- Equalization,
- Compressor
- Reverb

--> DAC (digital to analog converter)

--> Audio amplifier

--> Speaker

Demos:
- RTAA, whistle vs. rain stick
- EQ
- All this has to happen in <1.8ms.

Audio measurement software
R. Mann and John Vanderkooy (Physics)
Open source
Implemented in Matlab & Octave
Linux, Windows, Mac
(Raspberry PI, not fast enough yet)
Stimulus: Pure tone, white noise, pink noise.

Block diagram (Whiteboard)