DSP for Digital Artists

by Perry R. Cook, PhD

a 3-Day CalArts Intersession Short Course, January 14-18, 2019

Day 1: Basics, Definitions, FIR Filters, LTI Systems, Convolution, Sine Waves

Digital
Sampled/quantized signals
At regular intervals,
hold signal, convert to
digital number.

Signal
Values (pressure, voltage, etc.)
that change as function of time
Continuous: x(t) t = time
Digital (discrete): x(n) n = sample #

Processing
Do various math on
sampled digital signal.
y(n) = F(x(n))
Gain, Filter, Analysis

Delays (t, x, θ)
Acoustics, speed of sound
Causality (and non-causality)
(block and hop sizes)

Sinusoids (sine, cosine, exp)
Fundamental in nature
Solution of $\delta^2 x / \delta^2 t$
"Eigenfunctions of LTI Systems"

Parameters (Gain, θ)
Gains, Weights, phases,
(SRate), time delays, block
(window) and hop sizes

PsychoAcoustics:
(a branch of PsychoPhysics)

	Definition	Physical Correlate(s)	How to compute it
Loudness	Percept, Intensity	Power	Square of signal (RMS) $\Sigma_n x^2(n)$
Pitch	Percept, Low <-> High	Frequency	Cycles per second (spectral peak)

Log Perception and non-linear Scales

Timbre	Difference	Spectrum+	Many factors, centroid, peaks, time

Time & Timing, Tempo vs. Pitch, Resolvability of events (the 30ms boundary)

The Just Noticeable Difference (JND) Smallest perceived (better than guessing) change

Filters:
y(n) = F(x(n)) Operations on Digital Signals

Popular Simple Filters:
- **Linear Gain:** y(n) = g x(n) where g = some constant
- **Moving Average**: y(n) = 0.5x(n) + 0.5x(n-1) (low-pass)
  - High-Pass Cousin: y(n) = x(n) - x(n-1) (digital differentiator)

Also called **OneZero Filter**, Averaging more samples: More Zeroes!!
Also type of **Finite Impulse Response (FIR) Filter**

Sinusoids (Sine Waves)
Mass/Spring Damper, Pendulum, Rotations (Trig!), RLC Circuits

Code and Demos:
```
0-SamplingNMore In Processing,  Demonstrates Sampling, Quantization
Pendulum MassSpring RotateSine Processing Code for Physical Systems
0c-Quantize.ck    0b-PhoneBandwidth.ck Quantization and SRate Demos
ZeroX P           rocessing Code to demonstrate zerocrossings
5-ZCNative.ck 6-ZCUGs.ck 7-ZCPitch.ck Zeroes2.ck ZeroXing "Pitch" Demos
1-PeakNative.ck 1b-PeaksGUI.ck Track (and display) waveform peaks
OneZero.ck FIRNthOrder.ck       FIR Filters
2-SparsePower.ck 3-RMSPowerNative.ck     Sum of Squares Power
```

Assignment: 1) Some PsychoAcoustic Metric: Measure something about you. Use JND Paradigm.

DSP for Digital Artists

a 3-Day Short Course, by Perry R. Cook, PhD
CalArts Intersession January 14-18, 2019

Day 2: More Filters, Feedback Filters, More Sine Waves, Spectrum Analysis, Modal Synthesis

Some More Time-Domain "Pitch" Detection:

AMDF: pick m that minimizes:
$y(n) = \sum_n |x(n) - x(n-m)|$

Autocorrelation: pick m that maximizes:
$y(n) = \sum_n x(n) * x(n-m)$

Impulse Response:

h(n) = y(n), in response to input x(n) = δ (n)
δ (n) = 1 if n==0, 0 otherwise

Transfer Function:

H(f) = Y(f), for every x(t) = sin(2πft)
Digital: H(n) = Y(n), for x(n) = sin(2πfn/SR)

Linear, Time Invariant (LTI) Systems

Linearity: Homogeneity ($\alpha x \rightarrow \alpha y$) and Superposition ($x_1 + x_2 \rightarrow y_1 + y_2$)
Time Invariance: $x(\tau + t) \rightarrow y(\tau + t)$ for all τ

OnePole: (1^{st} Order feedback) y(n) = gx(n) + ry(n-1) NOTE: r <= 1.0 !!!
Set r and g exactly 1.0 yields "Digital Integrator"
Set 0.0 < r < 1.0 and g = (1.0 – r) Low-pass filter
High-pass Cousin: set -1.0 < r < 0.0 High-pass filter

Convolution

$y(n) = \sum_m x(n-m)\, h(m)$ implement Impulse Response as FIR filter

Filters:

y(n) = F(x(n)) Operations on Digital Signals

Popular Simple Filters:
Linear Gain: y(n) = g x(n) where g = some constant
Moving Average: y(n) = 0.5x(n) + 0.5x(n-1) (low-pass)
High-Pass Cousin: y(n) = x(n) - x(n-1) (digital differentiator)
Also called **OneZero Filter**, Averaging more samples: More Zeroes!!
Also type of **Finite Impulse Response (FIR) Filter**

Recursive Filters: **Feedback, Auto Regressive, IIR, Pole(s)**
OnePole Filter: y(n) = g x(n) + r y(n-1) NOTE: | r | < 1.0
NPole Filter: $y(n) = g\, x(n) + \sum_{m=1 to N} -b_m\, y(n-1)$

Super Special Filter, 2-Pole "Resonator" $y(n) = g\, x(n) + 2r \cos(2\pi f/SR)\, y(n-1) - r^2\, y(n-2)$

Modal Synthesis: Modes = natural resonances of system. Excite those and let them ring!

Code and Demos: See Day2.zip

Assignment: Impulse Responses & Modes: Look around, listen, find some systems. Find/Record Modal sound. Use FFTFindModes to analyze it. If you're brave, use FFTResynth (Noise and/or Residue) to recreate it.

DSP for Digital Artists

a 3-Day Short Course, by Perry R. Cook, PhD
CalArts Intersession January 14-18, 2019

Day 3: Physics, Delays, Strings, Winds, Statistical Models

Room Acoustics:

Modes??? = natural resonances of system, but too many of them.
Impulse Response??? Sure, but pretty long, and lots of CPU
Delay Lines!! Speed of sound is about 1125 feet per second

Ideal String: $c^2\, d^2y / dx^2 = d^2y / dt^2$ c = Speed of Sound (on string)
Curvature = Acceleration

Solutions: Harmonic Sum of Modes (Fourier) $y(x,t) = \sum_m a_m \sin(m\pi x/L) \cos(m\pi ct/L)$
L = string length

Traveling Waves (D'Alembert) $y(x,t) = y_L(t+x/c) + y_R(t-x/c)$
Left-going, Right-going

Instruments: Basic Karplus/Strong Plucked String, Full Commuted Model with Pick position

Ideal Tube: SAME Equations and Solutions!! (but different conditions at ends)

Instruments: Clarinet, Brass, Flute

Voice: Modes (Formants), excited by source(s)
OR: Same traveling wave, but tube changes shape continuously

Particles, Statistical Models:

Whistle: 1 particle, Helmholtz (sine) resonator/oscillator
Maraca: Many particles, single shared resonance

Code and Demos:

```
A.  1 Delay, add feedback
B.  2 Delay (stereo), add mod, feedback, change sign
C.  3 Feedback delays, reverb
C2. Another simple Reverb Model
C3. Add One-Pole Low-pass filters
D.  Feedback Delay, Shorten until pitch
E.  KarplusStrong  (Comb Filter)
E2. CombFilter (karplus resonator)
E3. DAlembertString
F.  3 Feedback Delay, make chord (1/400,1/500,1/600)
G.  Commuted Synthesis, Body Knock => string
G2. ChucKMandolin (full commuted synthesis, pluck position, damp, etc.)
H.  Clarinet
J.  Flute
K.  Brass
L.  PhysicalGUI Lots of Physical Wind Models
M.  Whistle
N.  Shakers
```