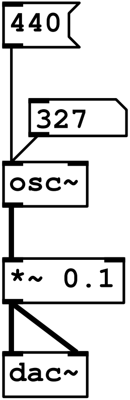
21M.380 Music and Technology
Sound Design (Spring 2016)

Instructor: Florian Hollerweger

Copyright notice



- The Pd patch figures in these notes were generated with the *Pure Data* software. See the license at

  `http://puredata.info/about/pdlicense/`

- Any Pd patch figures referenced as

  Figure: [...] (Farnell 2010, fig. ##.##)

  or

  Figure: [...] (cf., Farnell 2010, ch. ##.##)

  were generated by Florian Hollerweger using Andy Farnell's Pd code supplement to his book *Designing Sound* (Farnell 2010, MIT Press).

How to run the Pd patches from *Designing Sound*

To open any Pd patches that correspond to figures from *Designing Sound* (Farnell 2010) directly by clicking on the ⏵ button in the figure caption:

1. Download *Pd vanilla* and install it on your computer: `http://puredata.info/downloads/pure-data/`
2. Download this PDF to your local hard drive.
3. Download the Pd code to your local drive: `https://mitpress.mit.edu/sites/default/files/titles/content/ds_pd_examples.tar.gz`
4. Unpack the `.tar.gz` tarball and place its `PUREDATA` folder in the same parent directory as this PDF.
5. Open this PDF in a viewer that supports links to local files.
   - Linux: Most viewers
   - Mac OS X: *Adobe Reader* or *Skim* (but not *Preview*)
   - Windows: *Adobe Reader* (and probably others)
6. Use this button to test (Pd should open with a patch): ⏵

How to run other audio examples

Other figures are associated with audio examples available from the OpenCourseWare site. To play these by clicking on the ▶ button in the figure caption:

1. Download the examples to your local drive:
   `http://ocw.mit.edu/ans7870/21m/21m.380/s16/21m380_examples.zip`
2. Unpack the `.zip` archive and place its `examples` folder in the same parent directory as this PDF.
3. Open this PDF in a viewer that supports links to local files.
   - Linux: Most viewers
   - Mac OS X: *Adobe Reader* or *Skim* (but not *Preview*)
   - Windows: *Adobe Reader* (and probably others)
4. Use this button to test (audio player should open with a file): ▶

21M.380 Music and Technology
Sound Design

Lecture 1: Why and how to design sound?

Massachusetts Institute of Technology
Music and Theater Arts

Wednesday, February 3, 2016

Description

In this course, you will learn how to build sounds and sound effects from scratch, using the open-source graphical programming environment *Pure Data* (`https://puredata.info/`). You will learn how to analyze and synthesize everyday sounds and encapsulate them in dynamic sound objects that can be embedded into computer games, animations, movies, virtual environments, sound installations, and theater productions. Our work will be guided by Andy Farnell's book *Designing Sound*.

Intended learning outcomes

1. Reflect upon and analyze everyday sonic experiences and articulate them to others
2. Design and implement computer music applications using essential sound synthesis and programming techniques
3. Identify suitable synthesis techniques to develop a design strategy for a specific sound design problem

Student selection process

- Class tends to be overenrolled (let's see)
- Please tell me about yourself in the attached info sheet!
- Selection results will be announced in (or before) next class meeting

Locations

Image removed for privacy.

Figure: 21M.380 course locations

Locations

Figure: Lewis Music Library, [redacted] (Courtesy of Lewis Music Library.)

Desktop or laptop computer

Figure: Lewis Music Library, [redacted] (Courtesy of Lewis Music Library. cc BY-NC)

Studio headphones (or nearfield monitor loudspeakers)

Manufacturer and model	Price	Back
Beyerdynamic DT770 Pro 80 Ω	$175	closed
Beyerdynamic DT770 Pro 250 Ω	$175	closed
Audio-Technica ATH-M series	$50 – $170	closed
Sennheiser HD 25-1 II	$170	closed
Sennheiser HD 25-SP II	$120	closed
Sennheiser HD280 Pro	$90	closed
Shure SRH440	$100	closed
AKG K240 MKII	$150	semi-open
AKG K240 Studio	$85	semi-open
AKG K99	$80	semi-open
AKG K77	$50	semi-closed
AKG K44	$30	closed

Table: Some headphones suitable for use in this course

Pure Data (Pd)

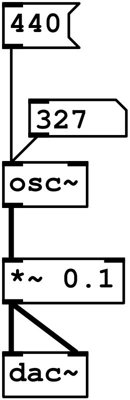


Figure: A typical Pure Data (Pd) patch

- Install *Pd vanilla* (0.46.7) from `http://puredata.info/downloads/pure-data`
- Subscribe yourself to mailing list: `http://lists.puredata.info/listinfo/pd-list`

Pd showcase

Figure: Beat Jazz ()

Pd showcase

Figure: 'Deus Cantando' by Peter Ablinger and Winfried Ritsch ()

Pd showcase

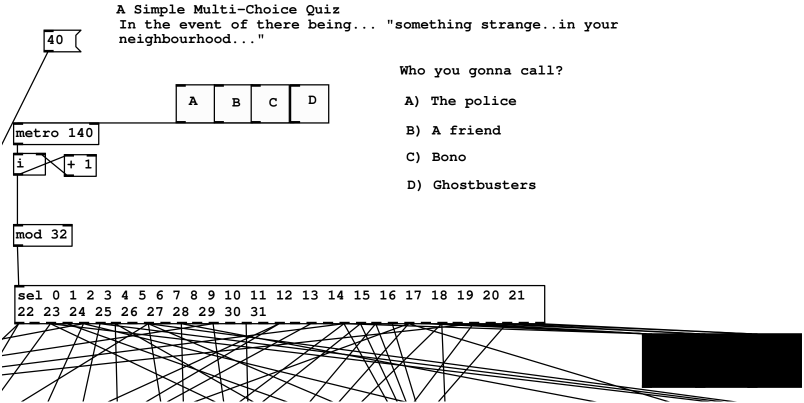


Figure: Who you gonna call? (© Oliver Devlin. License: . This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/)

Pd showcase

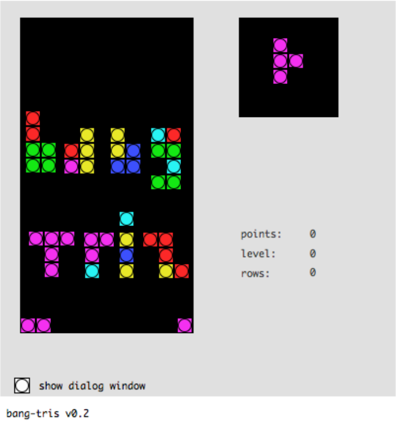


Figure: Bangtris (© Marius Schebella. License: pd~. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/)

Audacity (or other audio editing software)

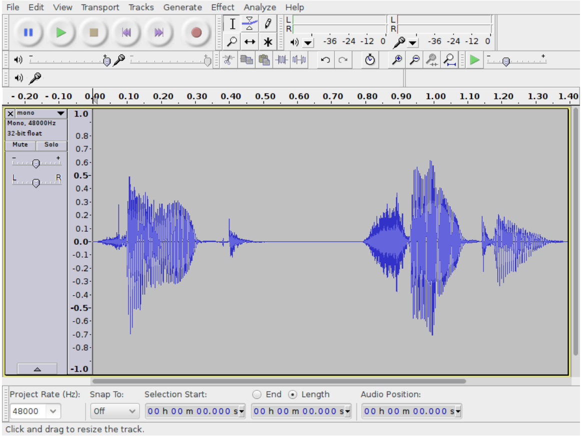


Figure: Audacity

Audacity (or other audio editing software)

Software package	Linux	Mac	Win	Price
Audacity	✔	✔	✔	$0
Hairersoft Amadeus Lite		✔		$25
Hairersoft Amadeus Pro		✔		$60
Sony Sound Forge Audio Studio		✔	✔	$60
snd	✔	(✔)		$0
Steinberg WaveLab Elements		✔	✔	$100
Adobe Audition CC		✔	✔	$20 / mth

Table: Some audio editors suitable for use in this course

Sonic Visualiser

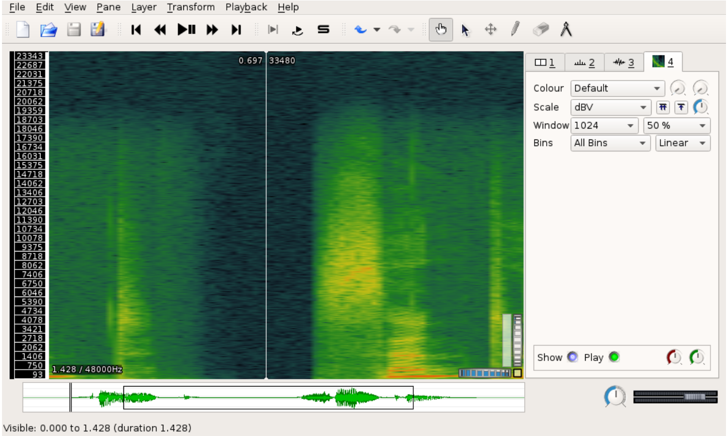


Figure: Sonic Visualiser

Handheld recorders at the Lewis Music Library

Figure: Zoom H4n portable audio recorder (© Zoom North America. All rights reserved. This content is excluded from our Creative Commons license. For more information, see `http://ocw.mit.edu/help/faq-fair-use/`)

Practice drumstick

Figure: Vic Firth 5B ‘Chop-Out’ Practice Stick ()

Required textbook and readings

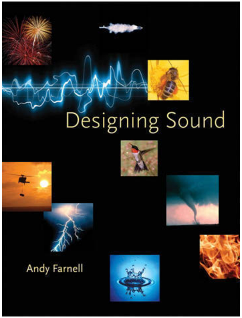


Figure:

- Andy Farnell (2010). *Designing Sound*. Cambridge, MA and London: MIT Press. 688 pp. ISBN: 978-0-262-01441-0. MIT LIBRARY: 001782567. Hardcopy and electronic resource.
- Reading assignments throughout semester (textbook chapters and additional articles)

Assignments, quizzes, and grading

Description	Code	% of final grade			$\sum$
2 Quizzes	QZ1+QZ2		10%	10%	20%
10 reading assignments	RD1–RD10				5%
3 Sound design exercises	EX1–EX3	5%	5%	10%	20%
3 Pd assignments	PD1–PD3	5%	5%	10%	20%
1 Written assignment	WR				5%
1 Recording/editing assignment	ED				5%
Final project in 4 parts	FP1–FP4				25%

Table: Assessment items and final grade contributions

Assignments, quizzes, and grading

Letter grade	Numeric score
A	90%–100%
B	80%–89%
C	70%–79%
D	60%–69%
F	0%–59%

Table: Grading scheme

Attendance

- Any absences have to be communicated to and approved by the instructor ahead of time.
- One unexcused absence without penalty (except for in-class presentations)

Schedule

№	Date	Content
1	Wed, 2/3	Why and how to design sound?
2	Mon, 2/8	The sound design process
3	Wed, 2/10	Everyday sound objects
	Mon, 2/15	**No class (Monday schedule held on Tuesday)**
4	Tue, 2/16	Introduction to Pure Data (Pd)
5	Wed, 2/17	Physics of sound
6	Mon, 2/22	Pd programming concepts
7	Wed, 2/24	Perception of sound
8	Mon, 2/29	Soundwalk
9	Wed, 3/2	Shaping sound with Pd

Schedule (cont.)

№	Date	Content
10	Mon, 3/7	Digital audio theory
11	Wed, 3/9	Sound recording and editing techniques
12	Mon, 3/14	Quiz, review, preview
13	Wed, 3/16	Analysis and requirements specification
	Mon, 3/21	**No class (spring vacation)**
	Wed, 3/23	**No class (spring vacation)**
14	Mon, 3/28	Additive synthesis
15	Wed, 3/30	Research and model making
16	Mon, 4/4	Waveshaping and wavetable synthesis
17	Wed, 4/6	Student presentations
18	Mon, 4/11	Modulation synthesis (AM and FM)
19	Wed, 4/13	Method selection and implementation
	Mon, 4/18	**No class (Patriots Day)**

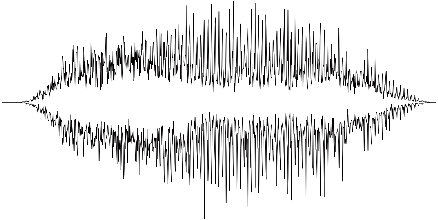
Schedule (cont.)

№	Date	Content
20	Wed, 4/20	Steam train drive-by
21	Mon, 4/25	Granular synthesis
22	Wed, 4/27	Student presentations
23	Mon, 5/2	Quiz and student presentations
24	Wed, 5/4	Thunder
25	Mon, 5/9	Music synthesizers
26	Wed, 5/11	Final project presentations

Why sound design?

- Product design (the Mercedes and the box of nails)
- Computer games
- Movies and animations
- Theater
- Virtual environments

Sample-based sound design



(a) Helicopter sound sample

```
play heli.flac lowpass 300
play heli.flac speed 0.5
play heli.flac stretch 2
play heli.flac reverse
play heli.flac overdrive
```

(b) Some transformations using SoX

Figure: Sound design based on a helicopter sample

Limitations of sample-based sound design

> *[W]e need to understand the limitations of sampled sound and the ambiguity in the word "realistic." Sampled sound is nothing more than a recording. The limitation that immediately presents itself is that sampled sound is fixed in time. No matter what clever tricks of blending, layering, filtering, and truncation we apply, the fact remains that samples are a one-off process. A recording captures the digital signal of a single instance of a sound, but not its behaviour.* (Farnell 2010, sec. 22.2)

Limitations of sample-based sound design

By analogy to traditional game sound technology, an event-based graphical game would only be a series of static photographs, much like the popular Myst game of the 19[9]0s. (Farnell 2010, sec. 22.2; © MIT Press)

Figure: Screenshot from the computer game Myst (© Cyan, Inc.)

Procedural sound design

> *The thesis of [procedural sound design] is that any sound can be generated from first principles, guided by analysis and synthesis. An idea evolving from that is that, in some ways, sounds so constructed are more realistic and useful than recordings because they capture behaviour.* (Farnell 2010, ch. 1)

Advantages of procedural sound design

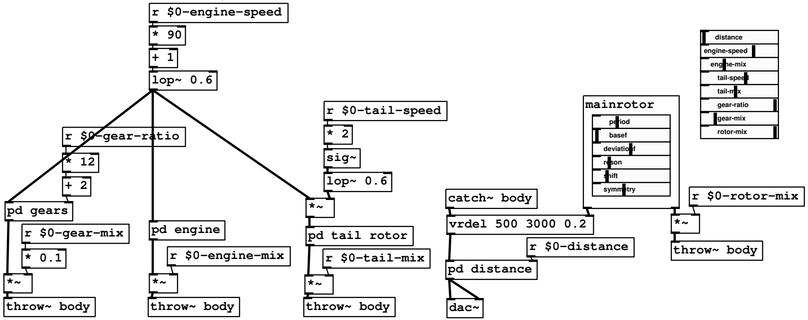


Figure: Procedural helicopter sound object (Farnell 2010, fig. 48.17)

Advantages of procedural sound design

> *[With procedural audio, t]he sound of flying bullets or airplane propellers can adapt to velocity in ways that are impossible with current resampling or pitch-shifting techniques. Synthesised crowds can burst into applause or shouting; complex weather systems where the wind speed affects the sound of rainfall; rain that sounds different when falling on roofs or into water; realistic footsteps that automatically adapt to player speed, ground texture, and incline—the dynamic possibilities are practically endless.* (Farnell 2010, sec. 22.4)

Procedural audio showcase

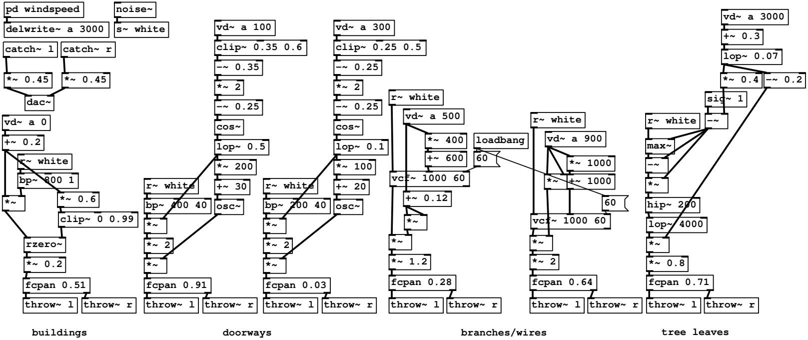


Figure: Wind synthesized in Pd (cf., Farnell 2010, ch. 41)

Procedural audio showcase

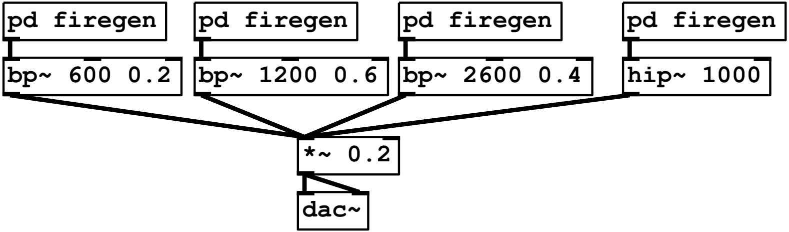


Figure: Fire synthesized in Pd (Farnell 2010, fig. 34.13)

Procedural audio showcase

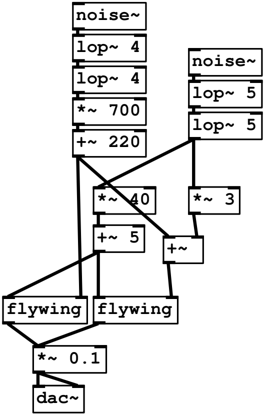


Figure: Buzzing housefly synthesized in Pd (Farnell 2010, fig. 50.14)

Procedural audio showcase

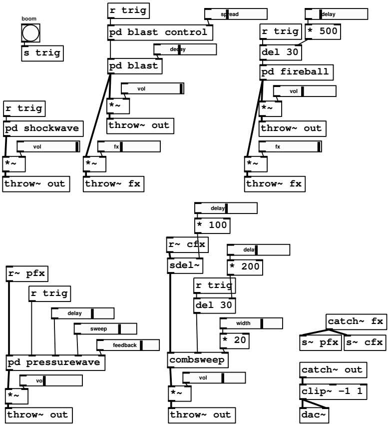


Figure: Explosion synthesized in Pd (Farnell 2010, fig. 54.9)

21M.380 Music and Technology
Sound Design

Lecture 2: The sound design process

Massachusetts Institute of Technology
Music and Theater Arts

Monday, February 8, 2016

Film Sound Cliches (*Film Sound Cliches* 2015)

Figure: The Wilhelm Scream (© Warner Bros. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/)

Film Sound Cliches (*Film Sound Cliches* 2015)

- Castle thunder: `http://www.hollywoodlostandfound.net/sound/castlethunder.html`
- The Universal telephone ring: `http://hollywoodlostandfound.net/sound/uniphone.html`

Film Sound Cliches (*Film Sound Cliches* 2015)

- “Snakes are always rattling”
- “All bicycles have bells (that sounds) [sic]”
- “[I]n U.S. films playing in big cities there’s always a police horn in the background—in films from other countries... never!”
- “Helicopters always fly from surround to front-speakers.”
- “The DJ always turns the music down when actors talk in disco and club-scenes”
- “Explosions in space make noise”
- “Dreams are always drenched in a lot of reverb.”

"The computer as a game technology" (Crawford 1997b)

> *The role of information storage in a computer is often misunderstood. A computer is not primarily an information storage device; it is instead an information processing device. Information storage is a necessary precondition for information processing, but it is not an end in itself.* (Crawford 1997b)

"The computer as a game technology" (Crawford 1997b)

> *Thus, a game that sports huge quantities of static data is not making best use of the strengths of the machine. A game that emphasizes information processing and treats information dynamically is more in tune with the machine.* (Crawford 1997b)

History

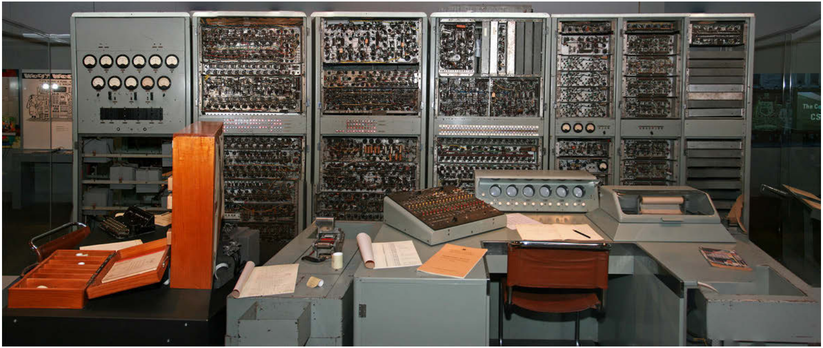


Figure: In 1951, Geoff Hill made CSIRAC the first computer to play music, which was reconstructed by Paul Doornbusch (2005) (© John O'Neill. CC BY-SA. This content is excluded from our Creative Commons license. For more information, see `http://ocw.mit.edu/help/faq-fair-use/`)

History

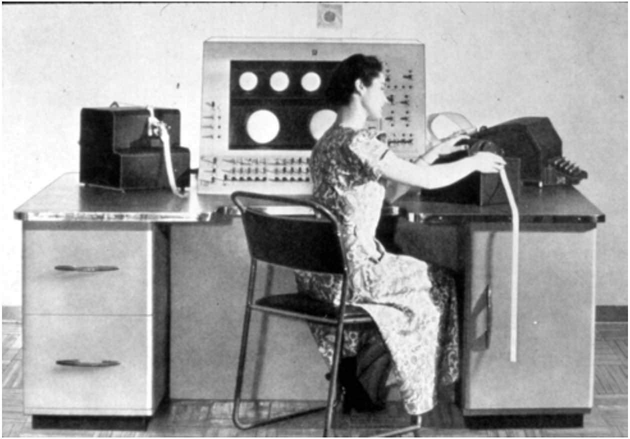


Figure: The earliest surviving recording of a computer playing music was performed by a Ferranti Mark I in 1951 (cf., Fildes 2008)

History

Figure: Memory drum of an ILLIAC I, the machine which Hiller and Isaacson used to compose the *Illiac Suite* in 1956 ()

History

Image removed due to copyright restrictions. Photo of Max Mathews. Source: `http://www.wired.com/images_blogs/gadgetlab/2011/01/max_mathews.jpg`

Figure: Max Mathews ($\star$ 1926, $\dagger$ 2011)

Bell Labs

Management tolerated music as part of research. (Max Mathews)

- AT&T's research lab in New Jersey, US
- 7 nobel prizes
- Selected inventions: transistor, telefax, laser, communication satellite, mobile phone (concept from 1940!), C, Unix, ...
- On average 25 products per US household
- Mathews' work: phone sound transmission quality

Bell Labs

Figure: Max Mathews used an IBM 704 to realize Newman Guttman's composition *In the Silver Scale* in 1957 (Public domain image. Source: https://www.flickr.com/photos/nasacommons/9467782802/)

Bell Labs

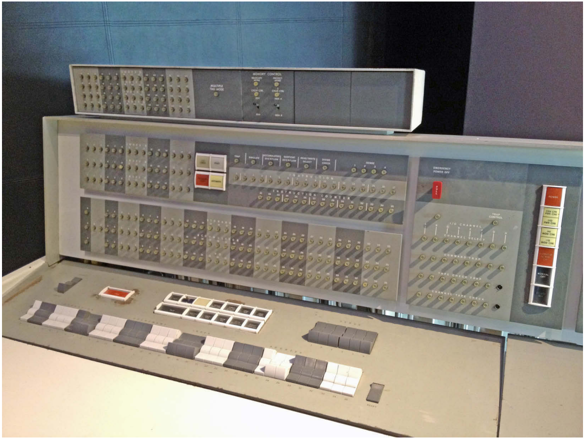


Figure: Operator's console of an IBM 7094, the machine which M. Mathews, Kelly, and Lockbaum used to peform *Daisy Bell* in 1961 (© Wikipedia user: Arnold Reinhold. CC BY-SA. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/)

Bell Labs

Max Mathews (1963). "The Digital Computer as a Musical Instrument." In: *Science* 142.3592, pp. 553–7. JSTOR: 1712380. URL: http://www.jstor.org/stable/1712380

Key insights

- No theoretical limitations ('any sound')
- Limited only by cost and psychoacoustic knowledge
- Serious composers should get engaged

Music N family

Language	Description
Music I	Max Mathews, Bell Labs (1957); 1 voice, triangle waves
Music III	Introduces *unit generator* concept
Music IV-B	Written in Fortran (non-machine specific)
Music V	A classic from 1968 (cf., M. V. Mathews 1969)

Table: Some computer music languages from the Music N family

Music N family

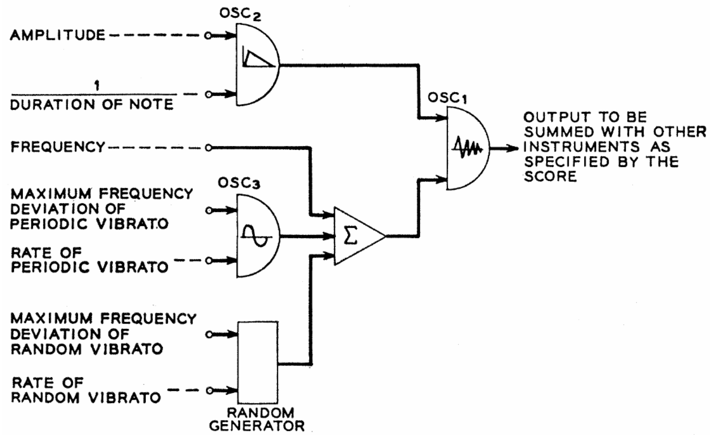


Figure: Unit generator concept (M. Mathews 1963, p. 555.)

Music N family

Music V gave us a kind of musical counterpart of Postscript for 2D graphics (the standard marking language used first for laser printers and more recently for computer displays). Apparently, Music V was born at least three decades too soon to be accepted as the PostScript of the music world. Instead, we got MIDI [...].
(J. O. Smith 1991)

Second-generation languages

Language	Description
CSound	Direct Music-N descendant, still in widespread use
CMix	By Paul Lansky, Brad Garton, and others
CMusic	By Richard Moore

Table: Second-generation computer music languages

Third-generation languages

Language	Description
SuperCollider	Real-time capable, interpreted language
ChucK	Popular for live-coding
Max/MSP	Graphic programming environment
Pure Data (Pd)	Max/MSP's open-source twin

Table: Contemporary computer music languages

Third-generation languages

```
adc => Gain g => dac;                          // feedforward
g => Gain feedback => DelayL delay => g; // feedback

.75::second => delay.max => delay.delay; // set delay
.5 => feedback.gain;                           // set feedback
.75 => delay.gain;                             // set effects mix

while( true ) 1::second => now;                // infinite loop
```

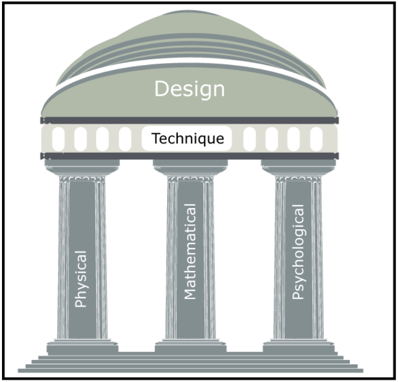
Listing 1: An example program from the ChucK documentation. (© ChucK authors. License: GNU GPL. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/)

Third-generation languages

```
(
play{x=165;b=SinOsc;p=Trig.ar(Saw.ar(x),1);
     y=b.ar(p*x);z=b.ar(p);
     (GVerb.ar(GrainIn.ar(2,y,y/2,z,p*z,-1),9))/9}
//basso gettato #SuperCollider
)
```

Listing 2: A SuperCollider piece that fits into a tweet (© José Padovani. BY-NC-SA.

Three pillars of sound design



(a) Three pillars of sound design (Image by MIT OpenCourseWare, after Farnell 2010, fig. 2.1.)

(b) MIT's Great Dome and Killian Court (Courtesy of Joey Rozier on Flickr.)

Figure: Relationship between theory, technique, and design

Top-down design vs. bottom-up implementation

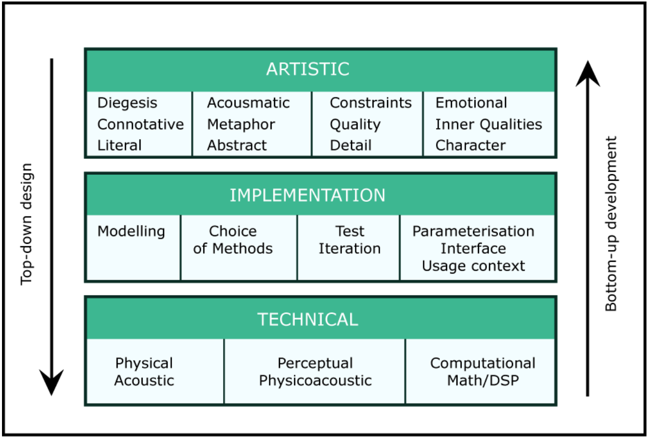


Figure: Overview of technique (Image by MIT OpenCourseWare, after Farnell 2010, fig. 15.1.)

Design stages

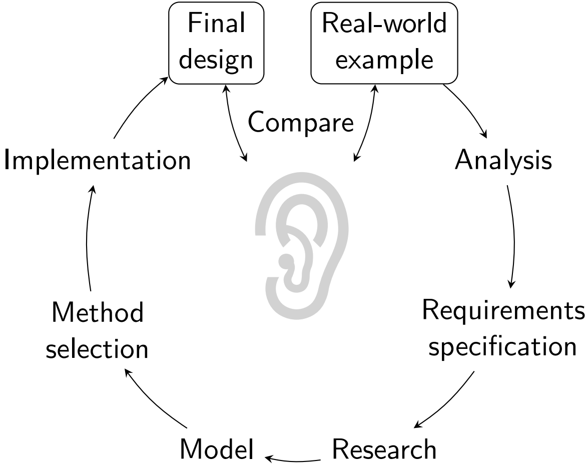


Figure: Stages of the sound design process (after Farnell 2010, figs. 16.7, 16.1)

21M.380 Music and Technology
Sound Design

Lecture 3: Everyday sound objects

Massachusetts Institute of Technology
Music and Theater Arts

Wednesday, February 10, 2016

"What we use for ..." (Ament 2009b)

Image removed due to copyright restrictions.

Figure: "That's the woman who listens to the liver!" (Ament 2009a, p. 112)

EX1 presentations

- Present the sound object you brought to class today (3 min. each)
- Why did you choose this object?
- Show us the sounds it can make and describe them.

21M.380 Music and Technology
Sound Design

Lecture 4: Introduction to Pure Data (Pd)

Massachusetts Institute of Technology
Music and Theater Arts

Tuesday, February 16, 2016

Starting Pure Data

In his book How to Be Creative Hugh Mac[L]eod [2004] gives away one of the best secrets about being a successful producer, that there is no correlation between creativity and ownership of equipment: as an artist gets more proficient the number of tools goes down. (Farnell 2010, p. 147)

Starting Pure Data

> *You can often tell an extremely powerful tool by its Spartan appearance. It does not need to advertise itself. There are no flashing graphics or whizzbangs, just a command prompt or a blank canvas. What this is saying is "I am ready to do your bidding, Master." Many get stuck here, because they never thought about what they want to do, expecting the tools to lead them rather than the other way about.* (Farnell 2010, p. 147)

The main Pd window

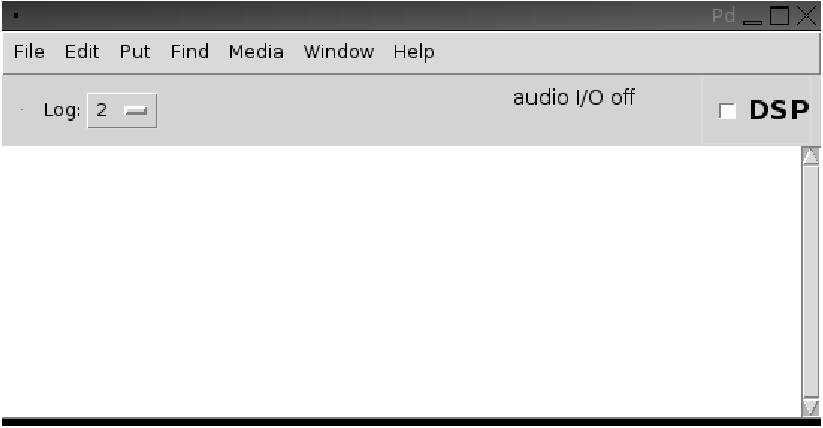


Figure: Pure Data console (after Farnell 2010, fig. 9.1)

Testing audio

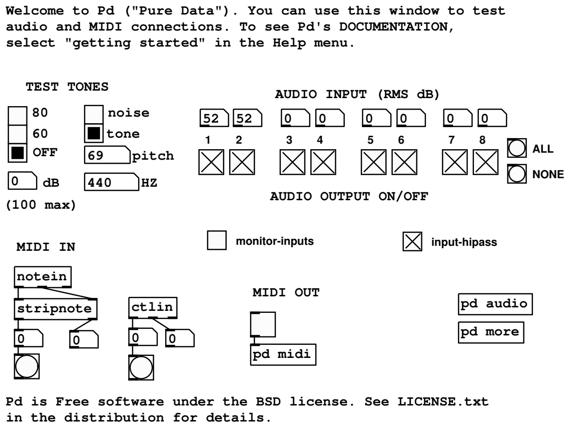


Figure: Test signal: Media 〉 Test Audio and MIDI... (after Farnell 2010, fig. 9.2)

Testing audio

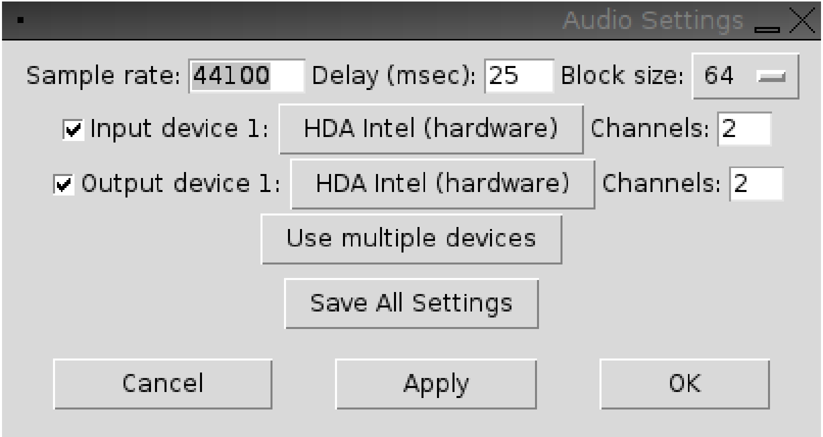


Figure: Audio settings pane: Preferences › Audio Settings... (Mac), Media › Audio Settings... (Win, Linux) (after Farnell 2010, fig. 9.3)

Creating a new Pd patch

Menu entry	Linux & Windows	Mac OS X
File > New	Ctrl + n	⌘ + n

Table: Creating a new patch in Pd

Adding basic building blocks

Menu entry	Linux & Windows	Mac OS X
Put › Object	Ctrl + 1	⌘ + 1
Put › Message	Ctrl + 2	⌘ + 2
Put › Number	Ctrl + 3	⌘ + 3
Put › Comment	Ctrl + 5	⌘ + 5

Table: Adding basic elements to a new Pd patch

Edit mode

Menu entry	Linux & Win	Mac	Meaning
Edit > Edit Mode	Ctrl + e	⌘ + e	Toggle edit/run
—	Hold Ctrl	Hold ⌘	Temporary run mode

Table: Switching between edit and run modes

Edit mode

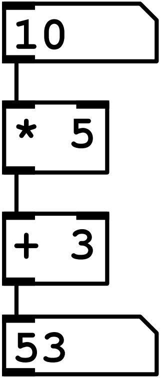


Figure: A basic Pd patch with objects and number boxes (Farnell 2010, fig. 9.9)

Edit mode

Linux & Windows	Mac	Meaning
Ctrl + a	⌘ + a	Select all objects on canvas
Ctrl + d	⌘ + d	Duplicate the selection
Ctrl + c	⌘ + c	Copy the selection
Ctrl + v	⌘ + v	Paste the selection
Ctrl + x	⌘ + x	Cut the selection
Hold ⇧	Hold ⇧	Select multiple objects

Table: Pd edit operations

Patch files (.pd)

Menu entry	Linux & Windows	Mac	Meaning
File > Save	Ctrl + s	⌘ + s	Save patch to .pd file
File > Open	Ctrl + o	⌘ + o	Open .pd file

Table: Saving and opening Pd patches to and from .pd files

Patch files (.pd)

```
#N canvas 87 655 91 178 10;
#X obj 14 79 osc~;
#X msg 14 18 440;
#X obj 14 116 *~ 0.1;
#X obj 14 152 dac~;
#X floatatom 24 55 5 0 0 0 - - -, f 5;
#X connect 0 0 2 0;
#X connect 1 0 0 0;
#X connect 2 0 3 0;
#X connect 2 0 3 1;
#X connect 4 0 0 0;
```

Listing 3: Looking at a .pd file in a text editor

GUI elements

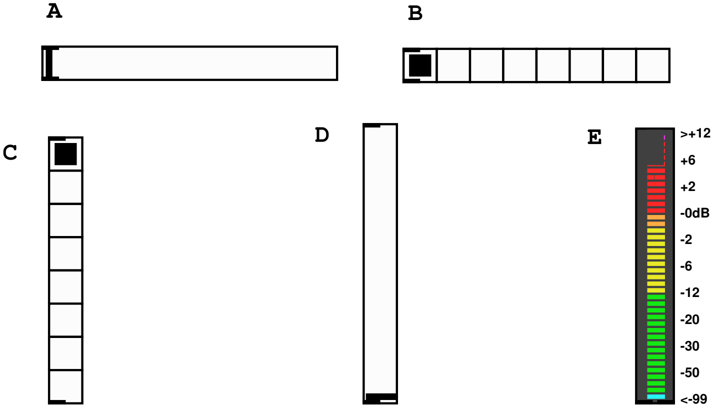


Figure: GUI Objects. A: Horizontal slider. B: Horizontal radio box. C: Vertical radio box. D: Vertical slider. E: VU meter (Farnell 2010, fig. 9.10)

Arrays

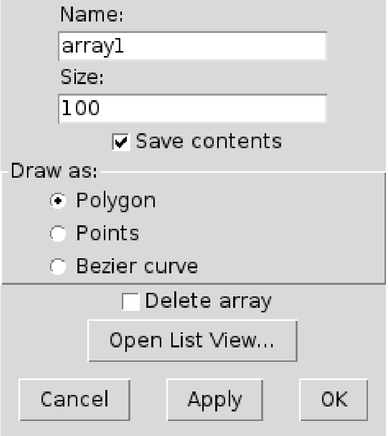


Figure: Create array with Put › Array (after Farnell 2010, fig. 9.12)

Arrays

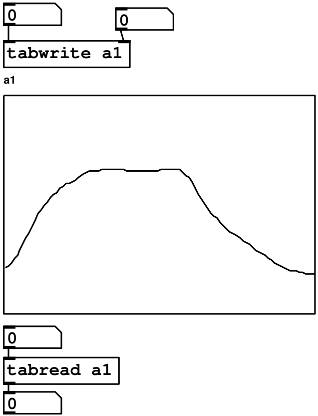


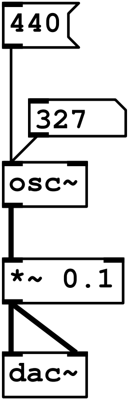
Figure: Accessing an array (Farnell 2010, fig. 9.13)

Pd's internal help system

Linux & Windows	Mac
Help ❯ Browser...	Help ❯ Browser
Help ❯ List of objects...	Help ❯ List of objects...
right-click Help on object	ctrl+click Help on object

Table: Ways of getting help in Pd

Plain text patch notation




```
[440(       ...   message
  |         ...   connection
  | [327\   ...   number box
  | /       ...   connections
[osc~]      ...   object
  |
[*~ 0.1]    ...   object with argument
  | \
[dac~]
```

Figure: Pd patch and equivalent notation in ASCII text

Evaluation order (top-down, right-left)

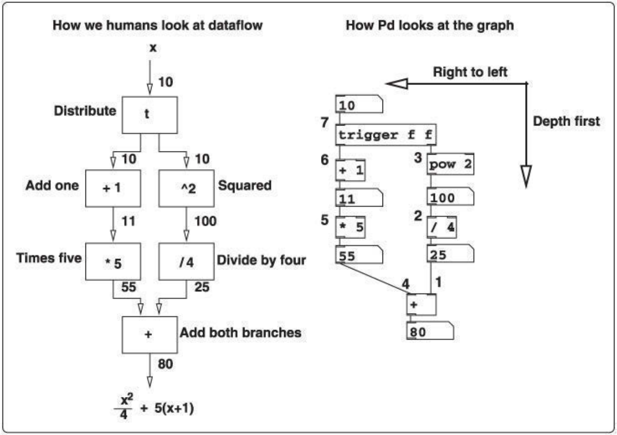


Figure: Dataflow computation (Farnell 2010, fig. 9.4. Courtesy of MIT Press. Used with permission. `https://mitpress.mit.edu/books/designing-sound`)

Pd architecture

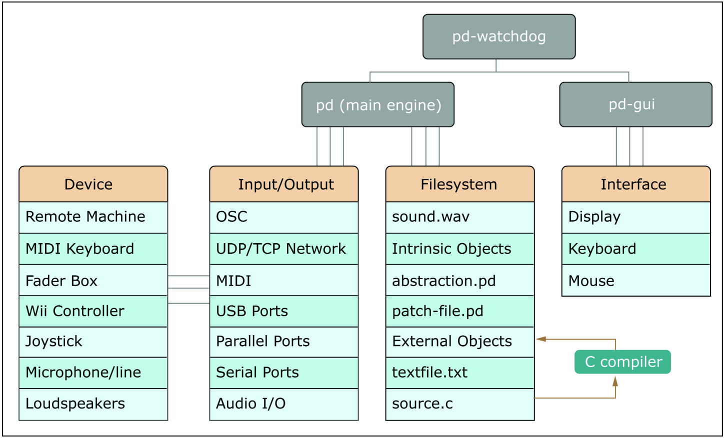


Figure: Pure Data software architecture (Image by MIT OpenCourseWare, after Farnell 2010, fig. 9.5.)

Hot vs. cold inlets

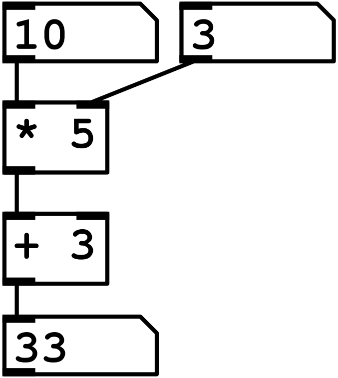


Figure: Left inlet is 'hot', right inlet is 'cold' (Farnell 2010, fig. 10.1)

Ambiguous evaluation order

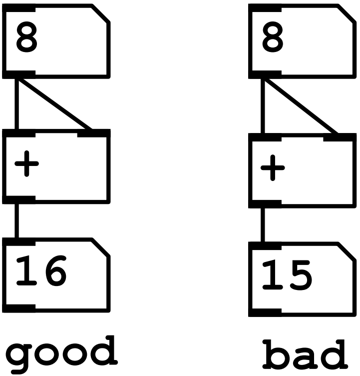


Figure: Bad ordering (Farnell 2010, fig. 10.2)

Forcing evaluation order with [trigger]

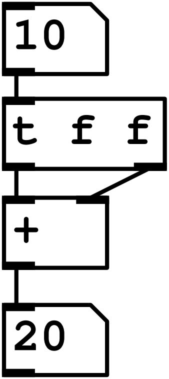


Figure: [trigger] ensures right-to-left evaluation (Farnell 2010, fig. 10.3)

Warming an inlet

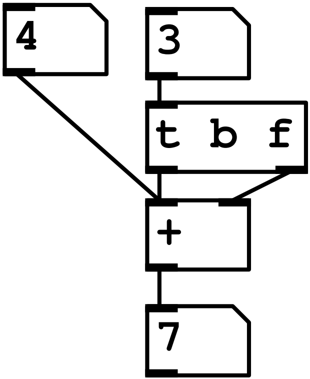


Figure: `[trigger bang float]` warms an inlet (Farnell 2010, fig. 10.4)

Counter

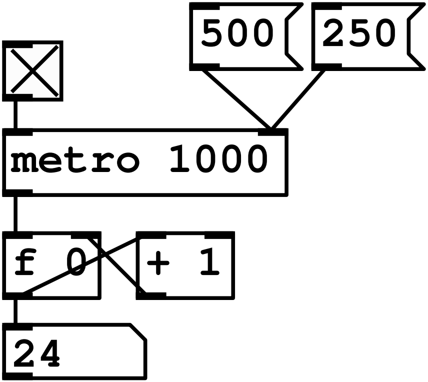


Figure: Counter (Farnell 2010, fig. 10.7)

Controlling message flow

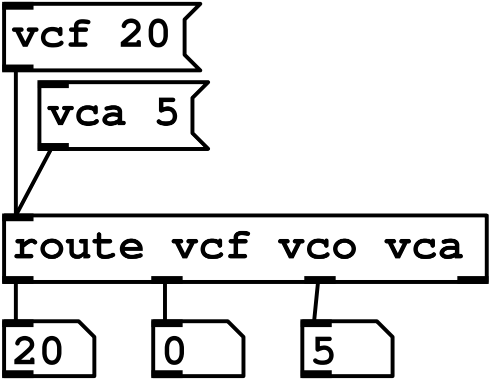


Figure: Routing values (Farnell 2010, fig. 10.10)

Controlling message flow

Pd object	Abbr.	Functionality
[select]	[sel]	Match first element of list
[route]		Match first element of list
[moses]		Moses splits streams
[send]	[s]	Send messages without wires
[receive]	[r]	Receive message from matching [send]

Table: Pd objects that control message flow

Packing and unpacking lists

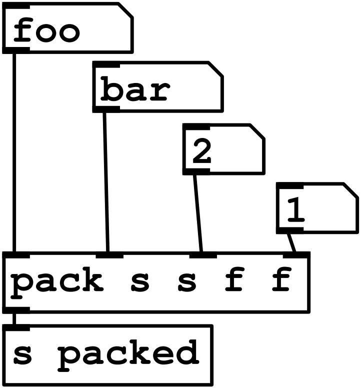


Figure: List packing (Farnell 2010, fig. 10.16)

Packing and unpacking lists

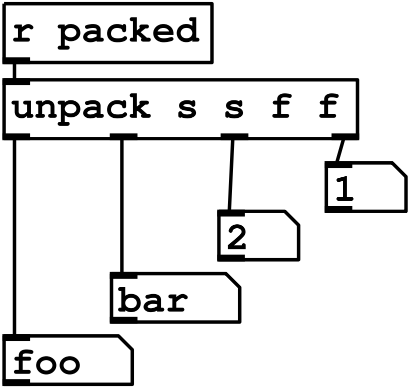


Figure: List unpacking (Farnell 2010, fig. 10.17)

Substitutions

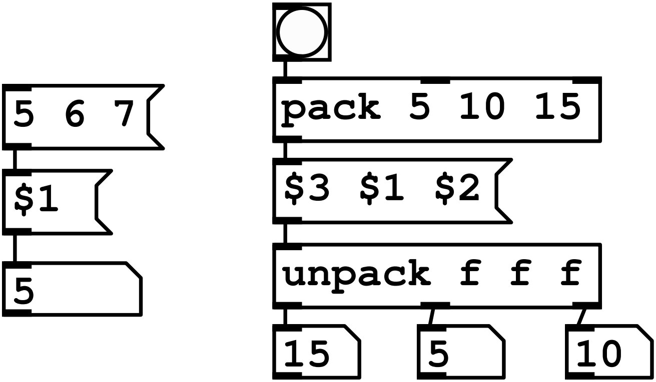


Figure: Dollar substitution (Farnell 2010, fig. 10.18)

Distributing lists across inlets

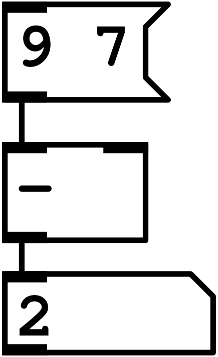


Figure: Distribution (Farnell 2010, fig. 10.20)

Arithmetic

Object	Function
[+]	Add two floating point numbers
[-]	Subtract right inlet from left inlet
[/]	Divide left inlet by right inlet
[*]	Multiply two floating point numbers
[div]	Integer divide
[mod]	Modulo operation

Table: Table of message arithmetic operators (Farnell 2010, fig. 10.24)

Trigonometry

Object	Function	Domain	Range
[cos]	Cosine in radians	$-\frac{\pi}{2}$ to $+\frac{\pi}{2}$	-1.0 to +1.0
[sin]	Sine in radians	$-\frac{\pi}{2}$ to $+\frac{\pi}{2}$	-1.0 to +1.0
[tan]	Tangent in radians		0.0 to ∞ at $+\frac{\pi}{2}$
[atan]	Arctangent in radians	$\pm\infty$	$\pm\frac{\pi}{2}$
[atan2]	Arctangent of x, y pair		$\pm\pi$
[exp]	Exponential function e^x		0 to ∞
[log]	Natural log (base e)	0.0 to ∞	$\pm\infty$ ($-\infty$ is -1000.0)
[abs]	Absolute value	$\pm\infty$	0.0 to ∞
[sqrt]	Square root	0.0 to ∞	
[pow]	Power function x^y	$x > 0$	

Table: Table of message trigonometric and higher math operators (Farnell 2010, fig. 10.25)

Comparing numbers

Object	Outlet = 1 if left inlet is ...
[>]	greater than
[<]	less than
[>=]	greater than or equal to
[<=]	less than or equal to
[==]	equal to
[!=]	not equal to
	... right inlet

Table: List of comparative operators (after Farnell 2010, fig. 10.27)

Scaling numeric ranges

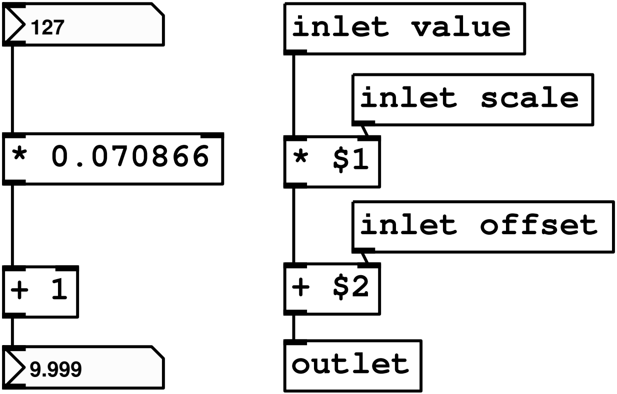


Figure: Scaling (Farnell 2010, fig. 10.31)

21M.380 Music and Technology
Sound Design

Lecture 5: Physics of sound

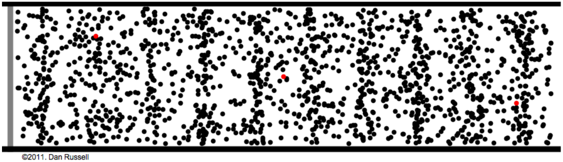
Massachusetts Institute of Technology
Music and Theater Arts

Wednesday, February 17, 2016

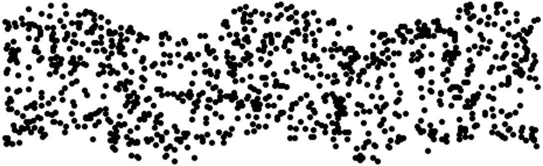
Physics of sound

Figure: Fallen tree in a forest (Courtesy of ChenYen.Lai on Flickr.)

Longitudinal vs. transverse waves



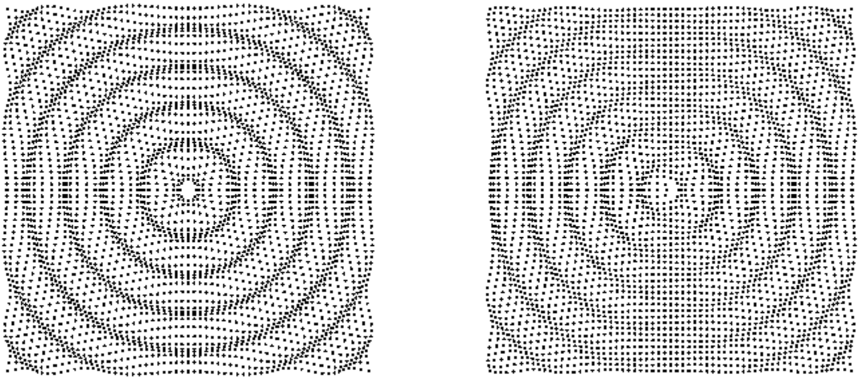
(a) Longitudinal plane wave ▶



(b) Transverse wave ▶

Figure: Wave snapshots (© Daniel A. Russell. Grad. Prog. Acoustics, Penn State. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/)

Radiation patterns



(a) Monopole

(b) Dipole

Figure: Monopole and dipole ()

Spherical vs. plane waves

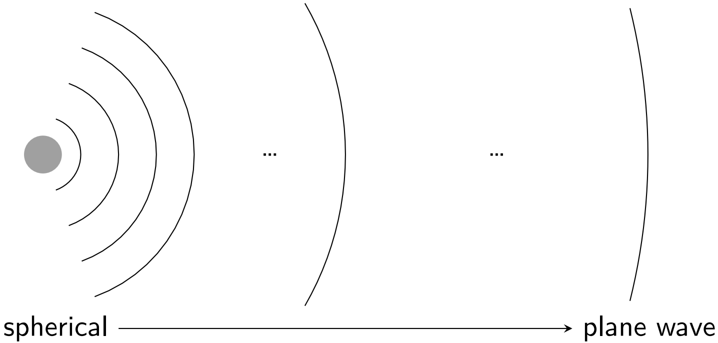


Figure: Spherical vs. plane wavefronts

Wave properties

Property	Symbol	Unit
Amplitude	A	Pa, mV, ...
Period	T	s
Frequency	f	Hz
Wavelength	λ	m
Speed of sound	c	$\mathrm{m\,s^{-1}}$
Phase	φ	° or rad

Table: Properties of sound waves

Amplitude

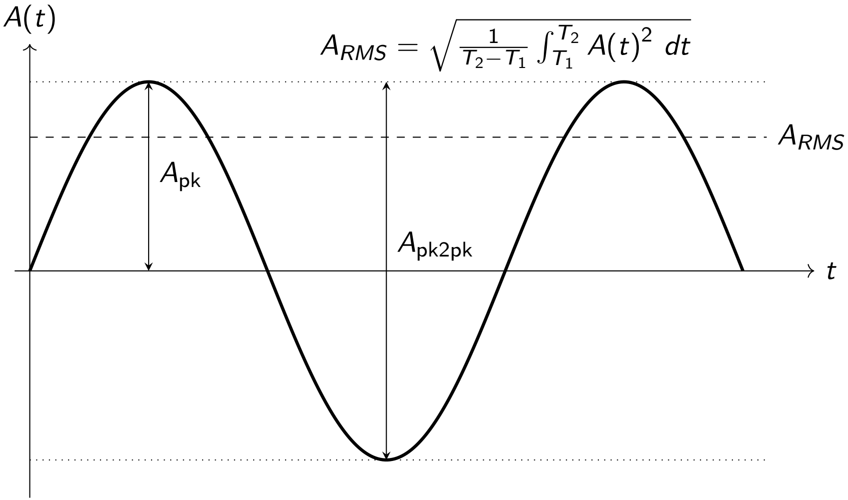


Figure: Peak, peak-to-peak, and RMS amplitudes of a sine wave

Amplitude

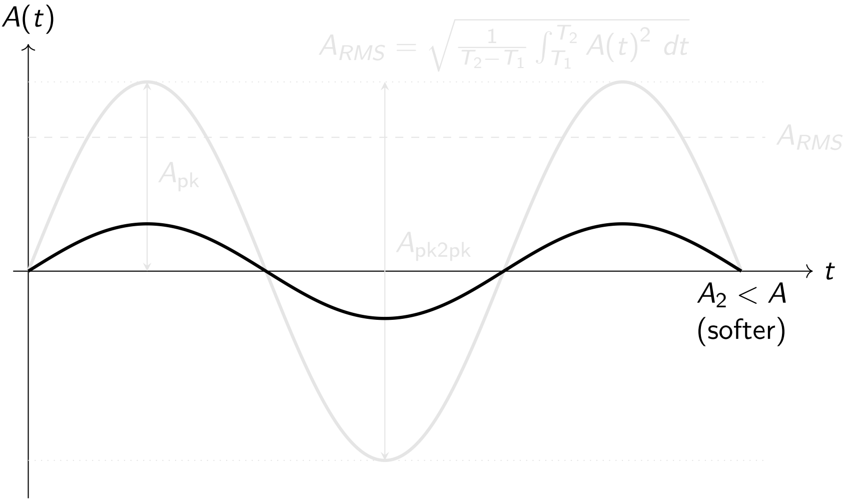


Figure: Amplitude vs. perceived loudness

Amplitude

Definition (Sound pressure level SPL)

$$L_p = 20 \cdot \log_{10}\left(\frac{p}{p_0}\right)$$

- L_p ... sound pressure level ($\mathrm{dB_{SPL}}$)
- p ... measured RMS sound pressure (Pa)
- p_0 ... reference sound pressure (Pa)

Common reference: $p_0 = 20\,\mu\mathrm{Pa} \equiv 0\,\mathrm{dB_{SPL}}$ (threshold of hearing)

Example

Sound pressure level measured by a reference microphone

Frequency and period

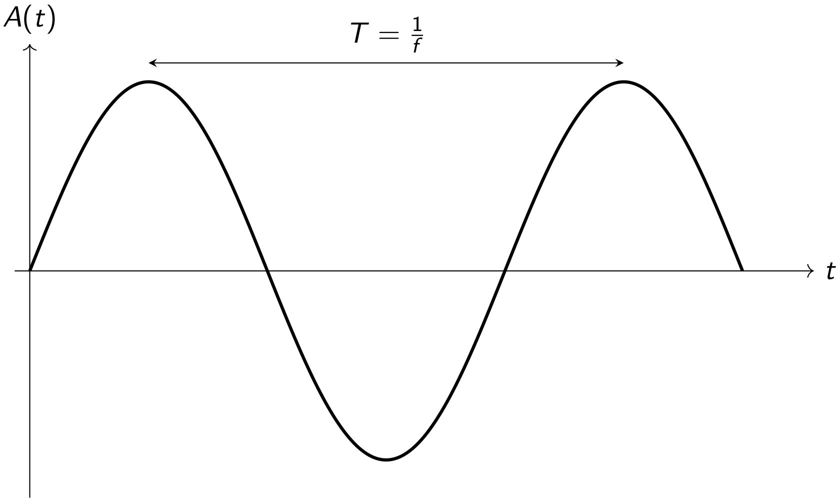


Figure: Period T and frequency f of a sine wave

Frequency and period

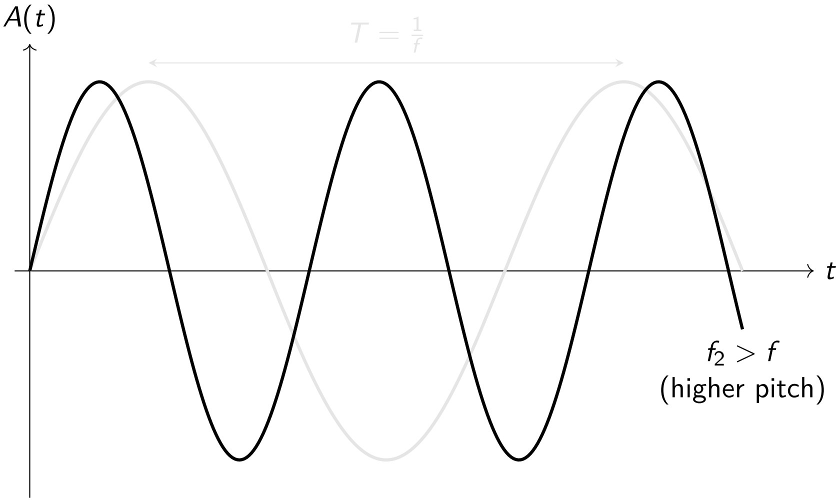


Figure: Frequency vs. perceived pitch

Frequency and period

f_1/f_2	Interval
1:1	Perfect unison
2:1	Perfect octave
3:2	Perfect fifth
4:3	Perfect fourth

Table: Frequency ratios vs. musical intervals

Wavelength

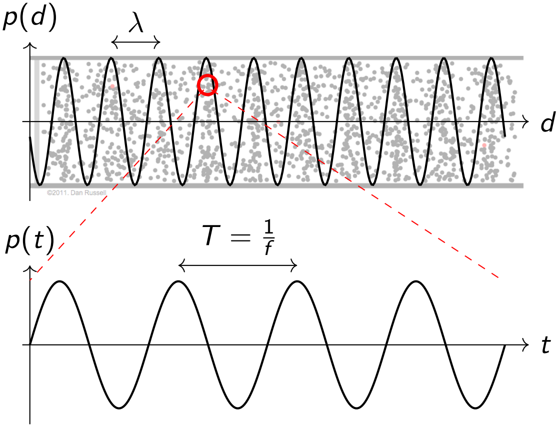


Figure: Sound as a spatial (top) and temporal (bottom) phenomenon (background image by Daniel A. Russell, Grad. Prog. Acoustics, Penn State)

Speed of sound

Wave math (Farnell 2010, eq. 3.9)

$$c = \lambda \cdot f \tag{1}$$

Depends on temperature in air

$$c_{air} \approx 331.3 + 0.606 \cdot \vartheta$$

Number to remember (👍·π)

$$c_{air,15\,\mathrm{C}} \approx 340\,\mathrm{m\,s^{-1}}$$

Medium	$c/\mathrm{m\,s^{-1}}$
Air (20 °C; 0 % hum.)	343.2
Water (fresh; 25 °C)	1497
Steel	4597

Table: c increases with ρ

- c … speed of sound ($\mathrm{m\,s^{-1}}$)
- λ … wavelength (m)
- f … frequency ($\mathrm{Hz} = \mathrm{s^{-1}}$)
- ϑ … temperature (°C)
- ρ … density ($\mathrm{kg\,m^{-3}}$)

Speed of sound

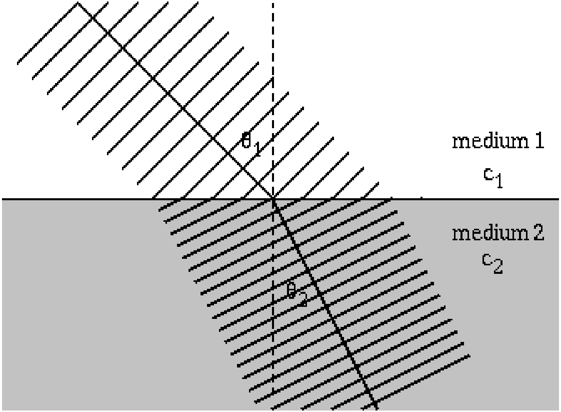


Figure: Change of c and λ across media of differen density ()

Phase

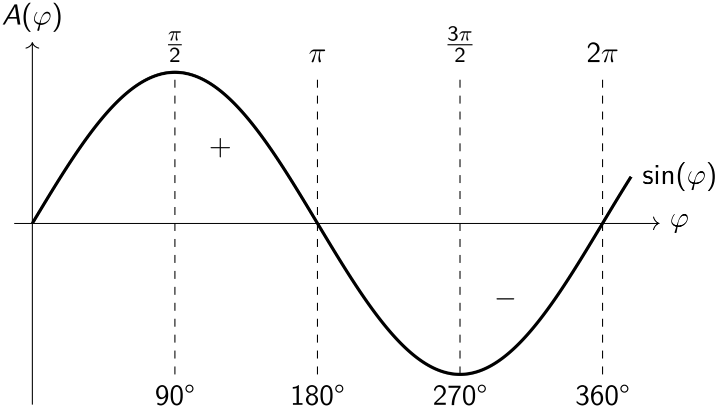


Figure: Phase cycle of a sine wave

Interference and phase cancellation

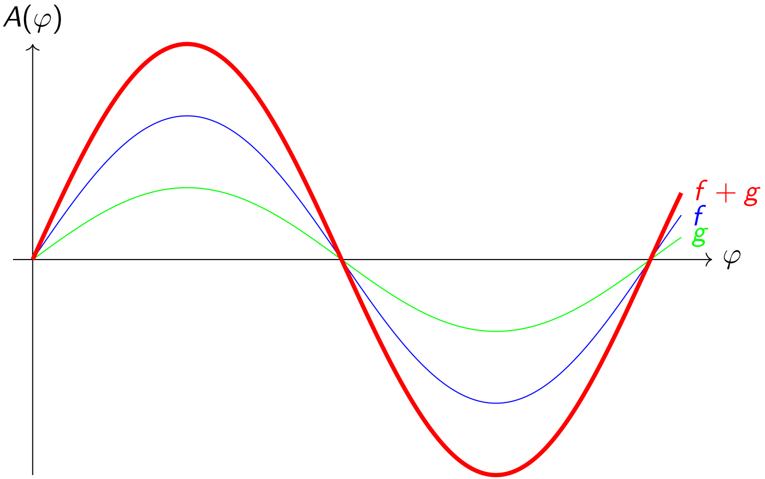


Figure: Constructive interference between two in-phase waves f (blue) and g (green), resulting in a higher-amplitude signal (red, thick)

Interference and phase cancellation

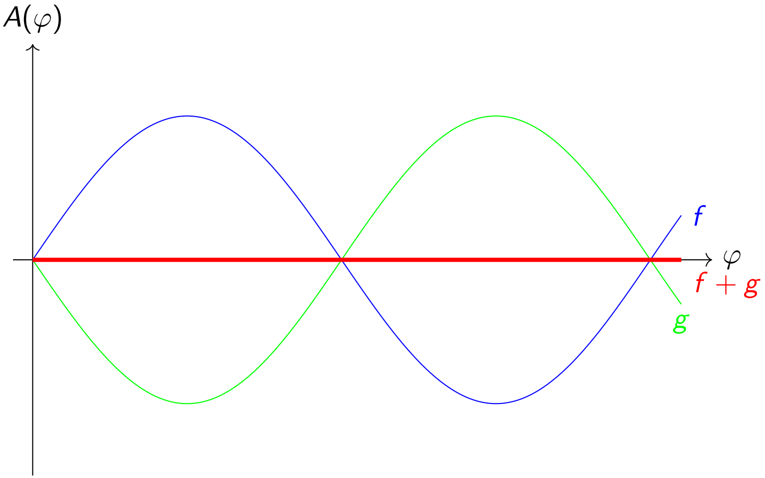


Figure: Destructive interference (phase cancellation) between two anti-phase waves f (blue) and g (green), resulting in a zero signal (red, thick), i.e., silence

Interference and phase cancellation

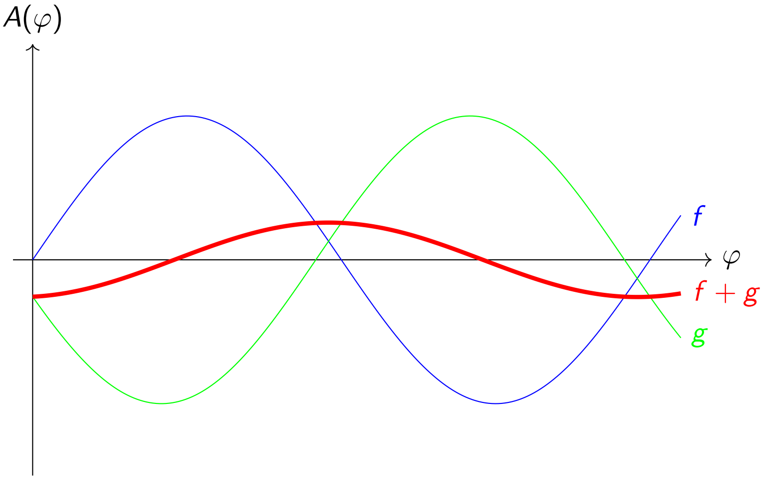


Figure: Mixed interference (mostly destructive) between two out-of-phase waves f (blue) and g (green), resulting in a lower-amplitude signal (red, thick)

Interference and phase cancellation

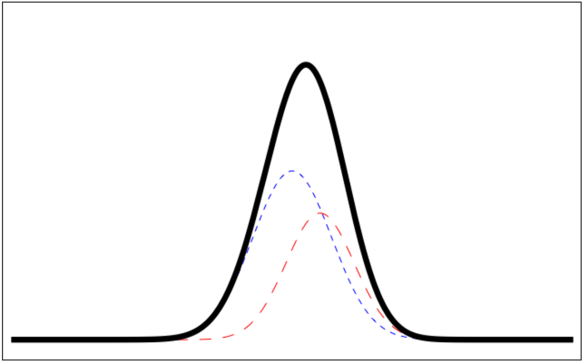


Figure: Superposition of two opposite direction wave pulses ()

Interference and phase cancellation

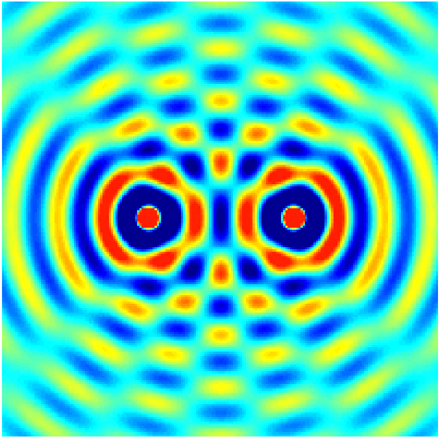


Figure: Interference between two spherical waves (Ⓟ Public domain image. Source: https://en.wikipedia.org/wiki/File:Two_sources_interference.gif)

Visualization as a spectrum

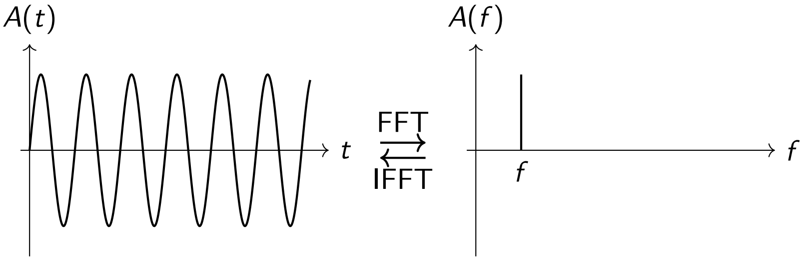


Figure: A sine wave's spectrum consists of a single frequency

Visualization as a spectrum

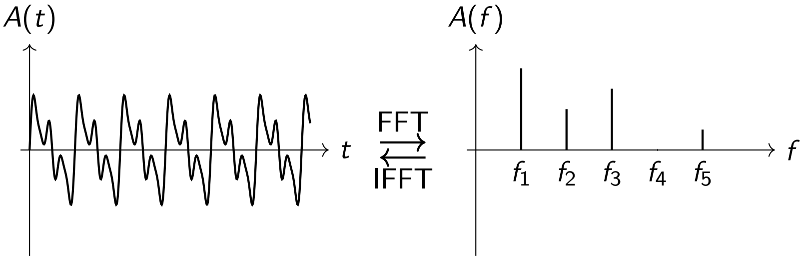


Figure: A periodic wave has a harmonic spectrum

Visualization as a spectrum

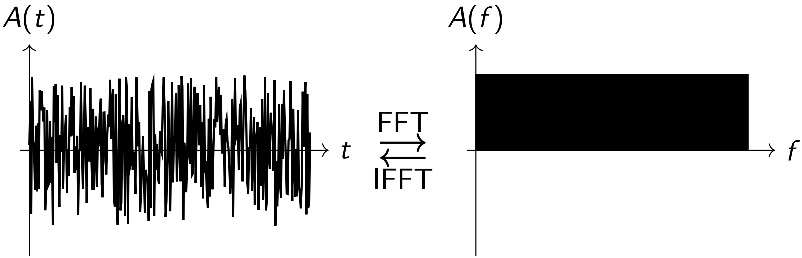


Figure: An aperiodic wave has an inharmonic spectrum

Harmonic sounds

Definition (Harmonic spectrum)

The frequency components f_N of a harmonic spectrum are integer multiples of its fundamental frequency f_1.

$$f_N = N \cdot f_1 \tag{2}$$

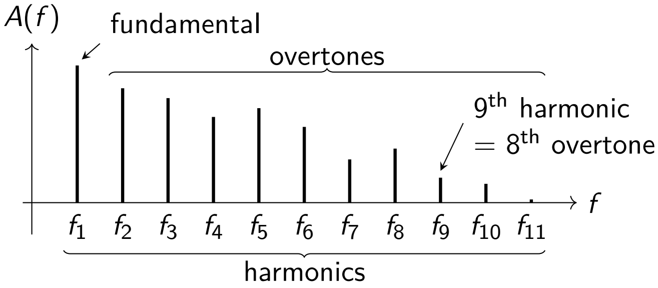


Figure: Harmonic spectrum

Inharmonic sounds

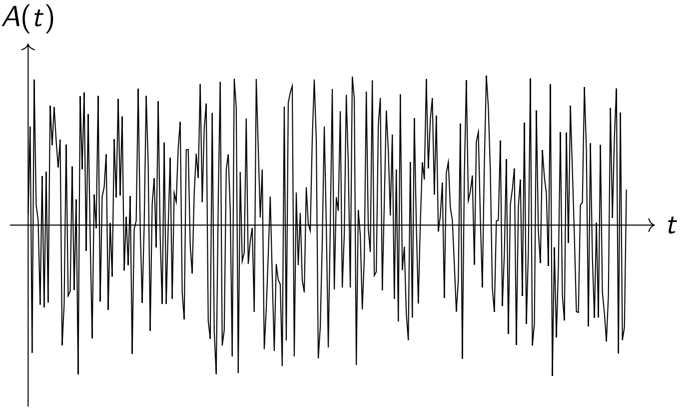
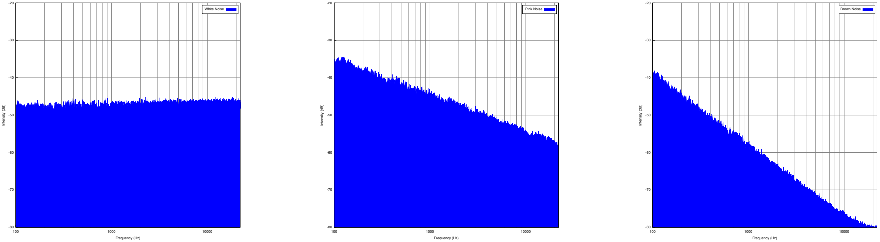


Figure: White noise in the time domain

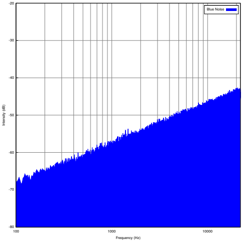
Inharmonic sounds



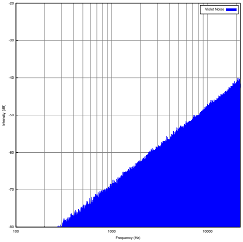
(a) White noise (b) Pink noise (c) Brown(ian) noise

Figure: Spectra of different noise colors (© Wikipedia user: Warrakkk. CC BY-SA. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/)

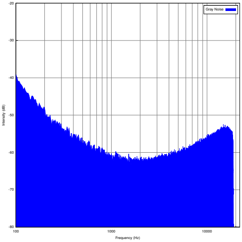
Inharmonic sounds



(a) Blue noise



(b) Violet noise



(c) Grey noise

Figure: Spectra of different noise colors (© Wikipedia user: Warrakkk. CC BY-SA. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/)

Inharmonic sounds

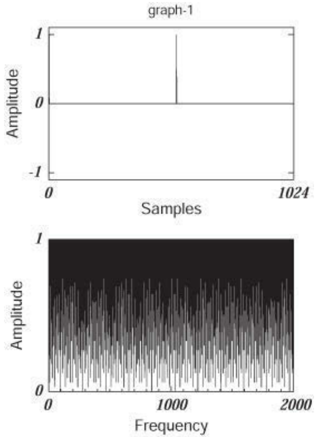


Figure: An impulse spike (Farnell 2010, fig. 7.15.)

Envelopes

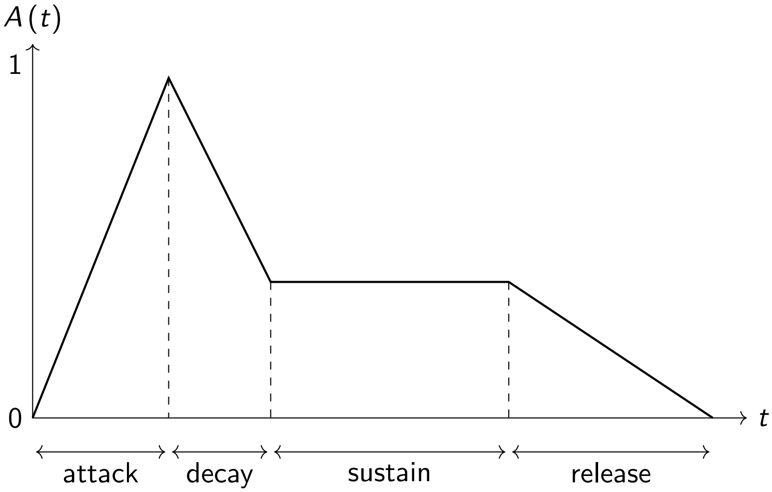


Figure: Linear ADSR envelope

Envelopes

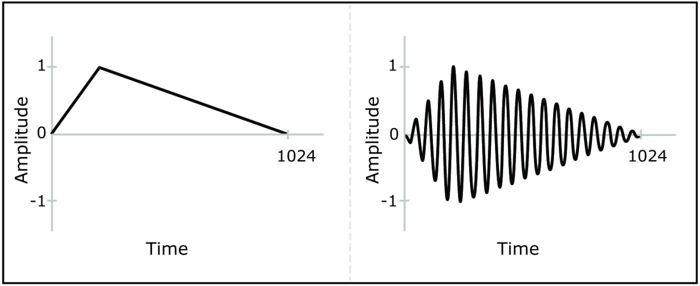


Figure: Envelope control signals (Image by MIT OpenCourseWare, after Farnell 2010, fig. 7.17.)

Visualization as a spectrogram

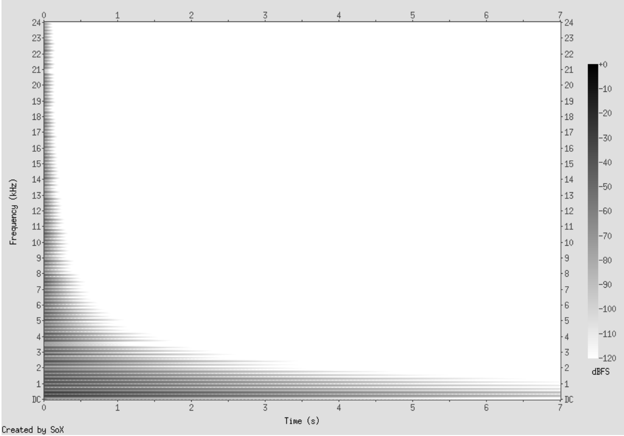


Figure: Spectrogram of a synthesized plucked guitar string

Visualization as a spectrogram

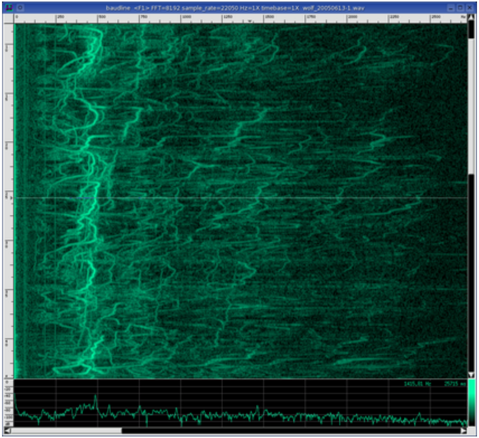


Figure: Spectrogram of howling wolves in Baudline (Courtesy of SigBlips. Used with permission)

Mass on a spring

Differential equation

$$\frac{d^2x}{dt^2} + \frac{kx}{m} = 0$$

Solution (cf., Farnell 2010, eq. 4.6)

$$f = \frac{\sqrt{\frac{k}{m}}}{2\pi}$$

- x ... displacement (m)
- t ... time (s)
- k ... stiffness ($\mathrm{N\,m^{-1}}$)
- m ... mass (kg)
- f ... frequency (Hz)

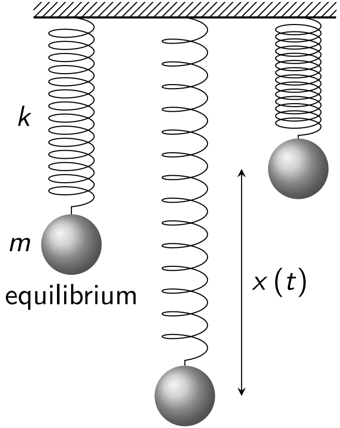


Figure: Oscillation of a mass on a spring

Pendulum

Differential equation

$$\frac{d^2\theta}{dt^2} + \frac{g\theta}{l} = 0$$

Solution (Farnell 2010, eq. 4.9)

$$f = \frac{1}{2\pi\sqrt{\frac{l}{g}}}$$

- θ ... angle (°)
- t ... time (s)
- g ... gravitational acceleration ($9.81\,\mathrm{m\,s^{-2}}$)
- l ... pendulum length (m)
- f ... frequency (Hz)

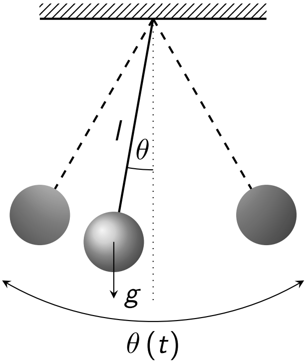


Figure: Pendulum

LC network

Differential equation

$$\frac{d^2 I}{dt^2} + \frac{I}{LC} = 0$$

Solution (Farnell 2010, eq. 4.11)

$$f = \frac{1}{2\pi\sqrt{LC}}$$

- I ... current (A)
- t ... time (s)
- L ... inductance (H)
- C ... capacitance (F)
- f ... frequency (Hz)

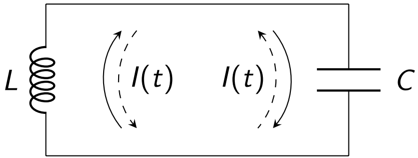


Figure: LC network

Closed-ended pipe

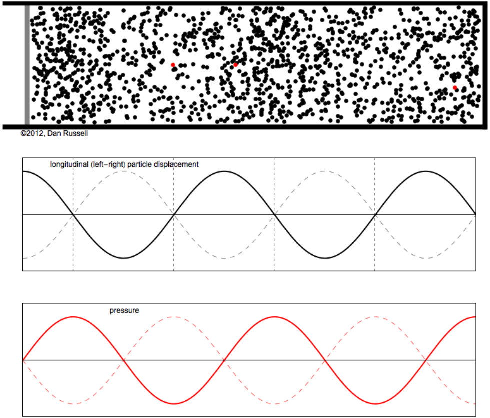


Figure: Particle displacement (center) vs. sound pressure (bottom) in a closed pipe (© Daniel A. Russell. Grad. Prog. Acoustics, Penn State. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/)

Closed-ended pipe

Without end correction

$$f_n = \frac{n \cdot c}{4 \cdot l}$$

With end correction

$$f_n = \frac{n \cdot c}{4\,(l + 0.4d)}$$

- f_n ... modes (Hz)
- $n \in 2\mathbb{N} - 1 = 1, 3, 5, \ldots$
- c ... speed of sound $(\mathrm{m\,s^{-1}})$
- l ... pipe length (m)
- d ... pipe diameter (m)

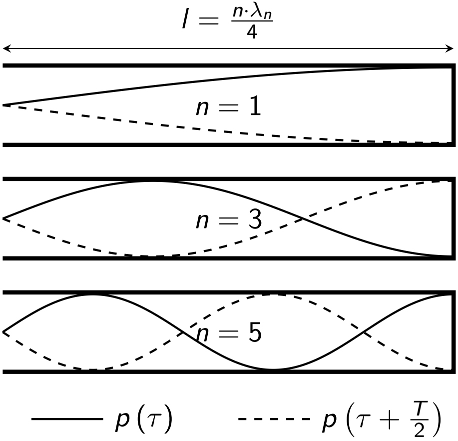


Figure: Sound pressure distribution in a closed-ended pipe

Open-ended pipe

Without end correction

$$f_n = \frac{n \cdot c}{2 \cdot l}$$

With end correction

$$f_n = \frac{n \cdot c}{2\,(l + 0.8d)}$$

- f_n ... modes (Hz)
- $n \in \mathbb{N} = 1, 2, 3, \ldots$
- c ... speed of sound $\left(\mathrm{m\,s^{-1}}\right)$
- l ... pipe length (m)
- d ... pipe diameter (m)

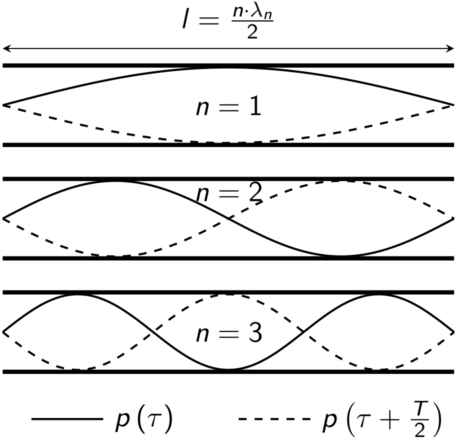


Figure: Sound pressure distribution in an open-ended pipe

String

Plucked string (cf., Farnell 2010, eq. 4.12)

$$f_n = \frac{n}{2l}\sqrt{\frac{T}{\mu}}$$

- f_n ... modes (Hz)
- n ... mode number ($n \in \mathbb{N} = 1, 2, 3, \ldots$)
- l ... string length (m)
- T ... linear tension (N)
- μ ... linear density ($\mathrm{kg\,m^{-1}}$)

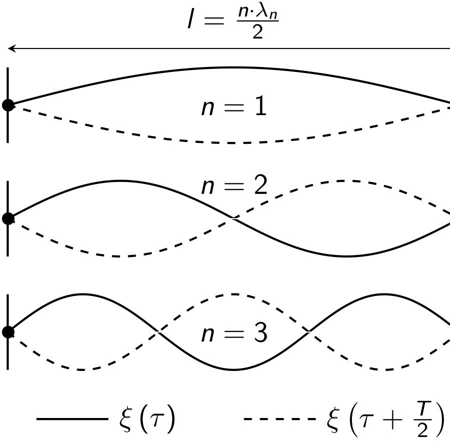


Figure: Vibrational modes of a string, showing particle displacement $\xi(t)$

Helmholtz resonator

Without end correction (Farnell 2010, eq. 5.16)

$$f = \frac{c \cdot d}{4\pi}\sqrt{\frac{\pi}{V \cdot l}}$$

- f … resonant frequency (Hz)
- c … speed of sound in air ($\mathrm{m\,s^{-1}}$)
- d … neck diameter (m)
- V … resonator volume ($\mathrm{m^3}$)
- l … neck length (m)

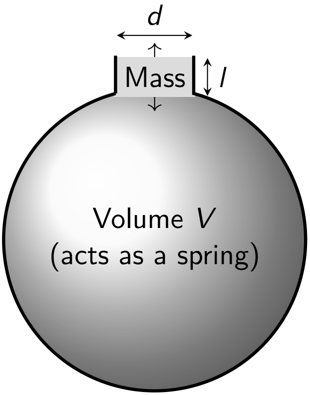


Figure: Helmholtz resonator

Transverse vibrations of bars, rods, and tubes

$$R = \frac{d}{\sqrt{12}}$$

$$R = \frac{d}{4}$$

$$R = \frac{\sqrt{r_i^2 + r_o^2}}{2}$$

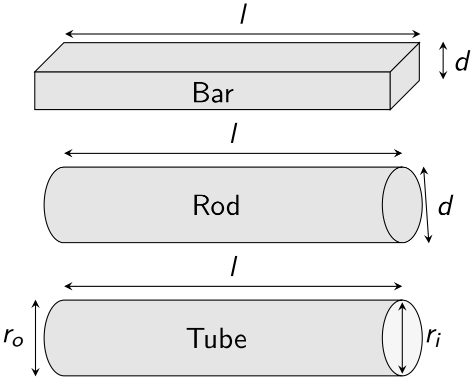


Figure: Difference between bar, rod, and tube (according to Hartmann 2013, p. 270)

Bar, rod, or tube with free ends

Modes (cf., Farnell 2010, eq. 4.16)

$$f_1 = \frac{3.5608}{l^2}\sqrt{\frac{ER^2}{\rho}}$$

$$f_2 = 2.756 \cdot f_1$$

$$f_3 = 5.404 \cdot f_1$$

$$f_4 = 8.932 \cdot f_1$$

$$f_5 = 13.344 \cdot f_1$$

$$f_6 = 18.638 \cdot f_1$$

- f_n ... modes (Hz)
- l ... bar length (m)
- E ... Young's modulus (Pa)
- R ... radius of gyration (m)
- ρ ... material density (kg m^{-3})

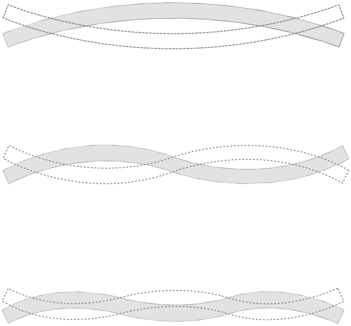
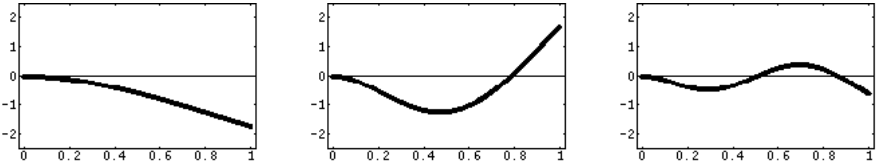


Figure: Modes $n = 1 \ldots 3$ of a free bar (Image by MIT OpenCourseWare, after Benson 2008, p. 120.)

Bar, rod, or tube clamped at one end



(a) $n = 1$ (b) $n = 2$ (c) $n = 3$

Figure: Clamped bar ()

Bar, rod, or tube clamped at one end

Modes (Farnell 2010, eq. 4.15)

$$f_1 = \frac{0.5596}{l^2}\sqrt{\frac{ER^2}{\rho}}$$

$$f_2 = 6.267 \cdot f_1$$

$$f_3 = 17.547 \cdot f_1$$

$$f_4 = 34.386 \cdot f_1$$

$$f_5 = 56.843 \cdot f_1$$

$$f_6 = 84.913 \cdot f_1$$

- f_n … modes (Hz)
- l … bar length (m)
- E … Young's modulus (Pa)
- R … radius of gyration (m)
- ρ … material density ($\mathrm{kg\,m^{-3}}$)

Bar, rod, or tube clamped at one end

Material	E / 10^{10}Pa	ρ / 10^3kg m^{-3}
Aluminium	7.05	2.7
Brass	10.05 ± 0.35	8.48
Copper	12.98	8.79
Gold	7.8	19.29
Iron	21.2	7.87
Lead	1.62	11.35
Silver	8.27	10.5
Steel	21.0	7.82
Zinc	9.0	7.12
Glass	6.1 ± 1.0	2.6 ± 0.2
Rosewood	1.4 ± 0.2	0.86 ± 0.04

Table: Young's modulus E and density ρ of different materials (Benson 2008, p. 117)

Stretched circular membrane

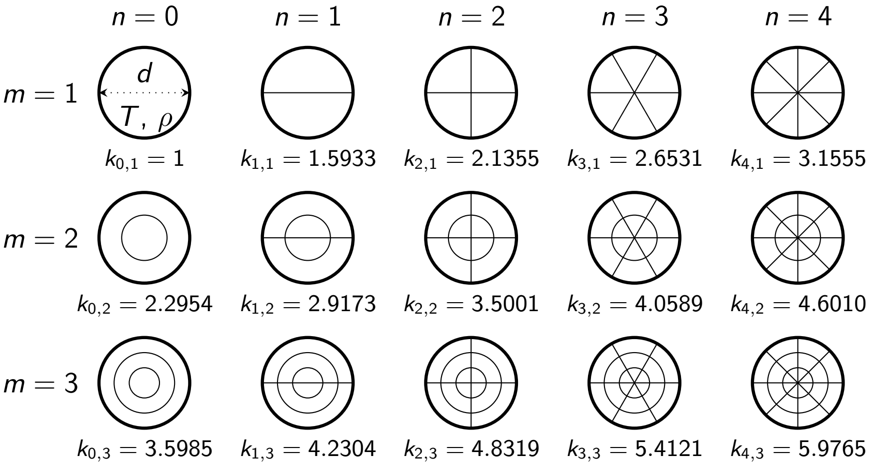


Figure: Modes of a stretched circular membrane (after Benson 2008, p. 106)

Stretched circular membrane

Modes (Nave 2015)

$$f_{0,1} = 0.766 \frac{\sqrt{\frac{T}{\rho}}}{d}$$

$$f_{n,m} = k_{n,m} \cdot f_{0,1}$$

- $f_{n,m}$ ... modes (Hz)
- n ... axial mode number ($n \in \mathbb{Z}^*$)
- m ... radial mode number ($m \in \mathbb{N}$)
- T ... surface tension ($\mathrm{N\,m^{-1}}$)
- ρ ... area density ($\mathrm{kg\,m^{-2}}$)
- d ... membrane diameter (m)
- $k_{n,m}$ ...modal ratios

$k_{n,m}$	$n = 0$	$n = 1$	$n = 2$	$n = 3$	$n = 4$
$m = 1$	1	1.5933	2.1355	2.6531	3.1555
$m = 2$	2.2954	2.9173	3.5001	4.0589	4.6010
$m = 3$	3.5985	4.2304	4.8319	5.4121	5.9765

Table: Modal ratios $k_{n,m}$ of a stretched circular membrane (Benson 2008, p. 106)

21M.380 Music and Technology
Sound Design

Lecture 6: Pd programming concepts

Massachusetts Institute of Technology
Music and Theater Arts

Monday, February 22, 2016

Blocks

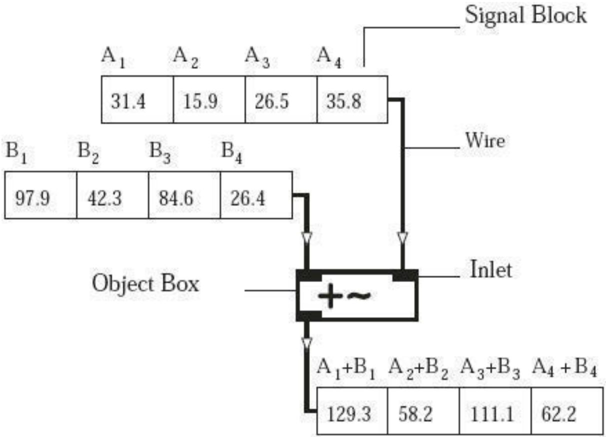


Figure: Pd processes audio data in blocks. Change block size via Media|Preferences › Audio settings... › Block size (Farnell 2010, fig. 11.1.)

Blocks

Pd object	Functionality
[env~]	Envelope follower (RMS amplitude)
[snapshot~]	Sample signal amplitude (instantaneous)
[print~]	Print content of audio blocks

Table: Conversion from audio signals to messages

Sending and receiving audio signals

Pd object	Abbr.	Functionality
[send~]	[s~]	Nonlocal signal connection with fanout
[receive~]	[r~]	Get signal from [send~]
[throw~]		Add signal to summing bus
[catch~]		Define and read a summing bus
[inlet~]		Add signal inlet to subpatch or abstraction
[outlet~]		Add signal outlet to subpatch or abstraction

Table: Sending and receiving audio signals

Oscillators

Pd object	Functionality	Output range
[osc~]	(Co)sine oscillator	-1 to $+1$
[noise~]	White noise generator	-1 to $+1$
[phasor~]	Sawtooth oscillator	0 to $+1$

Table: Audio oscillators

Oscillators

Pd object	Functionality
`[tabosc4~]`	Read table as waveform period
`[tabread~]`	Read from table
`[tabread4~]`	Read from table (interpolated version)

Table: Reading from tables at audio rate

Envelope generators

Pd object	Functionality
[line~]	Generate audio ramps (i.e., simple envelopes)
[vline~]	Deluxe version of [line~]

Table: Envelope generators

Soundcard input and output

Pd object	Functionality
[adc~]	Soundcard audio input
[dac~]	Soundcard audio output

Table: Soundcard input and output

Audio filters

Pd object	Functionality
[rpole~]	Real-valued one-pole filter
[rzero~]	Real-valued one-pole filter
[cpole~]	Complex-valued one-zero filter
[czero~]	Complex-valued one-zero filter
[biquad~]	Static biquad filter

Table: Raw audio filter objects

Audio filters

Pd object	Functionality
[lop~]	Low pass filter
[hip~]	High pass filter
[bp~]	Band pass filter
[vcf~]	Voltage-controlled bandpass filter

Table: User-friendly audio filter objects

Doing math on audio signals

Pd object	Functionality
`[+~]`	Add two signals
`[-~]`	Subtract right from left signal
`[/~]`	Divide left by right signal
`[*~]`	Multiply two signals
`[wrap~]`	Constrain signal between 0 and $+1$

Table: List of arithmetic operators (Farnell 2010, fig. 11.8)

Doing math on audio signals

Pd object	Functionality
[cos~]	Cosine of incoming signal times 2π
[log~]	Signal version of natural log
[sqrt~]	Square root for signals
[q8_sqrt~]	Fast square root with less accuracy
[pow~]	Signal version of power function

Table: List of trig and higher math operators (Farnell 2010, fig. 11.9)

Delay lines

Pd object	Functionality
`[delwrite~]`	Write to delay line
`[delread~]`	Read from delay line
`[vd~]`	Read from delay line (with variable delay time)

Table: Audio delay objects

Subpatches

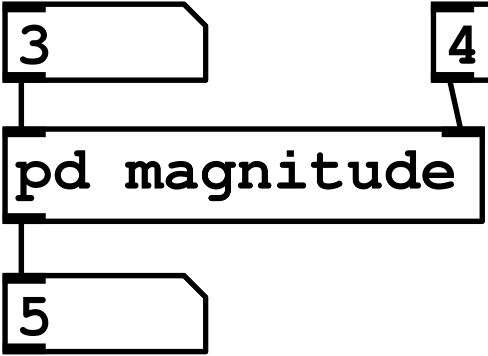


Figure: Pd patch with a [pd magnitude] subpatch (Farnell 2010, fig. 12.4)

Abstractions

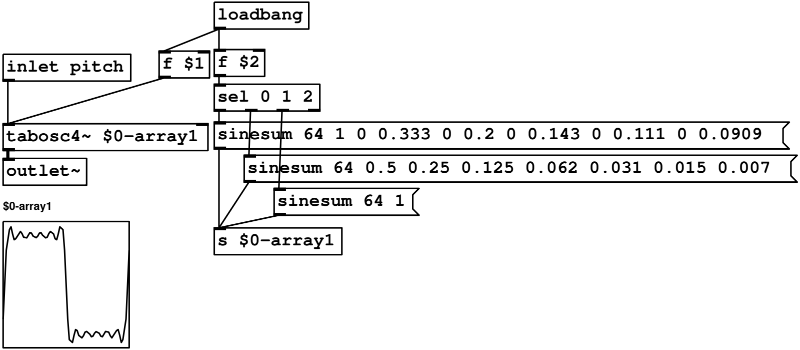


Figure: Table oscillator abstraction [my-tabosc2] with initialised frequency and shape (Farnell 2010, fig. 12.10)

Creation arguments

	In a message means...	In an abstraction means...
`$0`	t.b.c.	Some unique number
`$1`	1st element	1st creation argument
`$2`	2nd element	2nd creation argument
...	...	...
`$N`	Nth element	Nth creation argument
	... of incoming list	... of current instance

Table: Meaning of `$` variables in Pd

Creation arguments

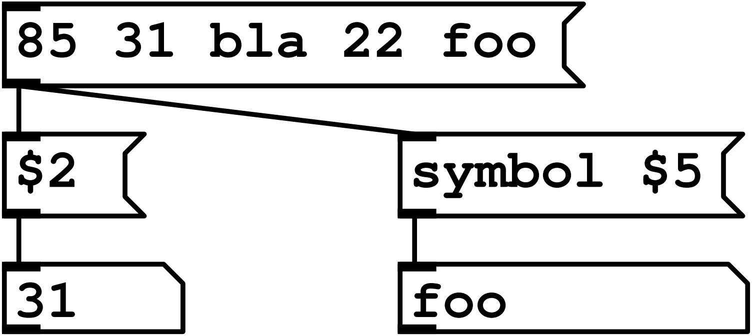


Figure: Dollar signs in messages

Defaults and states

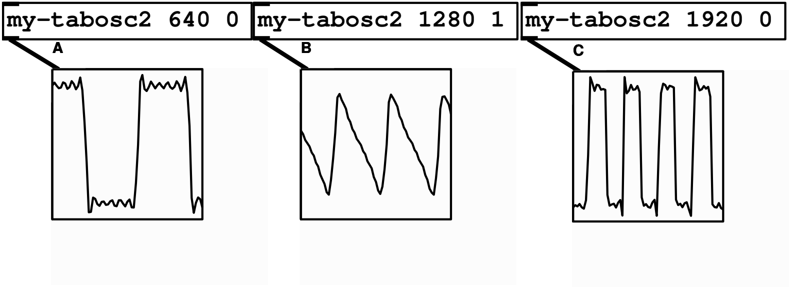


Figure: Three different waveforms and frequencies from the same table oscillator abstraction (Farnell 2010, fig. 12.11)

21M.380 Music and Technology
Sound Design

Lecture 7: Perception of sound

Massachusetts Institute of Technology
Music and Theater Arts

Wednesday, February 24, 2016

The human auditory system

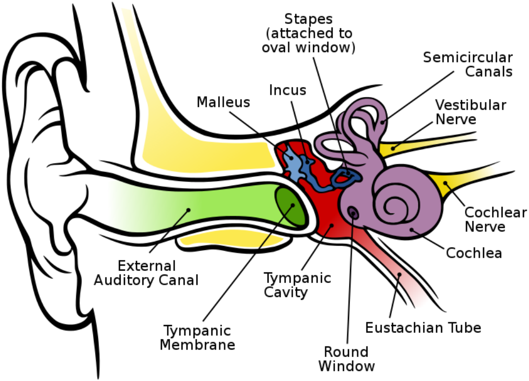


Figure: Anatomy of the human ear (Courtesy of Lars Chittka and Axel Brockmann. Used with permission.)

The human auditory system

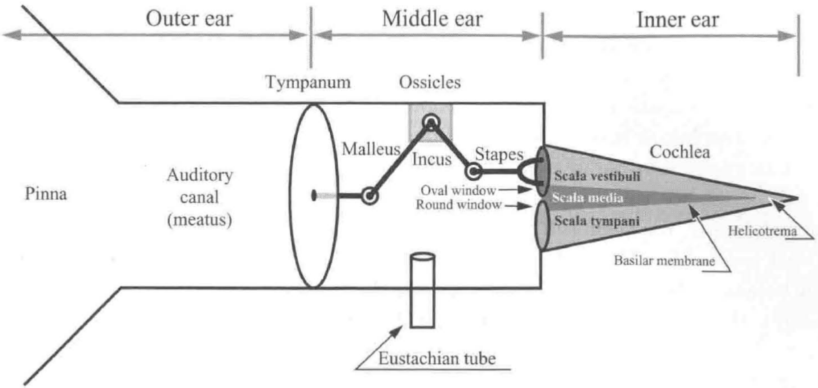


Figure: Schematic diagram of the human ear (Loy 2007, p. 151. Courtesy of MIT Press. Used with permission.
https://mitpress.mit.edu/books/musimathics)

Limits of human hearing

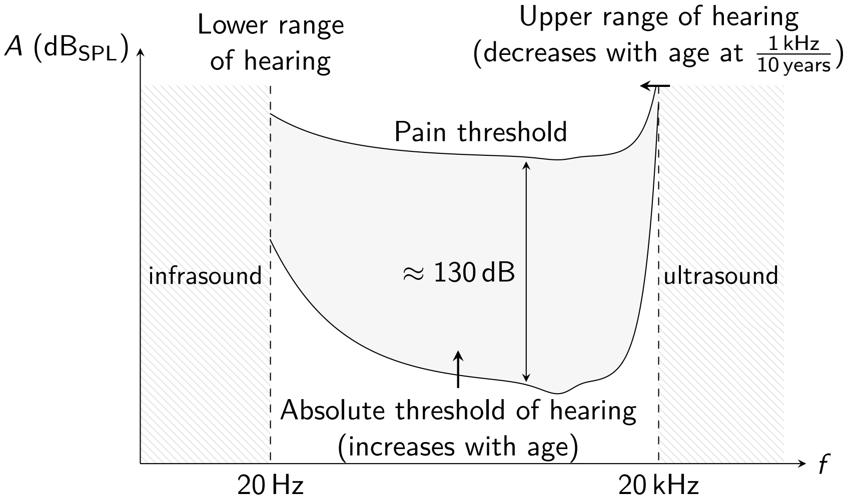


Figure: Amplitude and frequency limits of human hearing

Physics vs. perception of sound

Physical property	Perceptual effect
Amplitude	Loudness
Fundamental frequency	Pitch
Spectrum	Timbre
Sound source position	Perceived direction

Table: Some psychoacoustic relationships

Just noticeable difference

Definition (Just noticeable difference (JND))

The smallest change of a physical quantity that results in a perceptual effect

Examples

- JND for amplitude ($\approx 1\,\mathrm{dB}$)
- JND for frequency (depends on range)
- JND of source position ($\approx 1^\circ$ for front direction)

Sound localization

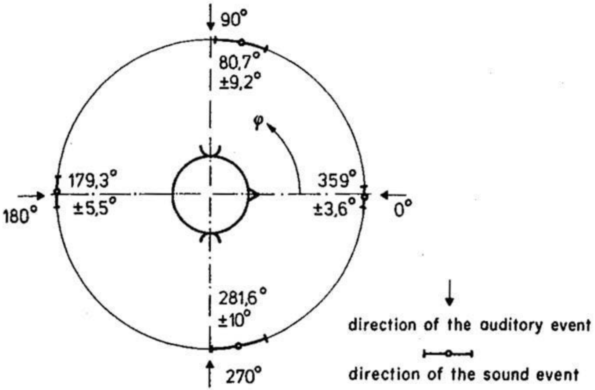


Figure: Localization blur in the horizontal plane. Experimental setup: 100 ms white noise pulses, head immobilized. (Blauert 1996, p. 41.

Sound localization

Accurate sound localisation is facilitated by

- High-frequency sounds with sharp attacks
- Free space without reflections
- Ability of the listener to move her head

Interaural time difference

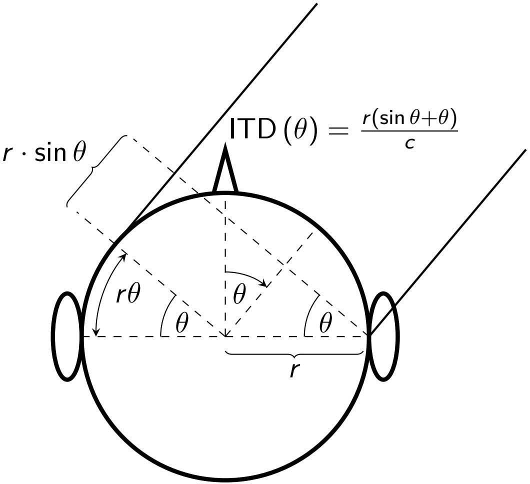


Figure: Simple model of interaural time differences

Interaural intensity difference

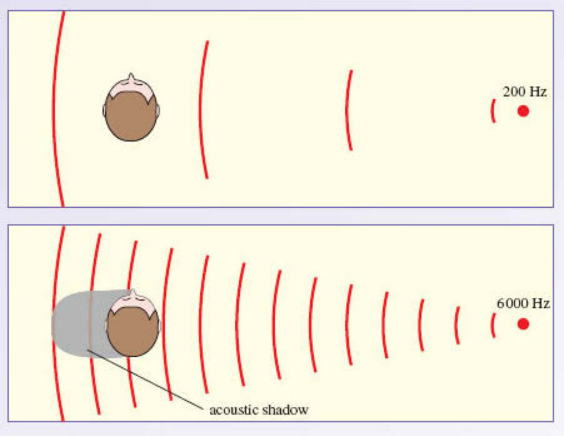


Figure: Interaural intensity differences due to acoustic head shadow (An OpenLearn chunk used by permission of The Open University copyright © 2015. BY-NC-SA)

ITD and IID over the frequency range

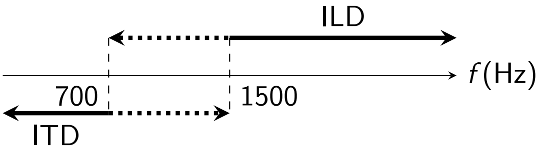


Figure: Interaural time and level differences complement each other over the audible frequency range.

Equal loudness contours

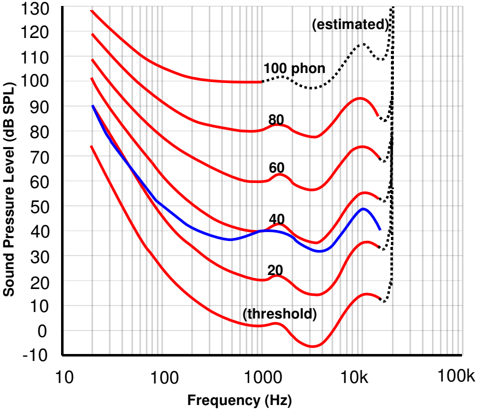


Figure: Equal loudness contours. Red: ISO226:2003 revision. Blue: Original ISO standard for 40 phon (Ⓟ Public domain image. With edits. Source: `https://en.wikipedia.org/wiki/File:Lindos1.svg`)

Critical bands

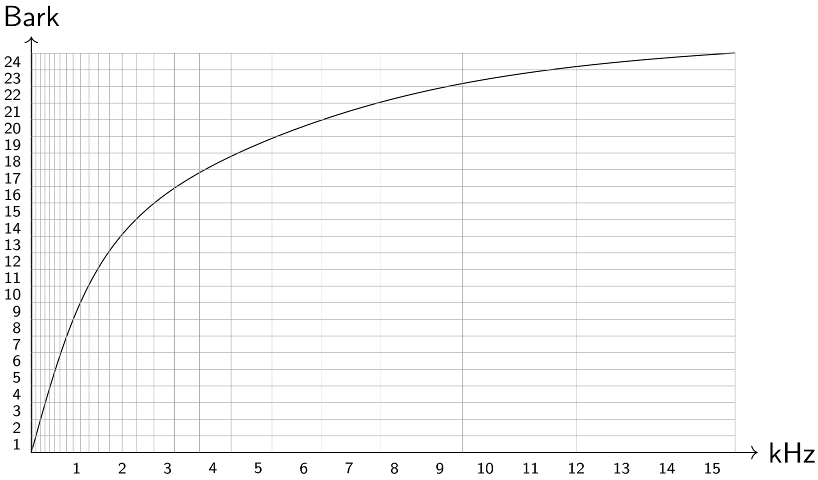
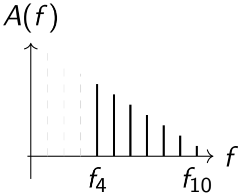
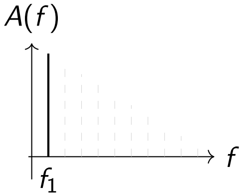


Figure: The Bark scale

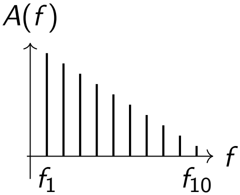
Missing fundamental



(a) Harmonics 4–10



(b) Fundamental



(c) Harmonics 1–10

Figure: Missing fundamental

Examples

- Some woodwind instruments (e.g., oboe)
- *MaxxBass* plug-in (© Waves Inc.)
- Extending the perceived range of subwoofers or organ pipes

Masking in the frequency domain

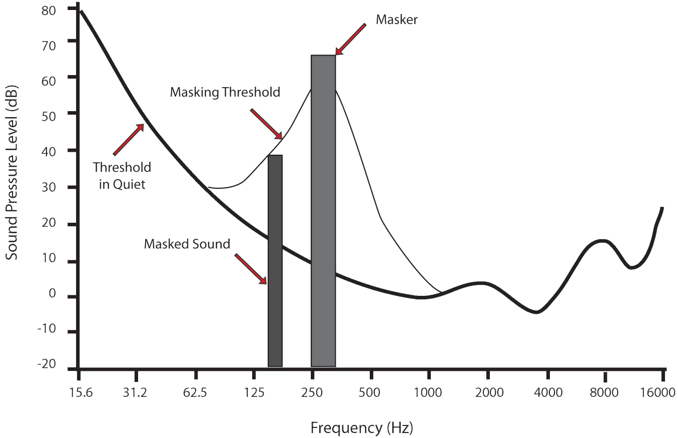


Figure: Masking in the frequency domain (Public domain image. Source: `https://en.wikipedia.org/wiki/File:Audio_Mask_Graph.png`)

Masking in the time domain

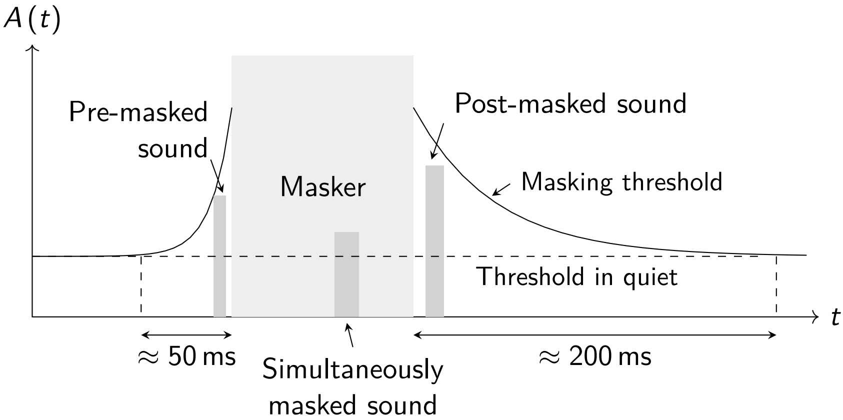
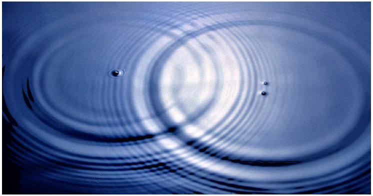


Figure: Masking in the time domain

Auditory scene analysis

(a) Sailboats (Courtesy of Ron Lute on Flickr. Used with permission.)



(b) Water ripples (Courtesy of Andrew Davidhazy. Used with permission)

Figure: A visual analogy of auditory scene analysis

Segregation

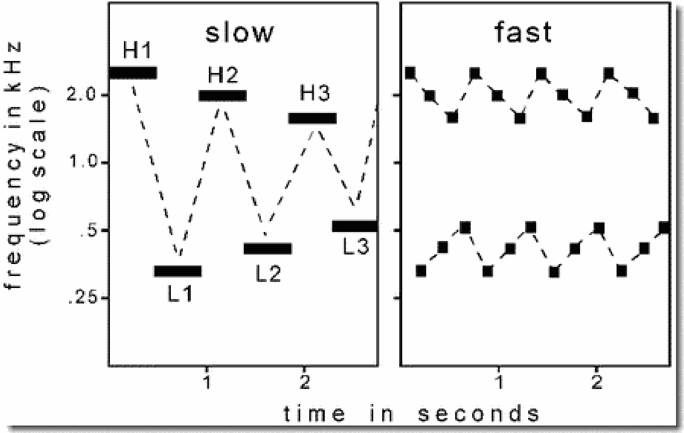


Figure: Stream segregation in a cycle of six tones (Bregman and Ahad (1996). Demonstrations of Auditory Scene Analysis: The Perceptual Organization of Sound. Audio compact disk. Montréal, Canada: Auditory Perception laboratory, Psychology Department, McGill University. Distributed by MIT Press)

Segregation

Figure: Segregation of high notes from low ones in a sonata by Telemann (Bregman and Ahad (1996). Demonstrations of Auditory Scene Analysis: The Perceptual Organization of Sound. Audio compact disk. Montréal, Canada: Auditory Perception laboratory, Psychology Department, McGill University. Distributed by MIT Press)

Harmonicity

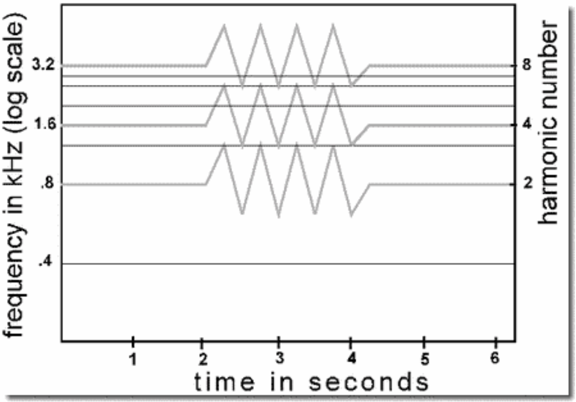


Figure: Fusion by common frequency change (principle of harmonicity) (Bregman and Ahad (1996). Demonstrations of Auditory Scene Analysis: The Perceptual Organization of Sound. Audio compact disk. Montréal, Canada: Auditory Perception laboratory, Psychology Department, McGill University. Distributed by MIT Press)

Continuity

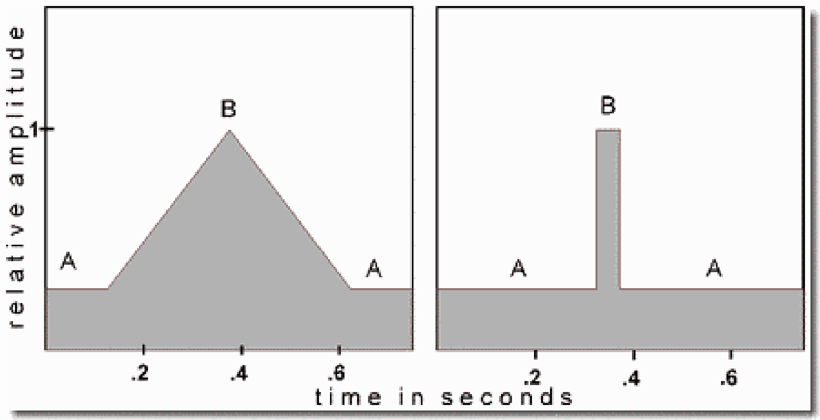


Figure: Homophonic continuity and rise time (Bregman and Ahad (1996). Demonstrations of Auditory Scene Analysis: The Perceptual Organization of Sound. Audio compact disk. Montréal, Canada: Auditory Perception laboratory, Psychology Department, McGill University. Distributed by MIT Press)

Momentum

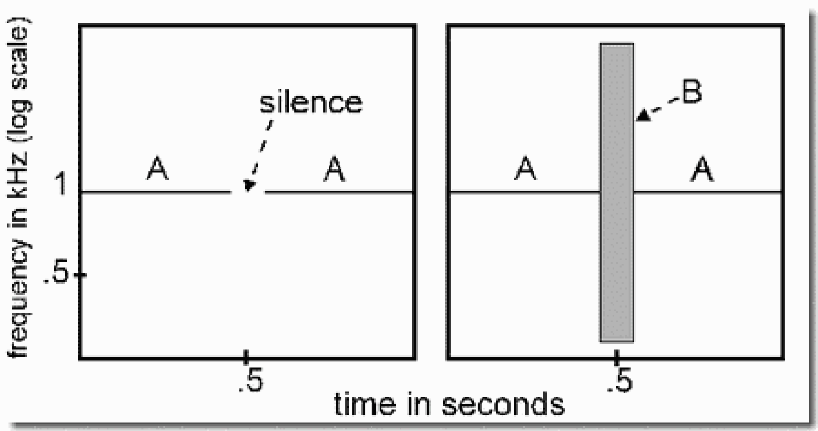


Figure: Apparent continuity (old-plus-new heuristic) (Bregman and Ahad (1996). Demonstrations of Auditory Scene Analysis: The Perceptual Organization of Sound. Audio compact disk. Montréal, Canada: Auditory Perception laboratory, Psychology Department, McGill University. Distributed by MIT Press)

Temporal correlation

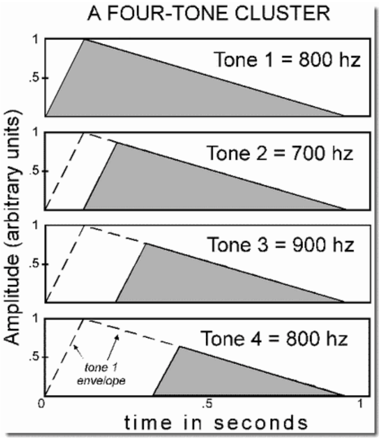


Figure: Effect of rate of onset on segregation (Bregman and Ahad (1996). Demonstrations of Auditory Scene Analysis: The Perceptual Organization of Sound. Audio compact disk. Montréal, Canada: Auditory Perception laboratory, Psychology Department, McGill University. Distributed by MIT Press)

Coherence

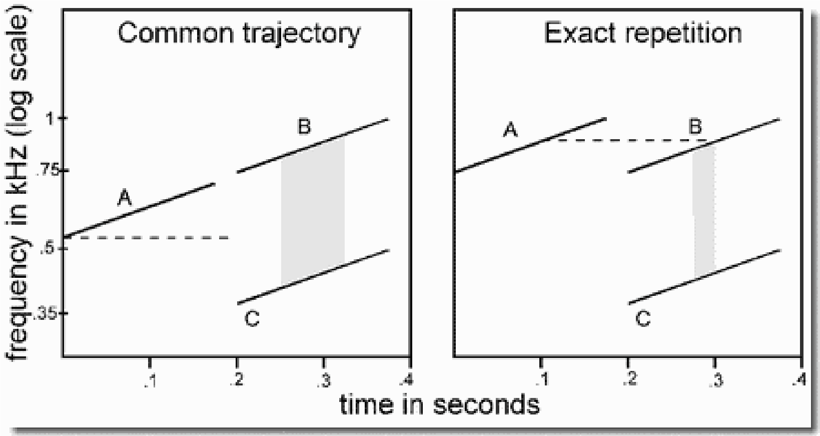


Figure: Capturing a component glide in a mixture of glides (Bregman and Ahad (1996). Demonstrations of Auditory Scene Analysis: The Perceptual Organization of Sound. Audio compact disk. Montréal, Canada: Auditory Perception laboratory, Psychology Department, McGill University. Distributed by MIT Press)

21M.380 Music and Technology
Sound Design

Lecture 8: Soundwalk

Massachusetts Institute of Technology
Music and Theater Arts

Monday, February 29, 2016

Soundwalk

Wherever we go we will give our ears priority. (Westerkamp 2007, p. 49)

21M.380 Music and Technology
Sound Design

Lecture 9: Shaping sound with Pd

Massachusetts Institute of Technology
Music and Theater Arts

Wednesday, March 2, 2016

Scaling and shifting a signal

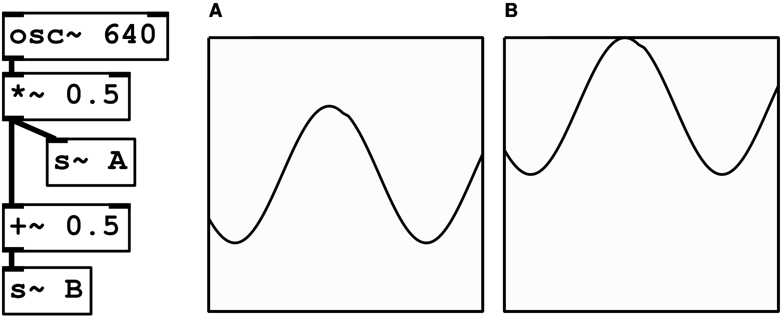


Figure: $B = \frac{A+1}{2}$ (Farnell 2010, fig. 13.2)

Signal inverse

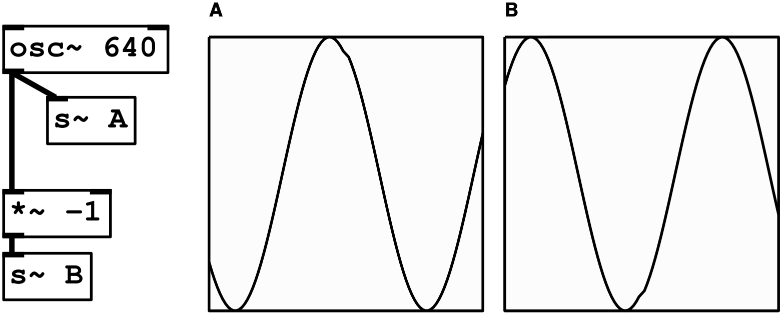


Figure: $B = -A$ (Farnell 2010, fig. 13.3)

Signal complement

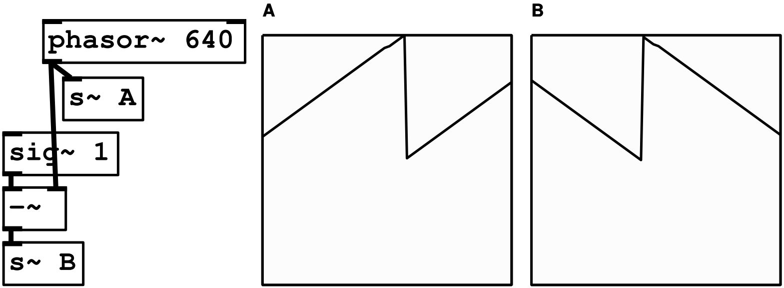


Figure: $B = 1 - A$ (Farnell 2010, fig. 13.4)

Signal reciprocal

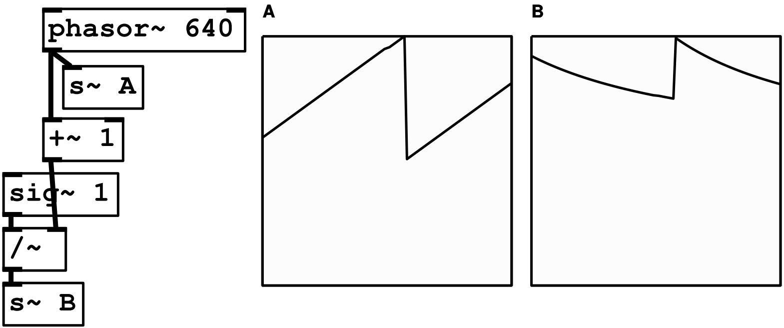


Figure: $B = \frac{1}{1+A}$ (Farnell 2010, fig. 13.5)

Minimum and maximum

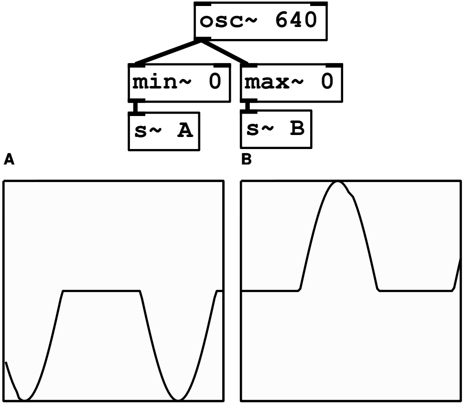
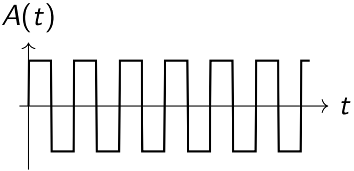
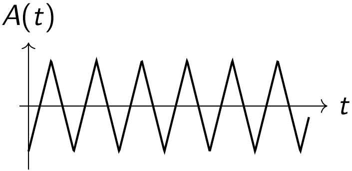


Figure: Min and max of a signal (Farnell 2010, fig. 13.6)

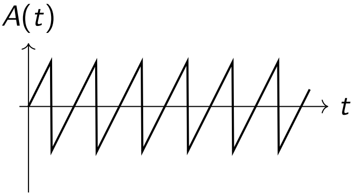
Basic waveforms



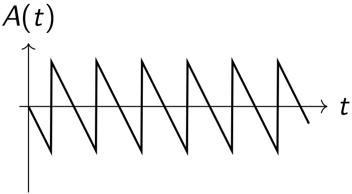
(a) Square wave



(b) Triangle wave



(c) Sawtooth wave



(d) Inverse sawtooth

Figure: Waveform archetypes

Square wave

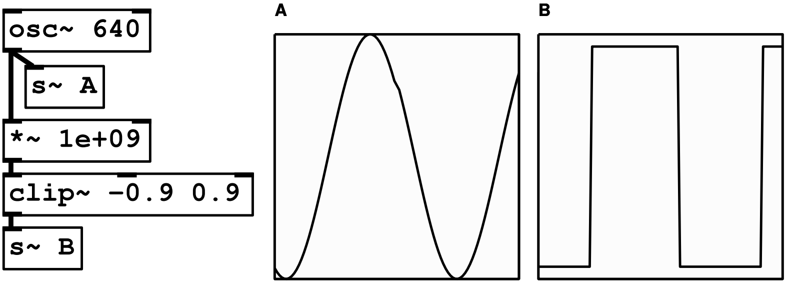


Figure: Square wave (Farnell 2010, fig. 13.7)

Triangle wave

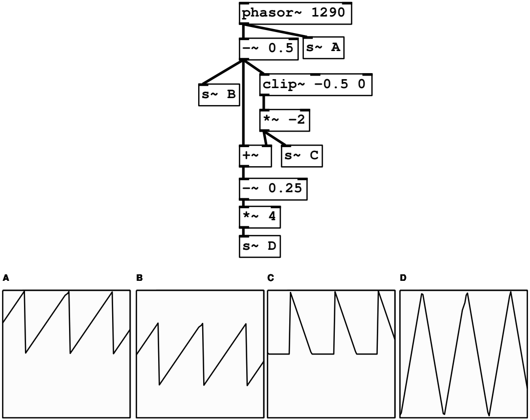


Figure: Triangle (Farnell 2010, fig. 13.8)

Triangle wave

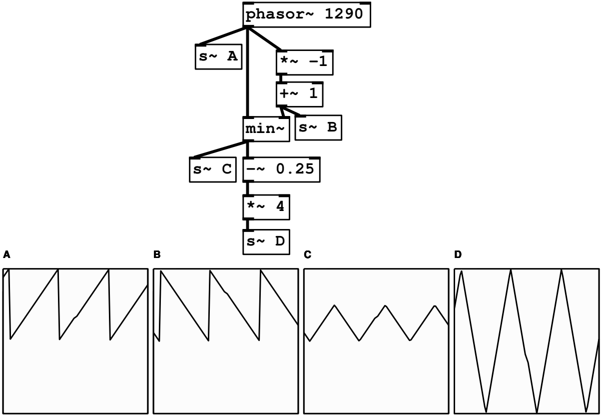


Figure: Another way to make a triangle wave (Farnell 2010, fig. 13.9)

Sawtooth wave

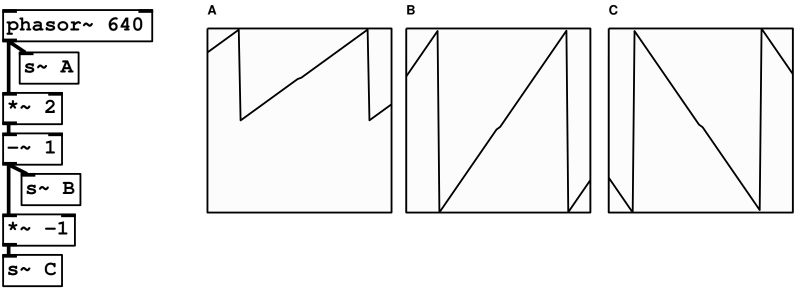


Figure: Sawtooth from phasor

Squaring and roots

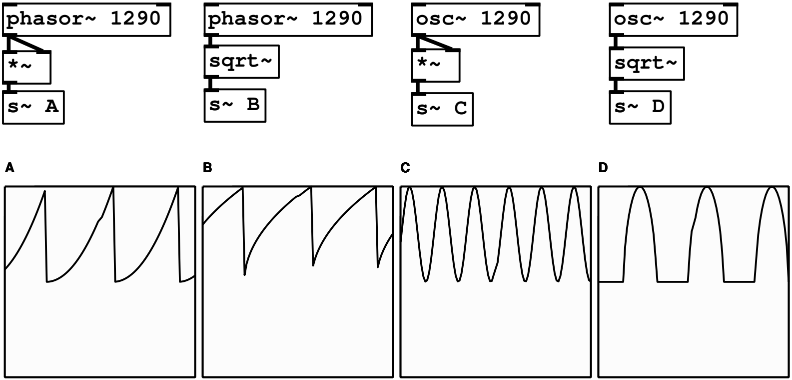


Figure: Square roots (Farnell 2010, fig. 13.10)

Pulse wave

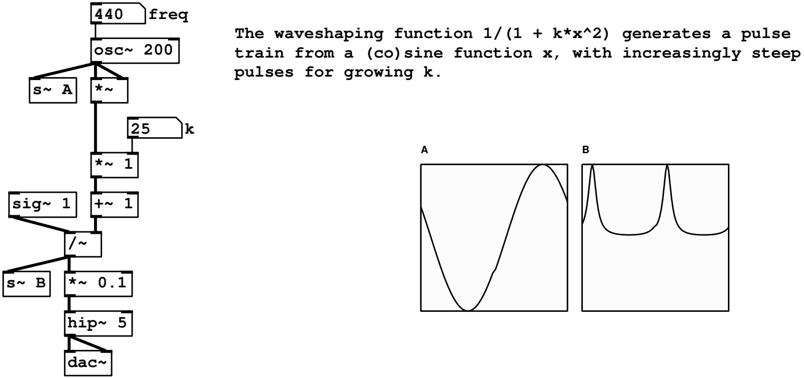


Figure: Generating a pulse train from a cosine oscillator with a $\frac{1}{1+kx^2}$ waveshaping function (cf., Farnell 2010, figs. 45.5, 46.2, 46.6, 50.10)

Pulse wave

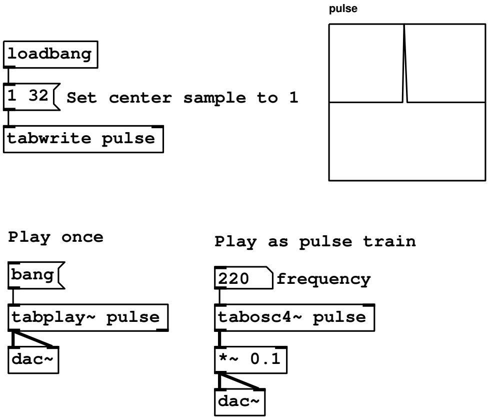


Figure: Generating a pulse or pulse train from a wavetable

Curved envelopes

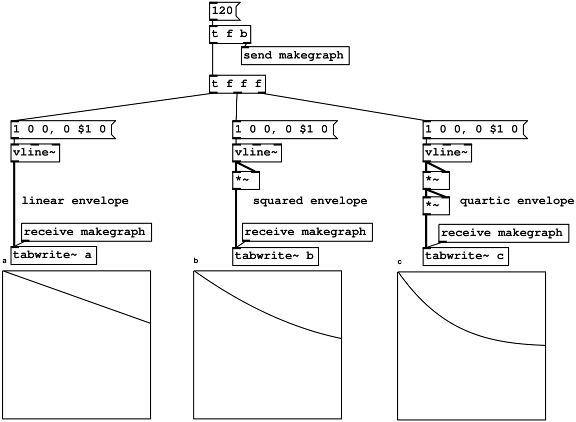


Figure: Linear, squared, and quartic decays (Farnell 2010, fig. 13.11)

Audio delays

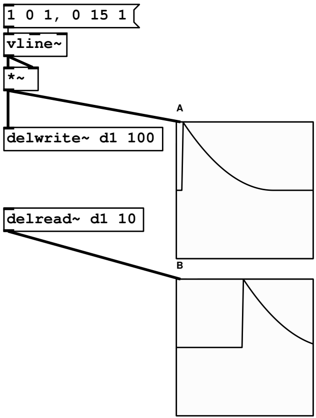


Figure: Delay (Farnell 2010, fig. 13.17)

Artifical reverb

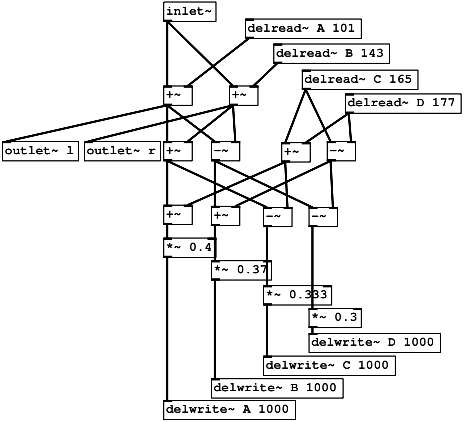


Figure: A recirculating Schroeder reverb effect (Farnell 2010, fig. 14.28)

Chorus effect

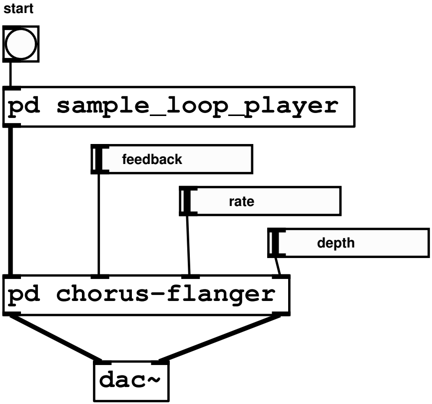
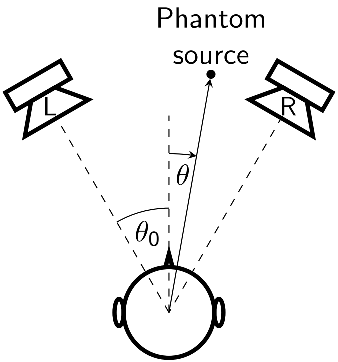
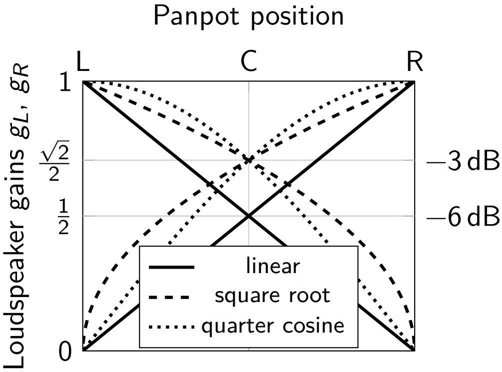


Figure: Chorus effect (Farnell 2010, fig. 14.27)

Panning



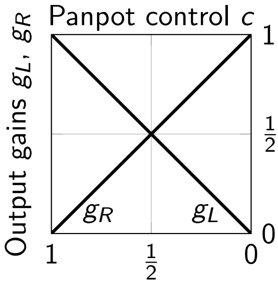
(a) Stereo loudspeaker setup



(b) Stereo panpot laws

Figure: Stereo panning

Linear panpot



Linear panpot

$$
\begin{aligned}
g_L &= c \\
g_R &= 1 - c
\end{aligned}
$$

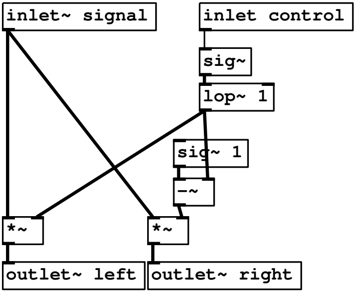
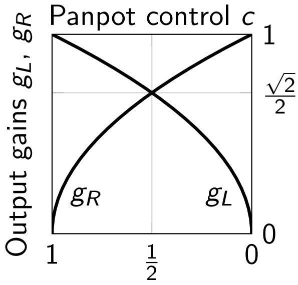


Figure: Linear panpot in Pd (Farnell 2010, fig. 14.7)

Square root panpot



Square root panpot

$$g_L = \sqrt{c}$$
$$g_R = \sqrt{1 - c}$$

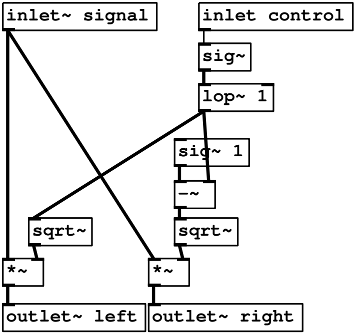
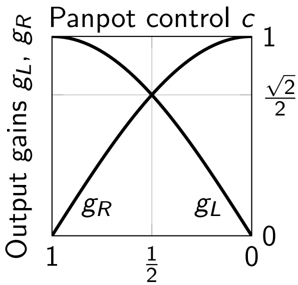


Figure: Square root panpot in Pd (Farnell 2010, fig. 14.8)

Quarter cosine panpot



Quarter cosine panpot

$$g_L = \cos\left(\frac{(1-c)\,\pi}{2}\right)$$

$$g_R = \cos\left(\frac{c\pi}{2}\right)$$

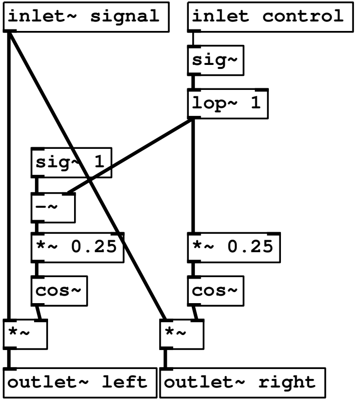


Figure: Quarter cosine panpot in Pd (Farnell 2010, fig. 14.9, with edits)

21M.380 Music and Technology
Sound Design

Lecture 10: Digital audio theory

Massachusetts Institute of Technology
Music and Theater Arts

Monday, March 7, 2016

Digital audio overview

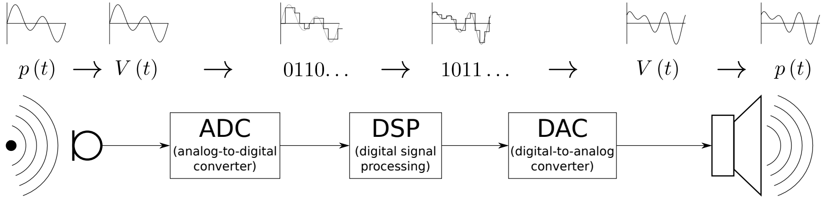


Figure: Digital reproduction chain

Magic numbers

Value	Unit	Meaning	Application
44.1	kHz	Sample rate	Audio CD
48 000	Hz		DAT tape
96	kHz		Audio production
192	kHz		Audio production
24	bit	Bit depth	Audio production
16	bit		Audio CD
32	bit		Audio production
256	$\mathrm{kbit\,s^{-1}}$	Bit rate	'High-quality' `.mp3`
192	$\mathrm{kbit\,s^{-1}}$		Common `.mp3` bit rate
128	$\mathrm{kbit\,s^{-1}}$		Common `.mp3` bit rate

Table: Magic numbers in digital audio

Analog-digital conversion

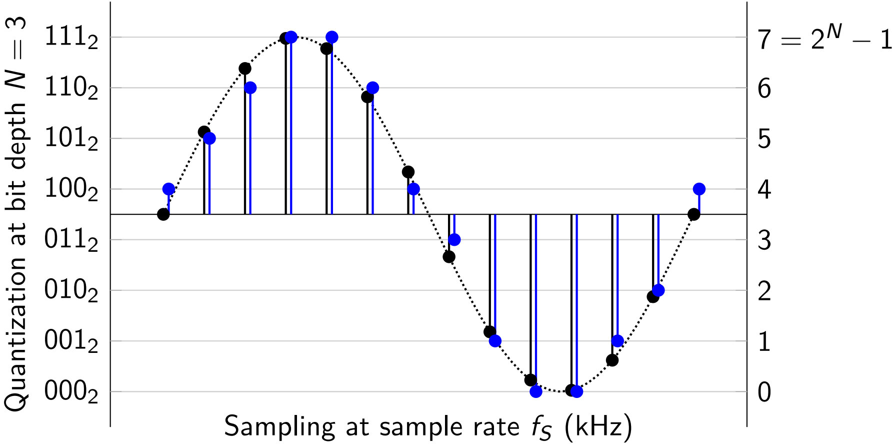


Figure: Analog/digital conversion

Sampling theorem and Nyquist frequency

Sampling theorem

$$f_S \overset{!}{>} 2 \cdot f_{max} \tag{3}$$

Nyquist frequency

$$f_N = \frac{f_S}{2} \tag{4}$$

- f_S ... Sample rate (Hz)
- f_{max} ... Highest frequency to be sampled (Hz)
- f_N ... Nyquist frequency (Hz)

Sampling does not imply a loss of information (but quantization does)

A signal that has been sampled in compliance with the sampling theorem (but not quantized) can be truthfully restored.

Violating the sampling theorem results in aliasing

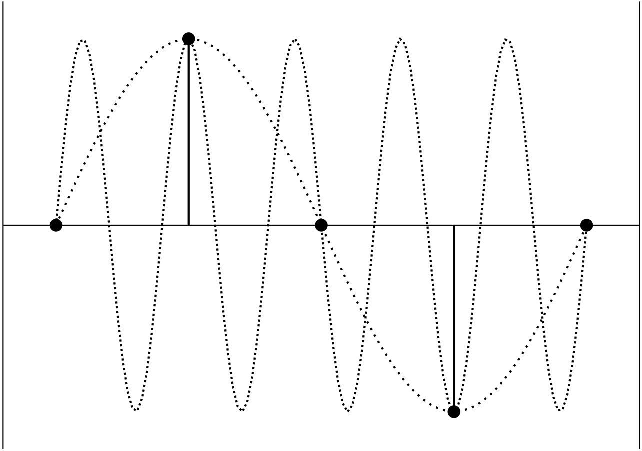


Figure: Violation of the sampling theorem creates an ambiguity

Violating the sampling theorem results in aliasing

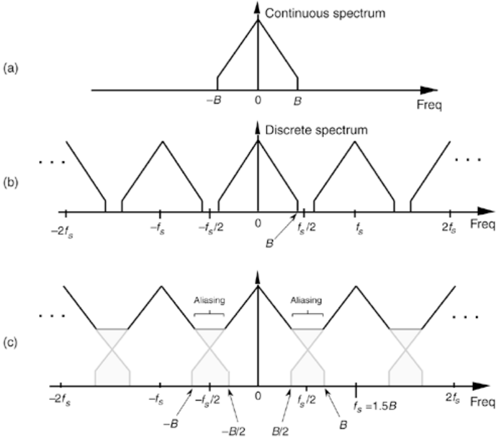


Figure: Sampling creates spectral sidebands of the original spectrum that repeat periodically around multiples of f_S. (Lyons 2004, fig. 2.4, with edits.

Violating the sampling theorem results in aliasing

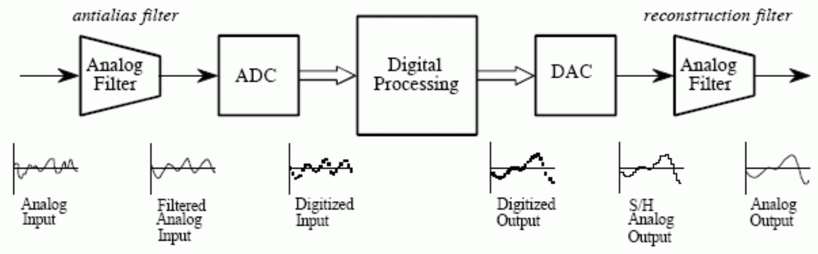


Figure: ADC/DAC conversion chain with anti-aliasing and reconstruction filters (S. W. Smith 1997, fig. 3.7. Courtesy of Steven W. Smith. Used with permission. Source: `http://www.dspguide.com/ch3/4.htm`)

Binary numbers

There are 10 types of people in the world... those who know binary and those who don't. (by courtesy of [redacted], MIT class of 2014)

Binary-to-decimal conversion

$$\begin{aligned} 1001_2 &= 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\ &= 8 + 0 + 0 + 1 \\ &= 9_{10} \end{aligned}$$

- Hence, 1001 binary equals 9 decimal
- Analogous: $975_{10} = 9 \cdot 10^2 + 7 \cdot 10^1 + 5 \cdot 10^0$

Bit depth

Numeric values that can be expressed by N bit

$$2^N \tag{5}$$

Examples

- 16 bit audio: $2 \times 10^{16} = 65\,536$
- 24 bit audio: $2 \times 10^{24} = 16\,777\,216$

Binary	Decimal
000_2	0
001_2	1
010_2	2
011_2	3
100_2	4
101_2	5
110_2	6
111_2	7

Table: Numeric values that can be expressed with a bit depth of $N = 3$

Higher bit depth provides more dynamic range

Dynamic range of digital audio

$$
\begin{aligned}
\Delta L_{dig} &= 20 \cdot \log_{10}\left(2^N\right) \\
&= 20 \cdot N \cdot \log_{10}(2) \\
&\approx 20 \cdot N \cdot 0.3 \\
&= (6 \cdot N)\,\mathrm{dB}
\end{aligned}
$$

- ΔL_{dig} ... dynamic range (dB)
- N ... bit depth ($N \in \mathbb{N}$)

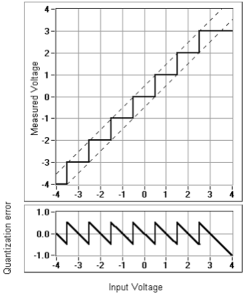


Figure: Quantizing error of a 3 bit A/D converter (© National Instruments Corporation. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/)

Digital audio file formats

Data compression	Codec	Container formats
Uncompressed	PCM	`.wav`, `.aif`, `.aiff`
Lossless (reversible)	FLAC ALAC	`.flac` `.m4a`

Lossy (irreversible)	MPEG layer III
	AAC
	Vorbis
	Opus
		`.mp3`
		`.m4a`, `.m4b`, `.aac`
		`.ogg`
		`.opus`

Table: Audio codecs and container formats

21M.380 Music and Technology
Sound Design

Lecture 11: Sound recording and editing techniques

Massachusetts Institute of Technology
Music and Theater Arts

Wednesday, March 9, 2016

Sound recording and editing techniques

spy 1:	*Picks up telephone* ***(sfx: Dialing tone from handset)***
spy 1:	*Dials number* ***(sfx: Ringing tone from handset)***
spy 2:	*"Hello, this is the Badger."*
spy 1:	*"This is Fox. The dog has the bone, the seagull flies tonight."*
spy 2:	*"Good, Fox. Now the Americans will pay for their deception... hold on..."*
	(sfx: click—telephone line goes dead)

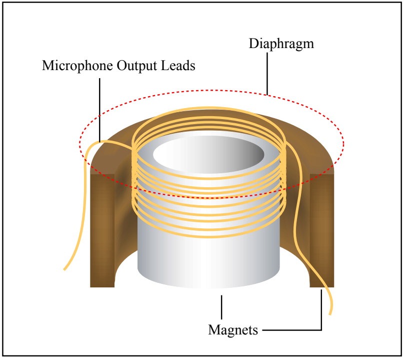
(Farnell 2010, ch. 25)

Sound recording and editing techniques

A	B	C	D	E
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]				

Table: Student groups

Dynamic microphones (moving coil, ribbon)

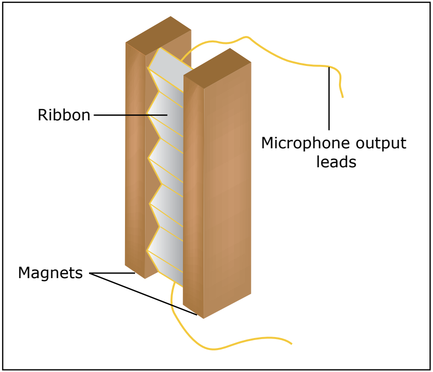


(a) Electromagnetic induction in a moving-coil microphone (Image by MIT OpenCourseWare)

(b) Shure SM58 (© Shure. With edits. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/)

Figure: Dynamic moving coil microphones

Dynamic microphones (moving coil, ribbon)



(a) Electromagnetic induction in a ribbon microphone (Image by MIT OpenCourseWare)

(b) Royer R-101 (© Royer Labs.)

Figure: Dynamic ribbon microphone

Condenser microphones

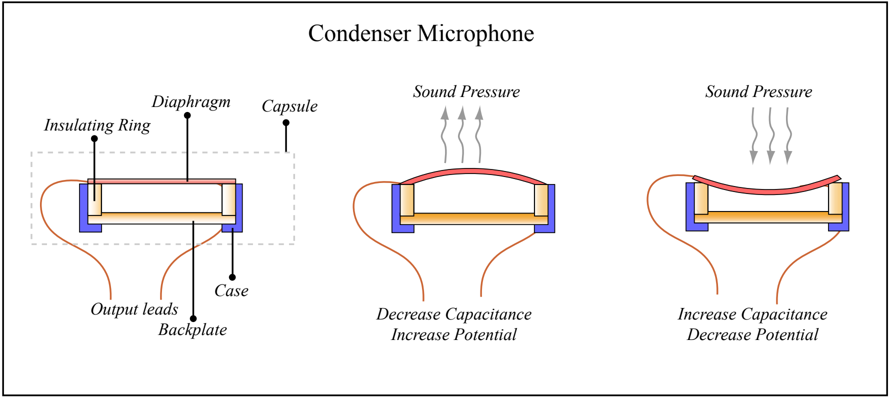


Figure: Capacitance in a condenser microphone (Image by MIT OpenCourseWare)

Condenser microphones

Figure: AKG C 414 XL II dual-large-diaphragm condenser microphone (© AKG Acoustics. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/)

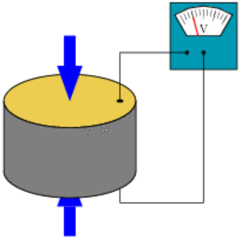
Condenser microphones

(a) Audio-Technica AT4041 small-diaphragm condenser (© Audio-Technica)

(b) Soundman OKM binaural in-ear electret condensers (© Soundman e. K.)

Figure: Codenser microphones ()

Piezo microphones

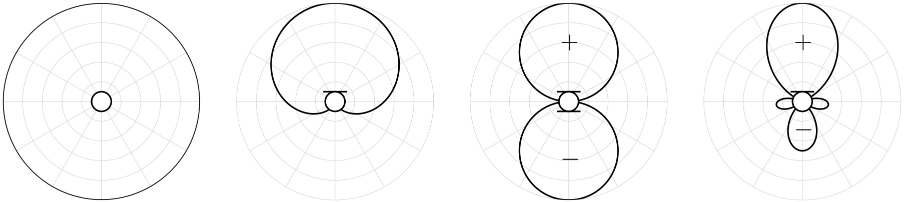


(a) Piezoelectric principle (© Wikipedia user: Tizeff CC BY-SA. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/)

(b) Hydrophone (© DolphinEAR Hydrophones. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/)

Figure: Piezo microphones

Microphone polar patterns



(a) Omni (b) Cardioid (c) Figure-eight (d) Shotgun

Figure: Microphone directivity patterns

Proximity effect

Directional microphones (all but omnis) record sound sources at close distances $d \leq \lambda$ with a boost of low frequencies.

Microphone polar patterns

(a) Earthworks TC20 omni

(b) Shure SM58 cardioid

(c) Royer R-101 figure-of-eight

Figure: Examples of omni, cardioid, and figure-of-8 microphones ()

Stereo recording techniques

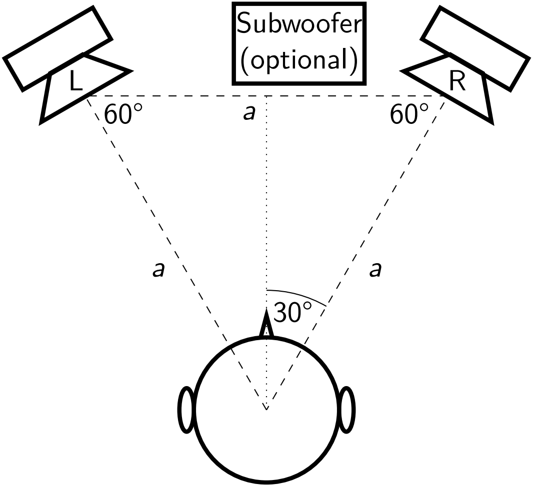


Figure: Standard stereo loudspeaker setup

Stereo recording techniques

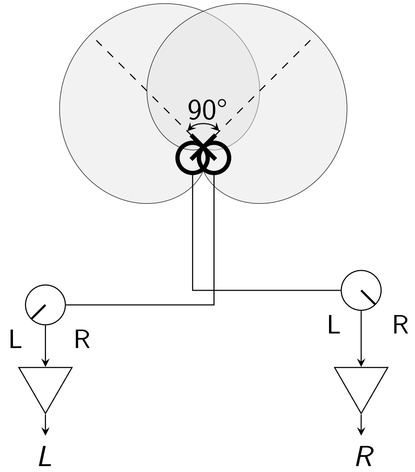


Figure: XY stereo recording technique

Stereo recording techniques

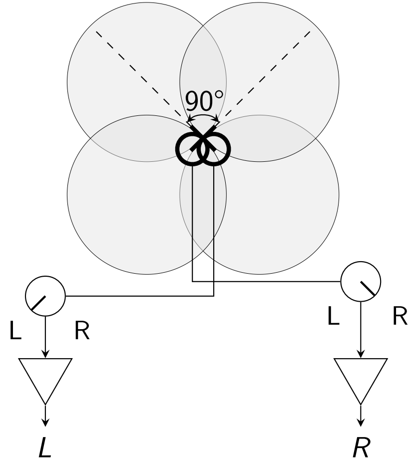


Figure: Blumlein pair

Stereo recording techniques

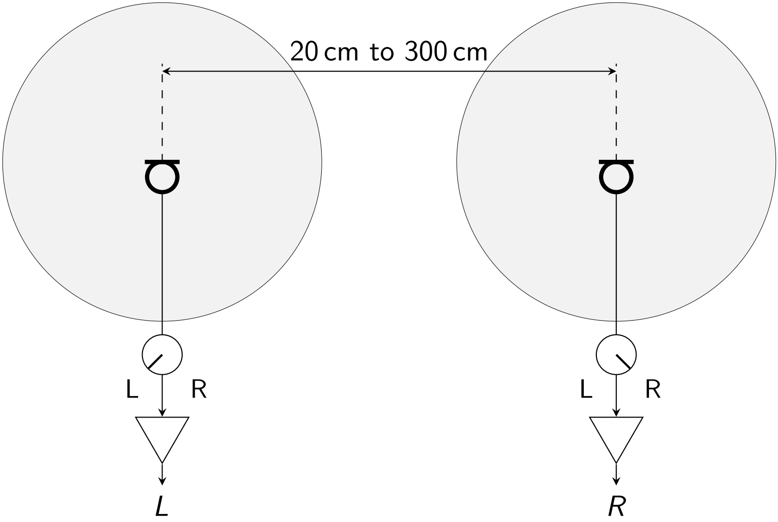


Figure: AB stereo recording technique

Stereo recording techniques

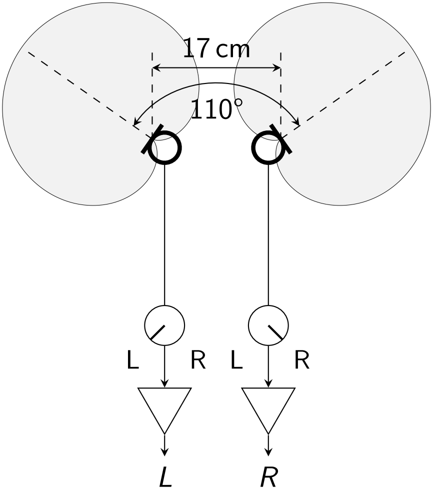


Figure: ORTF stereo recording technique

Zoom H4n operation

Figure: Zoom H4n portable audio recorder ()

Level setting and clipping

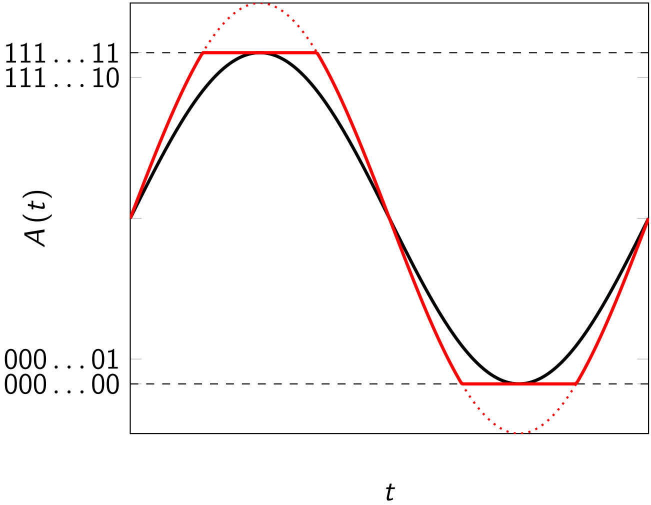


Figure: Unclipped (black) and clipped (red) digital signal

Clicks and crossfades

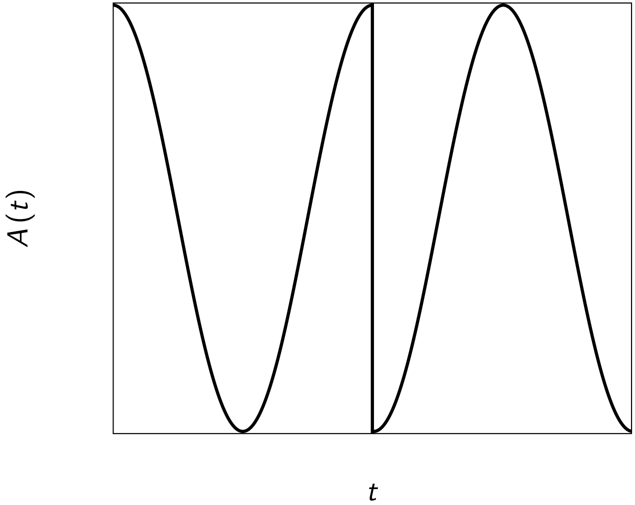
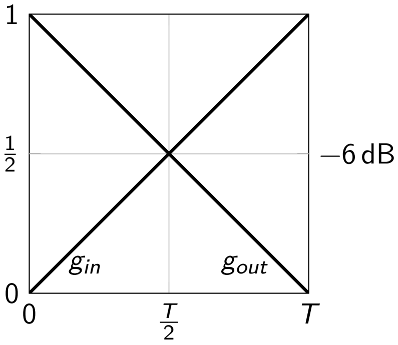
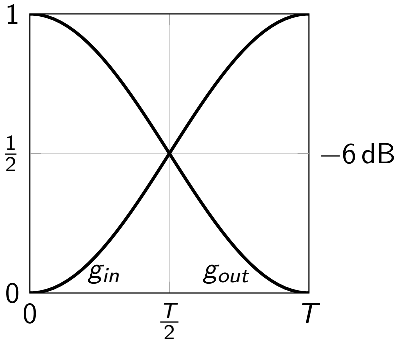


Figure: Signal discontinuity resulting in an audible click

Clicks and crossfades



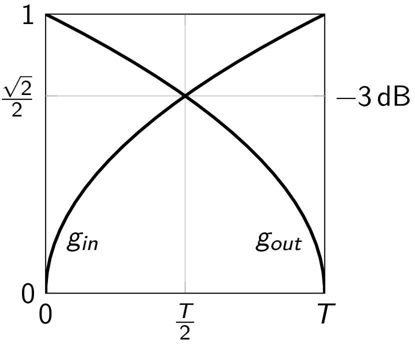
(a) Linear



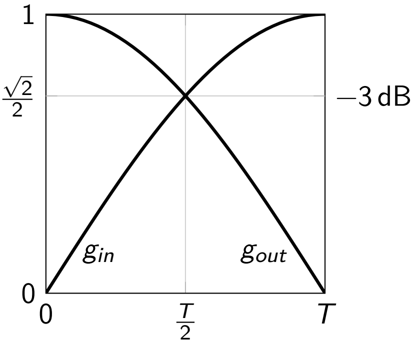
(b) Half cosine

Figure: Symmetrical crossfades at constant amplitude ($g_{in} + g_{out} = 1$) for crossfading correlated signals

Clicks and crossfades



(a) Square-root



(b) Quarter cosine

Figure: Symmetrical crossfades at constant power ($g_{in}^2 + g_{out}^2 = 1$) for crossfading uncorrelated signals

Filters and equalizers (EQs)

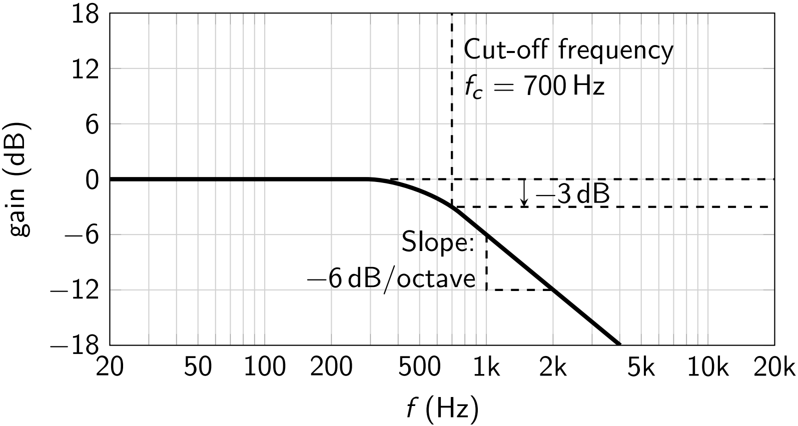


Figure: Frequency response of a low pass filter

Filters and equalizers (EQs)

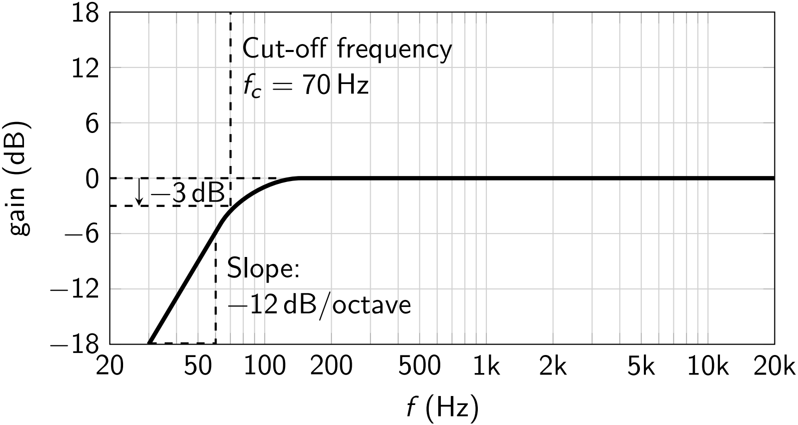


Figure: Frequency response of a high pass filter

Filters and equalizers (EQs)

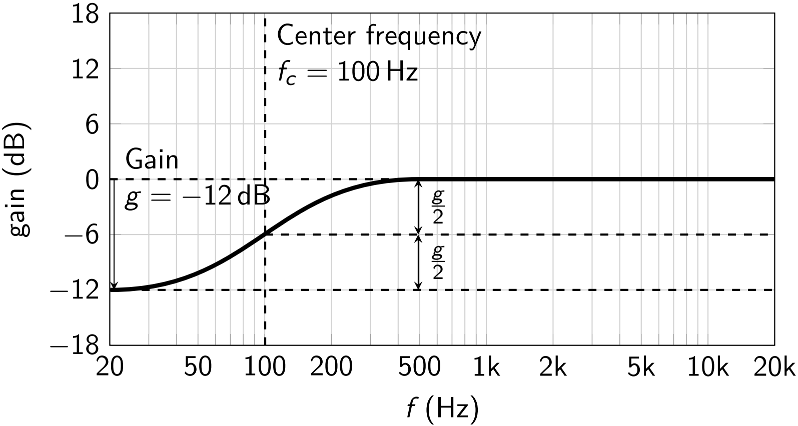


Figure: Frequency response of a low-frequency shelving filter

Filters and equalizers (EQs)

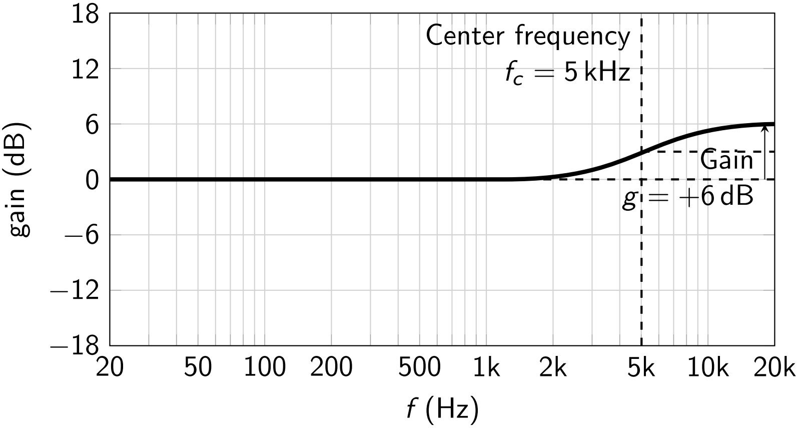


Figure: Frequency response of a high-frequency shelving filter

Filters and equalizers (EQs)

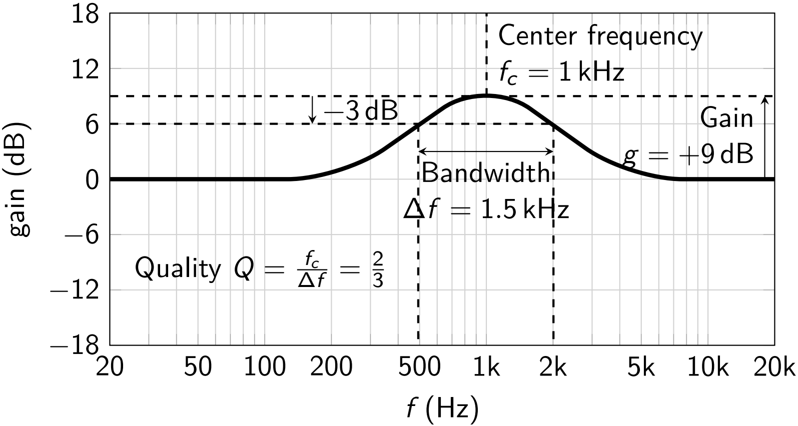


Figure: Frequency response of a peaking filter

Normalization

Method

1. Find the maximum (peak or RMS) level in the audio signal.
2. Amplify (or attenuate) the entire signal, such that the new maximum lies at a pre-defined target level.

Example

If the maximum peak level is $-7\,\mathrm{dB}$ and we want to normalize to $-3\,\mathrm{dB}$ peak level, the entire signal has to be amplified by $+4\,\mathrm{dB}$.

Properties

- Can be automated
- Inherently non-realtime
- Changes overall level (but neither spectrum nor dynamics)

DC offset removal

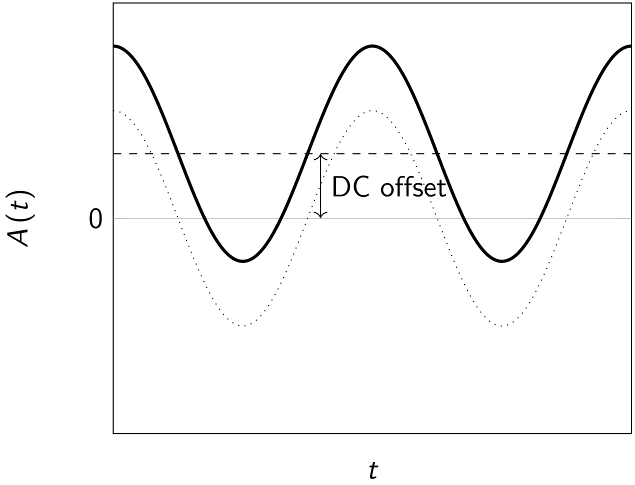


Figure: DC offset

21M.380 Music and Technology
Sound Design

Lecture 12: Quiz, review, preview

Massachusetts Institute of Technology
Music and Theater Arts

Monday, March 14, 2016

21M.380 Music and Technology
Sound Design

Lecture 13: Analysis and requirements specification

Massachusetts Institute of Technology
Music and Theater Arts

Wednesday, March 16, 2016

Analysis and requirements specification

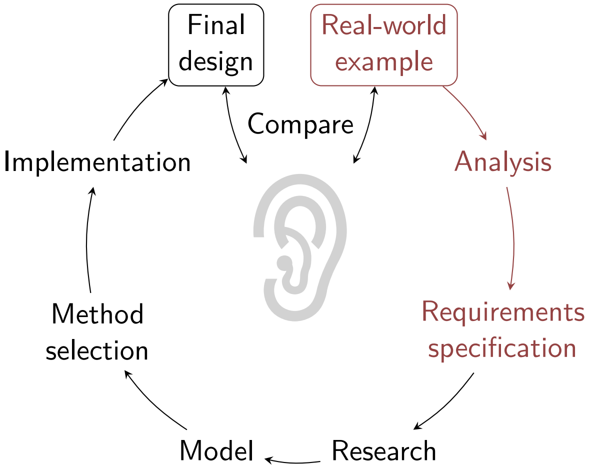


Figure: Stages of the sound design process (after Farnell 2010, figs. 16.7, 16.1)

Example: Steam train drive-by

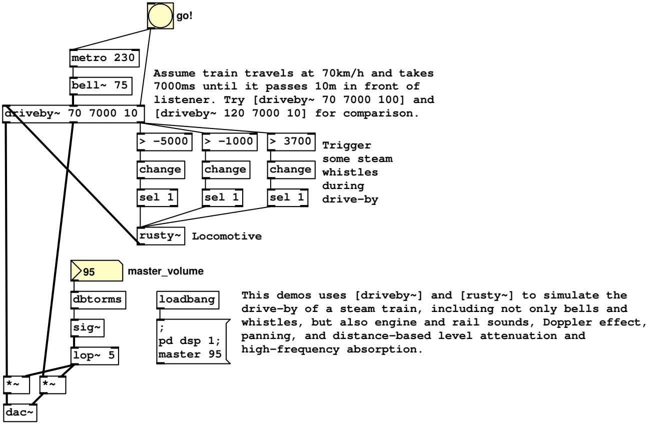


Figure: Steam train drive-by synthesized in Pd

Software packages

Software package	Mac	Windows	Linux	License
Sonic Visualiser	✔	✔	✔	open source
Baudline	✔		✔	free (as in beer)
Praat	✔	✔	✔	open source
VLC	✔	✔	✔	open source
Audacity	✔	✔	✔	open source
snd	✔		✔	open source

Table: Useful software packages for sound analysis

Converting audio and video files with VLC

Example: Remote video to local audio

- Download and install VLC: `http://www.videolan.org/`
- Open VLC
- Media › Convert / Save...
- Convert / Save › Profile: Audio - CD
  (don't use a lossy compressed format here)
- Destination file: `test.wav`
- Start

Spectrum

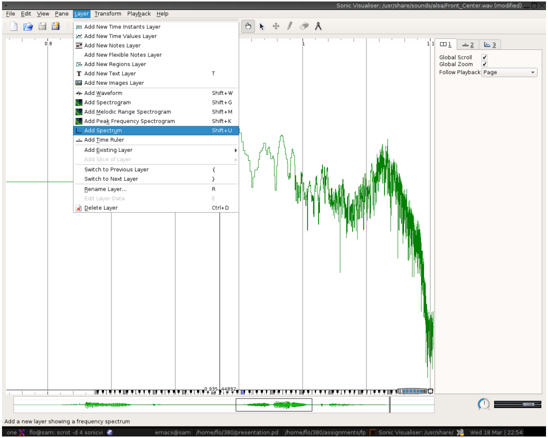


Figure: Spectrum in Sonic Visualiser: Layer › Add Spectrum

Spectrum

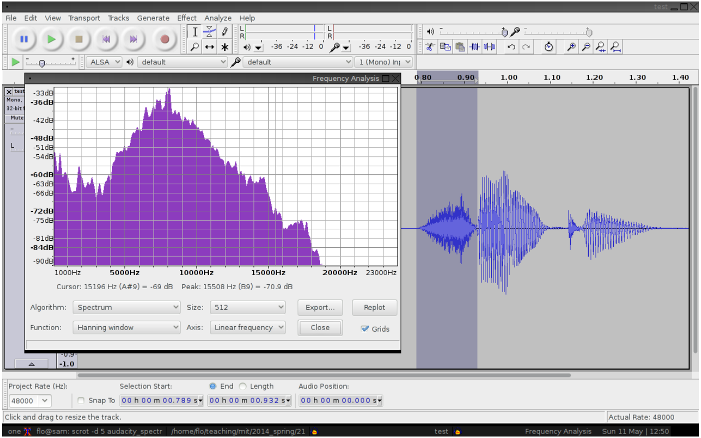


Figure: Spectrum in Audacity: Analyze > Plot Spectrum ...

Spectrogram

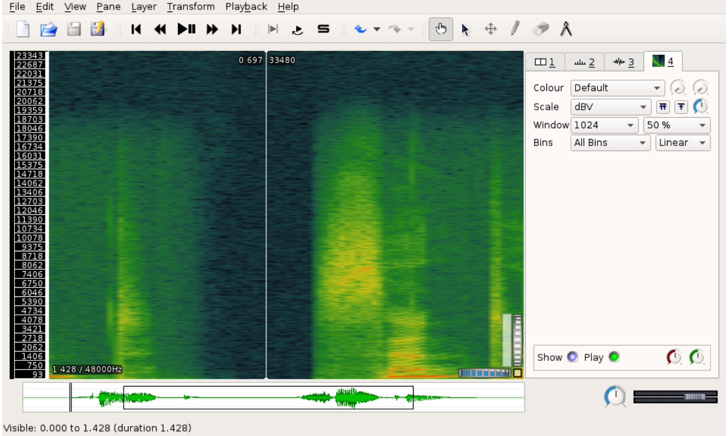


Figure: Spectrogram in Sonic Visualiser: Layer > Add Spectrogram

Spectrogram

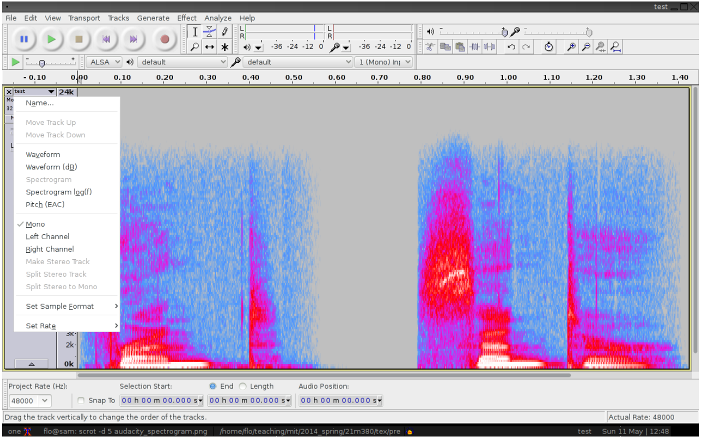


Figure: Spectrogram in Audacity: Track name › Spectrogram

Spectrogram

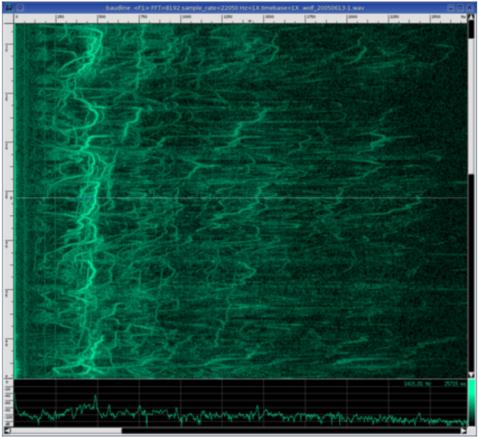


Figure: Realtime spectrogram in Baudline: `Right-click` Record (Courtesy of SigBlips. Used with permission)

Requirements specification

Sound sources

- Steam engine (‘chugga-chugga’)
- Steam whistle (‘choo-choo’)
- Rail joints (‘clackety-clack’)
- Railroad crossing warning bell (‘ding-ding-ding-...’)

Environmental acoustic effects

- Geometric attenuation (inverse distance law)
- High-frequency absorption over distance (low-pass filter)
- Doppler effect (pitch shift)
- Left-right movement (panning)

21M.380 Music and Technology
Sound Design

Lecture 14: Additive synthesis

Massachusetts Institute of Technology
Music and Theater Arts

Monday, March 28, 2016

Fourier series and theoretical limitations

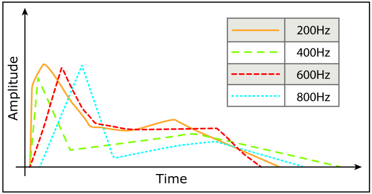
Fourier series (Farnell 2010, eq. 17.1)

$$f(\theta) = \frac{1}{2}a_0 + \sum_{k=0}^{\infty} a_k \cos(k\theta) + b_k \sin(k\theta)$$

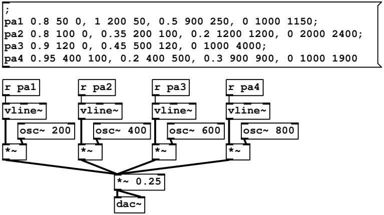
Limitation	Solution
Static spectra only	Amplitude envelopes
Harmonic spectra only	Frequency envelopes
Lots of control data	Discrete summation

Table: Theoretical limitations of the Fourier series with regards to additive synthesis

Generalization to dynamics spectra



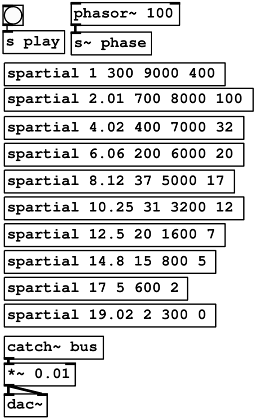
(a) Breakpoint envelopes for differen frequencies (Image by MIT OpenCourseWare, after Farnell 2010, fig. 17.1.)



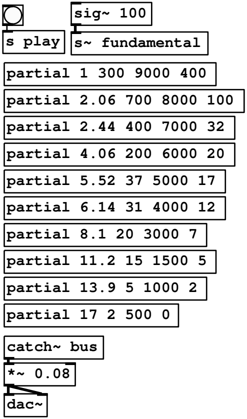
(b) Example in Pd (Farnell 2010, fig. 17.1)

Figure: Breakpoint envelopes in additive synthesis

Generalization to dynamics spectra



(a) Partials with synchronized phases



(b) Asynchronous phases

Figure: Different approaches to phase in additive synthesis (Farnell 2010, fig. 17.2)

Generalization to inharmonic spectra

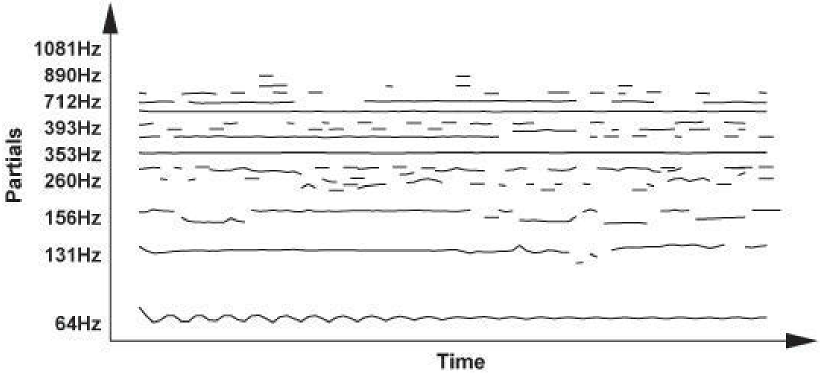


Figure: Partial tracing in Pd: Help ⟩ Browser... ⟩ Pure Data ⟩: 4.data.structures ▸ 14.partialtracer.pd (Farnell 2010, fig. 17.3.)

Control data reduction

Discrete summation formula (Farnell 2010, eq. 17.2)

$$\sum_{k=0}^{N} a^k \sin(\theta + k\beta) = \frac{\sin\theta - a\sin(\theta - \beta)}{1 + a^2 - 2a\cos\beta}$$

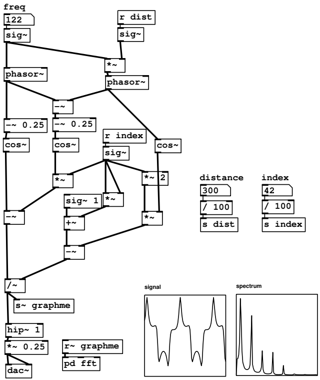
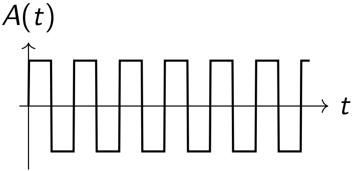
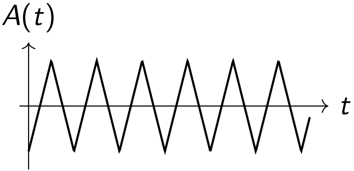


Figure: Discrete summation form of additive synthesis (Farnell 2010, fig. 17.4)

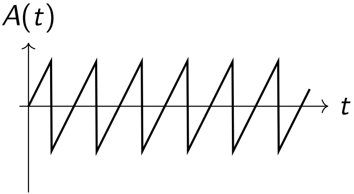
Bandlimited oscillators



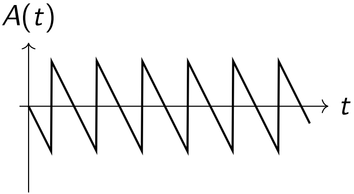
(a) Square wave



(b) Triangle wave



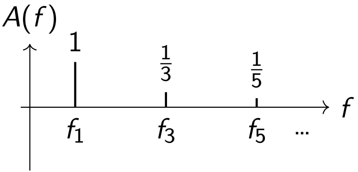
(c) Sawtooth wave



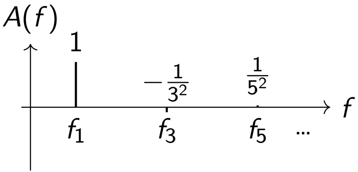
(d) Inverse sawtooth

Figure: Waveform archetypes

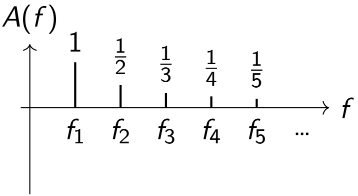
Bandlimited oscillators



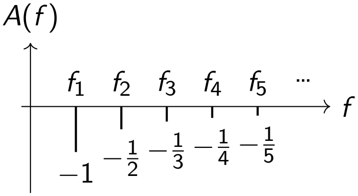
(a) Square wave



(b) Triangle wave



(c) Sawtooth wave



(d) Inverse sawtooth

Figure: Spectra of waveform archetypes

Bandlimited oscillators

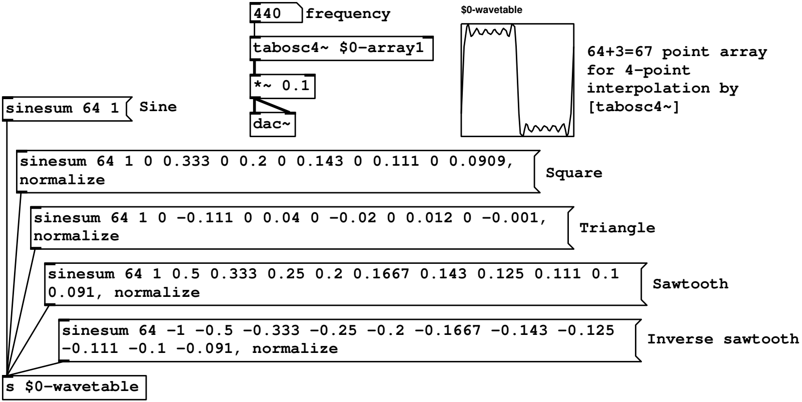


Figure: Creating archetypal waveforms as wavetable oscillators

Bandlimited oscillators

Band-limited pulse

$$\frac{a}{n}\sum_{k=1}^{n}\cos\left(k\theta\right) =$$

$$\frac{a}{2n}\left\{\frac{\sin\left(\frac{(2n+1)\theta}{2}\right)}{\sin\left(\frac{\theta}{2}\right)} - 1\right\}$$

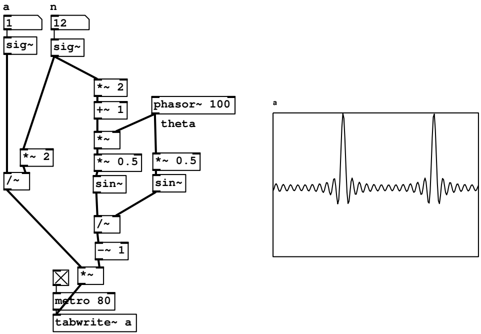
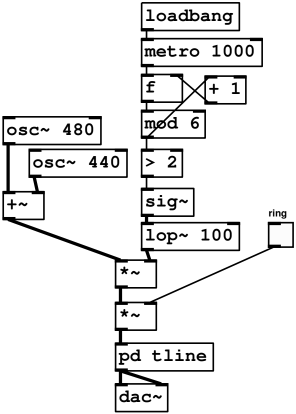
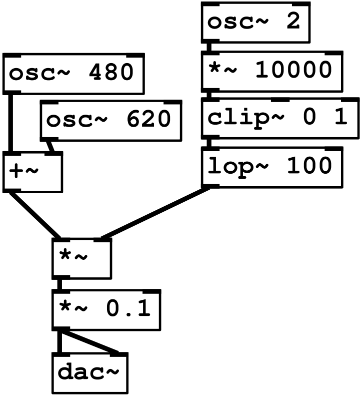


Figure: Closed form for band limited pulse (Farnell 2010, fig. 17.5)

Phone tones



(a) Ringing tone



(b) Busy tone

Figure: Additive synthesis to generate phone tones (Farnell 2010, fig. 25.5)

DTMF tones

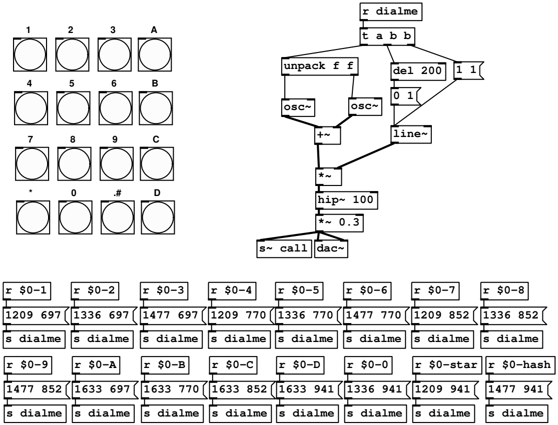


Figure: Additive synthesis to generate DTMF tones (Farnell 2010, fig. 26.3)

Alarm sounds

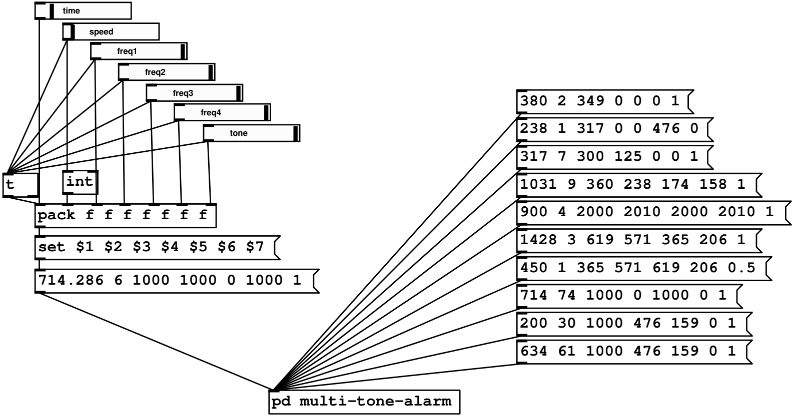


Figure: Additive synthesis to generate programmable alarm sounds (Farnell 2010, fig. 27.7)

Telephone bell

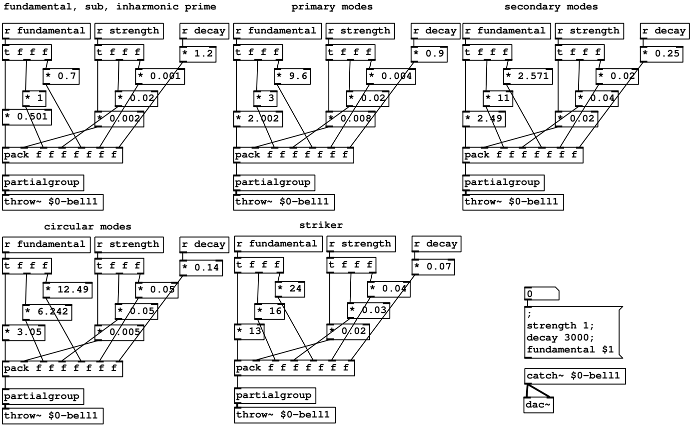


Figure: Additive synthesis to generate partials of a bell (Farnell 2010, fig. 29.13)

Telephone bell

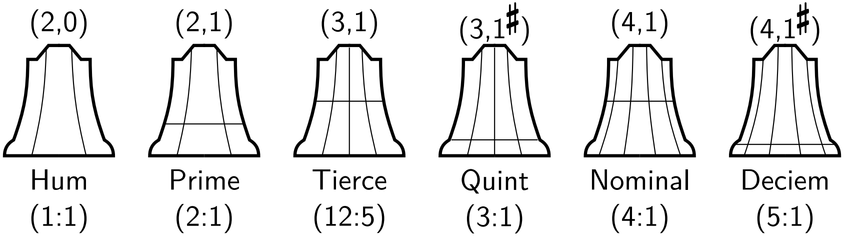


Figure: Characteristic modes of a bell (after Benson 2008, p. 130)

Jet engine turbine

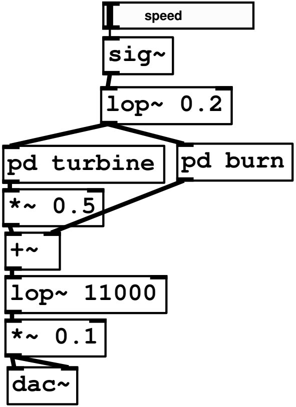


Figure: [pd turbine] uses additive synthesis to simulate the turbine of a jet engine (Farnell 2010, fig. 47.6)

21M.380 Music and Technology
Sound Design

Lecture 15: Research and model making

Massachusetts Institute of Technology
Music and Theater Arts

Wednesday, March 30, 2016

Research and model making

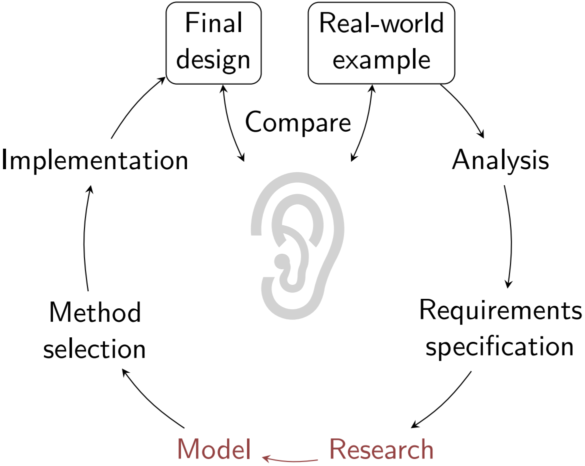


Figure: Stages of the sound design process (after Farnell 2010, figs. 16.7, 16.1)

Example: Steam train drive-by

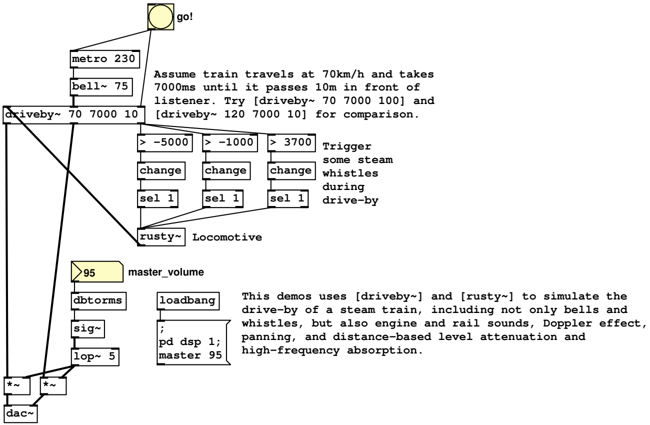


Figure: Steam train drive-by synthesized in Pd

Example: Steam train drive-by

Sound sources

- Steam engine ('chugga-chugga')
- Steam whistle ('choo-choo')
- Rail joints ('clackety-clack')
- Railroad crossing warning bell ('ding-ding-ding-...')

Environmental acoustic effects

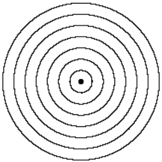
- Geometric attenuation (inverse distance law)
- High-frequency absorption over distance (low-pass filter)
- Doppler effect (pitch shift)
- Left-right movement (panning)

Example: Steam train drive-by

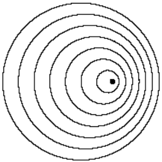
A	B	C	D
[redacted]	[redacted]	[redacted]	[redacted]
Steam whistle Doppler	Rail joints Distance attenuation	Steam engine HF absorption	Bell Panning

Table: Student groups

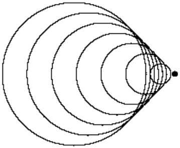
Doppler effect



(a) $v = 0$



(b) $v < c$



(c) $v > c$

Figure: Doppler shift ()

Doppler effect

Sound source moving towards stationary observer

$$f' = f \cdot \frac{c}{c - v}$$

Sound source moving away from stationary observer

$$f' = f \cdot \frac{c}{c + v}$$

- f' ... frequency according to observer (Hz)
- f ... frequency emitted by sound source (Hz)
- c ... speed of sound ($\mathrm{m\,s^{-1}}$)
- v ... velocity of sound source ($\mathrm{m\,s^{-1}}$)

Geometric attenuation over distance

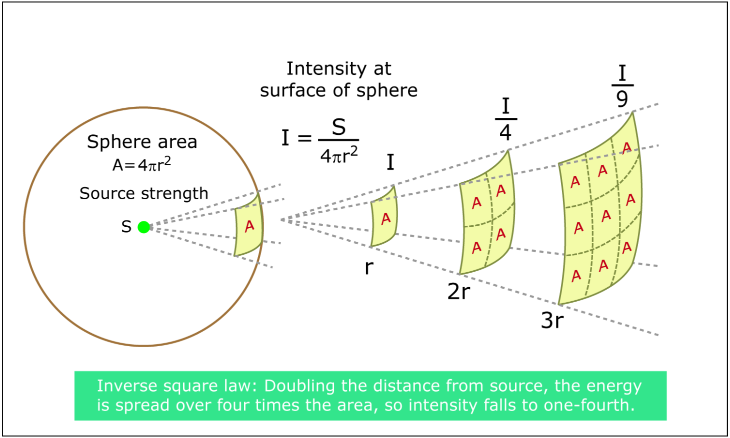


Figure: Inverse square law (Image by MIT OpenCourseWare, after Farnell 2010, fig. 5.4.)

Geometric attenuation over distance

Definition (Inverse square law)

The sound intensity I of a spherical wavefront in a free field decreases with the square of the distance r from the source.

$$I \propto \frac{1}{r^2} \tag{6}$$

Definition (Inverse distance law)

The sound pressure p of a spherical wavefront in a free field decreases with the distance r from the source.

$$p \propto \frac{1}{r} \tag{7}$$

High-frequency absorption over distance

Stokes' law of sound attentuation (Farnell 2010, eq. 5.9)

$$\alpha = \frac{2\eta\,(2\pi f)^2}{3\rho c^3}$$

- α ... attenuation ($\mathrm{Np\,m^{-1}} \approx \frac{1}{8.69}\mathrm{dB\,m^{-1}}$)
- η ... viscosity ($\mathrm{Pa\,s} = \mathrm{kg\,s^{-1}\,m^{-1}}$)
- f ... frequency (Hz)
- ρ ... density ($\mathrm{kg\,m^{-3}}$)
- c ... speed of sound ($\mathrm{m\,s^{-1}}$)

Example

12 km for 3 dB loss at 1 kHz

Details

ISO 9613-1/2, p. 5

Model

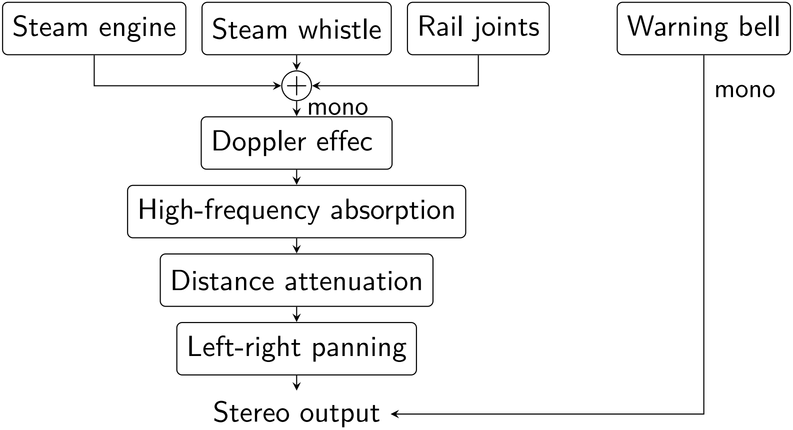


Figure: Block diagram of steam train drive-by model

Geometry

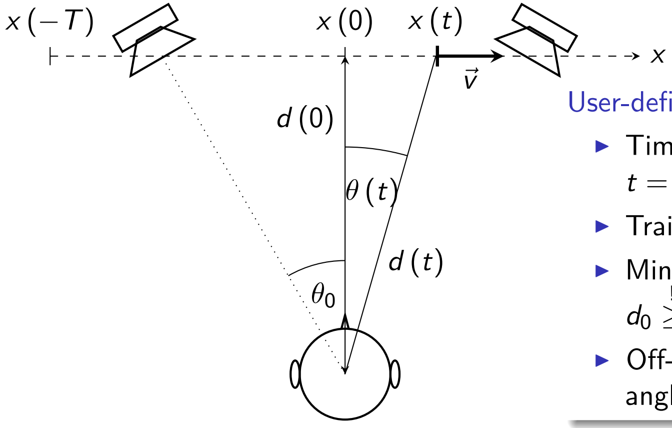


Figure: Geometry of steam train drive-by

User-definable parameters

- Time T from 'go' to $t = 0$
- Train velocity v
- Minimum distance $d_0 \overset{!}{\geq} 10\,\mathrm{m}$
- Off-center loudspeaker angle θ_0

Geometry

Parameters that need to be computed

- Current distance $d(t)$ between listener and train
- Current angle $\theta(t)$ between front-back axis and direction of train

Specification details

Executive decisions

- Single bang triggers entire drive-by
- Drive-by duration: $t = -T$ to $t = 4 \cdot T$, followed by fade out
- Train moves at constant velocity v from left to right
- Train moves on straight path perpendicular to listener's line of sight
- Engine and rail sounds should reflect train speed v
- Stereo panning within stereo base $\pm\theta_0$
- Hard left/right panning beyond $\pm\theta_0$
- Patch should output 0 dB fs for $d = 10\,\text{m}$
- Whistle should sound at three occasions: $t = \{-5\,\text{s}, -1\,\text{s}, +3.7\,\text{s}\}$

21M.380 Music and Technology
Sound Design

Lecture 16: Waveshaping and wavetable synthesis

Massachusetts Institute of Technology
Music and Theater Arts

Monday, April 4, 2016

Waveshaping synthesis

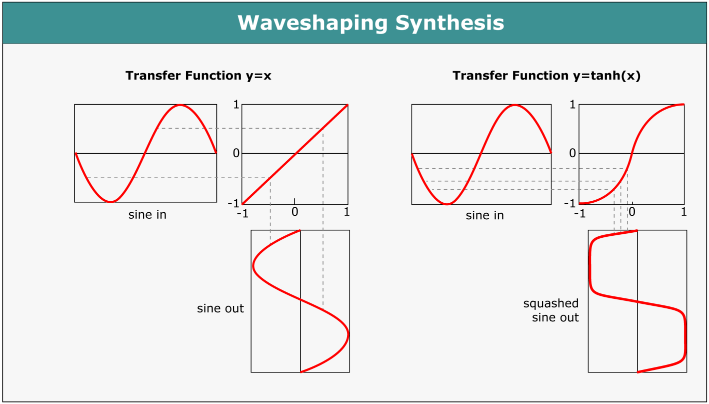


Figure: Waveshaping with identity and tanh transfer functions (Image by MIT OpenCourseWare, after Farnell 2010, fig. 19.1.)

Waveshaping synthesis

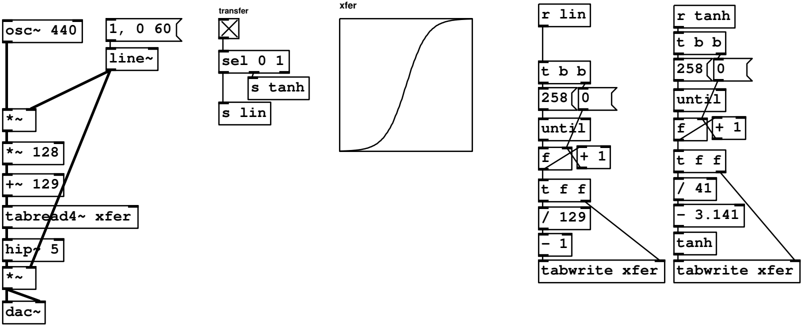


Figure: A table-based waveshaper noise (Farnell 2010, fig. 19.2)

Waveshaping synthesis

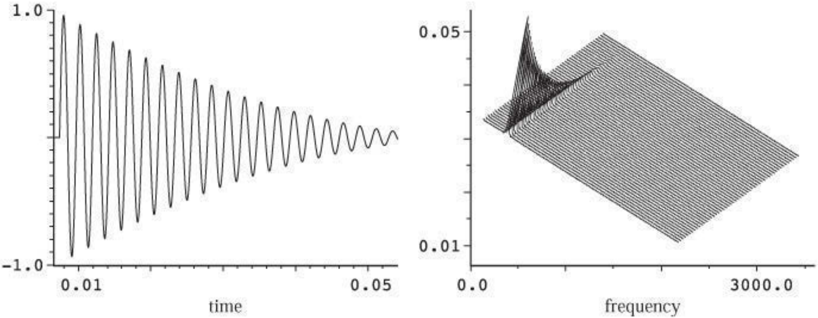


Figure: A linear transfer function has no effect (Farnell 2010, fig. 19.3.)

Waveshaping synthesis

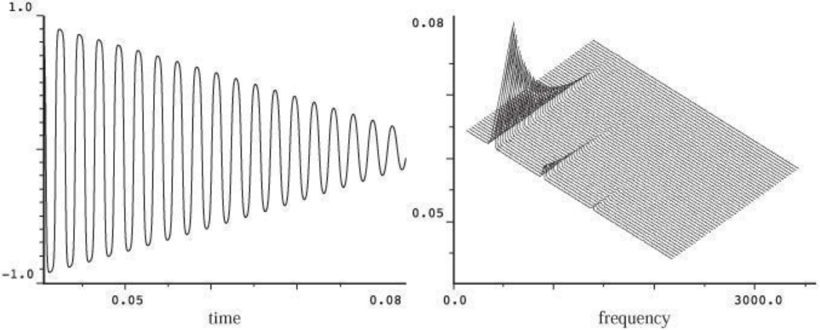


Figure: A tanh transfer function makes more harmonics when the input is louder (Farnell 2010, fig. 19.4.)

Waveshaping synthesis

Properties (Farnell 2010, p. 283)

- Nonlinear
- Deliberate harmonic distortion
- Provides more harmonics than we put in
- Number of harmonics depends on input amplitude

Method (Farnell 2010, p. 283)

- Find the transfer function that generates the desired spectrum for a given input signal.

Waveshaping synthesis

Advantages (Farnell 2010, pp. 257, 284)

- Simple creation of *dynamic* spectra (due to non-linearity)
- Resynthesize instruments with lots of spectral flux (brass, strings)
- Resembles nature (louder sounds often have more harmonics)

Disadvantages

- Bandwidth of resulting spectrum not limited (aliasing!)
- Hard to predict resulting spectra in detail (due to non-linearity)

Applications

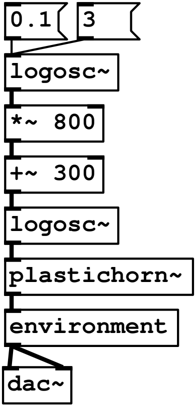


Figure: [plastichorn] models the plastic horn of a police siren via waveshaping (Farnell 2010, fig. 28.9)

Applications

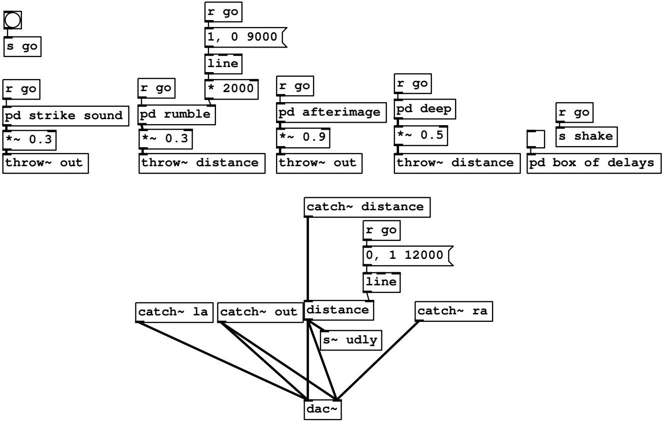


Figure: Waveshaping is used in the [pd rumble] and [pd deep] subpatches of this thunder patch (Farnell 2010, fig. 40.13)

Applications

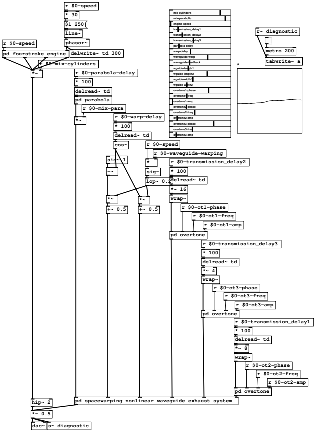


Figure: Car engine patch using $\frac{1}{1+kx^2}$ waveshaping function in [pd fourstroke engine] and $\frac{-4d^2+1}{2}$ function in [pd overtone] (Farnell 2010, fig. 45.8)

Applications

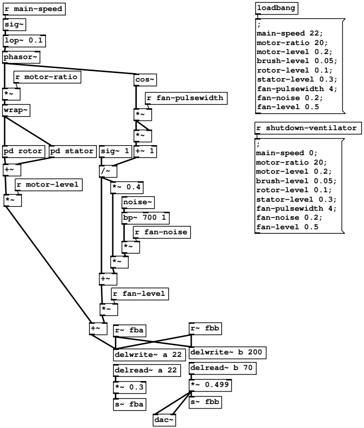


Figure: This patch simulates a ventilation system and includes a $\frac{1}{1+x^2}$ waveshaping function towards the top right to simulate the pulse from a fan (Farnell 2010, fig. 46.6)

Applications

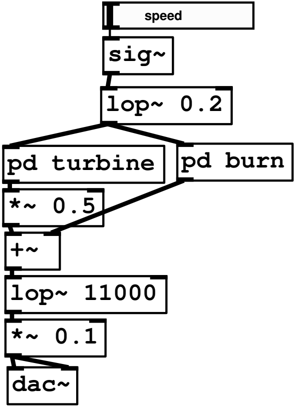


Figure: This jet engine patch uses waveshaping to simulate a forced flame in [pd burn] (Farnell 2010, fig. 47.6)

Applications

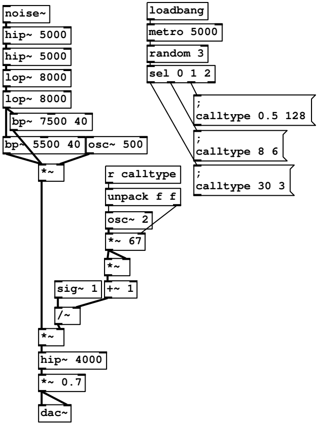


Figure: $\frac{1}{1+x^2}$ waveshaping in the simulation of a cicada call (Farnell 2010, fig. 50.10)

Applications

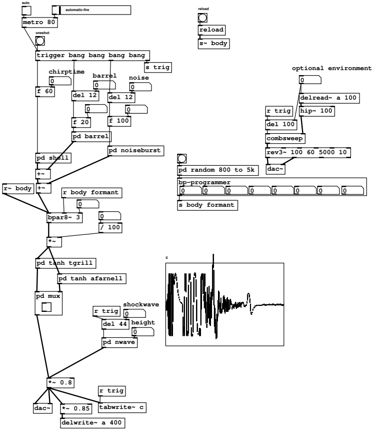
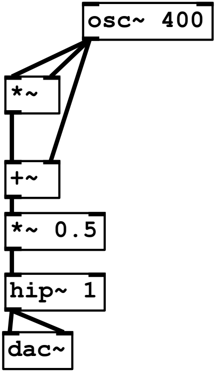
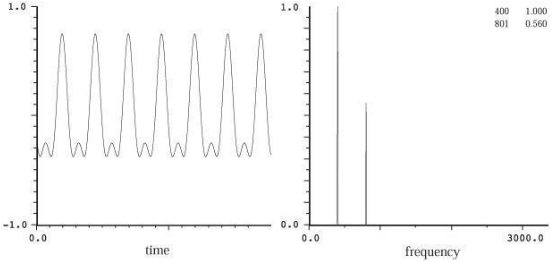


Figure: A $\tanh(x)$ waveshaping function is used in this patch to simulate the sound of gunfire (Farnell 2010, fig. 53.9)

Chebyshev Polynomials



(a) Pd patch (Farnell 2010, fig. 19.6)



(b) Resulting waveform and spectrum (Farnell 2010, fig. 19.7.)

Figure: $x^2 + x$

Chebyshev Polynomials

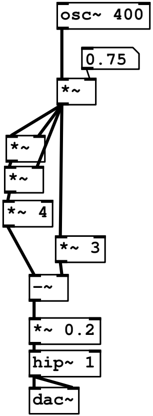
Recursive definition (Farnell 2010, fig. 19.5; Burk et al. 2011, sec. 4.6)

$$
\begin{aligned}
T_0(x) &= 1 \\
T_1(x) &= x \\
T_2(x) &= 2x^2 - 1 \\
T_3(x) &= 4x^3 - 3x \\
T_4(x) &= 8x^4 - 8x^2 + 1 \\
T_5(x) &= 16x^5 - 20x^3 + 5x \\
&\cdots \\
T_{n+1}(x) &= 2x \cdot T_n(x) - T_{n-1}(x)
\end{aligned}
$$

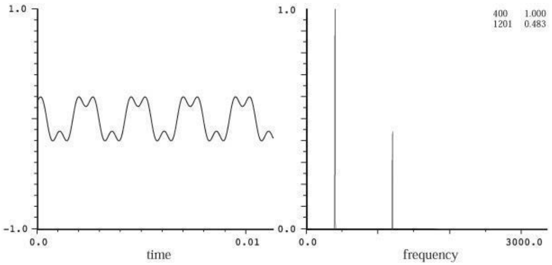
Properties

- Generate specific harmonics ☺ if input x is sinusoidal
- $T_{\{1,3,5,\dots\}}$ … odd harmonics
- $T_{\{2,4,6,\dots\}}$ … even harmonics
- T_n generates only f_n ☺ for full-range input amplitude (Burk et al. 2011, sec. 4.6)
- For lower amplitudes, generates mixture of odd or even harmonics up to n

Chebyshev Polynomials



(a) Pd patch (Farnell 2010, fig. 19.8)



(b) Resulting waveform and spectrum (Farnell 2010, fig. 19.10.)

Figure: $T_3 = 4x^3 - 3x$ (with scaling and DC offset removal)

Chebyshev Polynomials

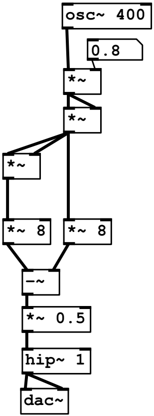


Figure: $8x^4 - 8x^2$ (Farnell 2010, fig. 19.9)

Demo

- Turn on DSP
- Hold ⇧ and drag number box from 0.0 to 1.0
- Audible effect? Crossfade from f_2 to f_4

Chebyshev Polynomials

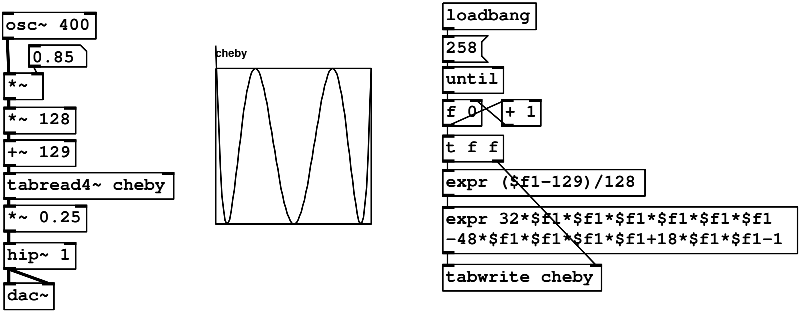


Figure: Higher-order polynomials are better implemented using tables (Farnell 2010, fig. 19.11)

Wavetable synthesis

Figure: PPG Wave 2.2 synthesizer from 1982 (© John R. Southern and Wikipedia user: Shoulder-synth.

Tables in Pd

Pd object	Function	Application
[tabread]	Read table at message rate	Control data
[tabread4]	Same w/ 4-point interpolation	
[tabwrite]	Write to table at message rate	
[tabread~]	Read table at audio rate	Samplers and wavetable oscillators
[tabread4~]	Same w/ 4-point interpolation	
[tabwrite~]	Write signal to table	
[tabplay~]	Play table as audio sample	
[tabosc4~]	Play table as waveform period	
[tabsend~]	Keep writing block to table	FFT (Hann window)
[tabreceive~]	Keep reading block from table	

Table: Table objects in Pd

Tables in Pd

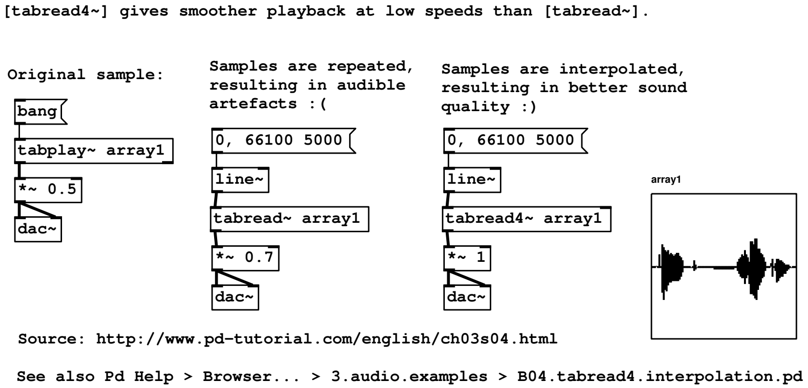


Figure: Benefits of interpolation when reading from wavetables

Tables in Pd

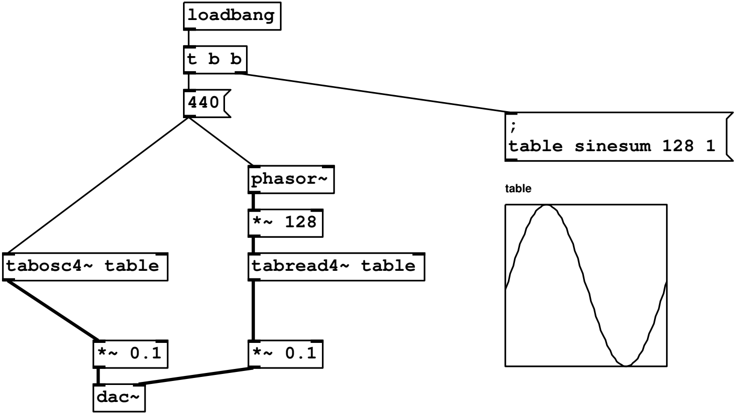


Figure: Two equivalent wavetable oscillators in Pd

Equivalence of waveshaping and wavetable synthesis

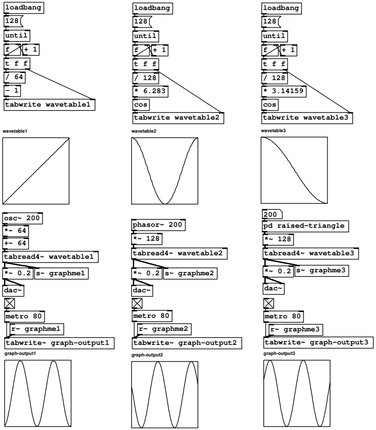


Figure: Equivalence of waveshaping (left) and wavetable synthesis (center). Right: Exploiting waveform symmetry in wavetable synthesis. (Farnell 2010, fig. 18.1)

Basic wavetable synthesis

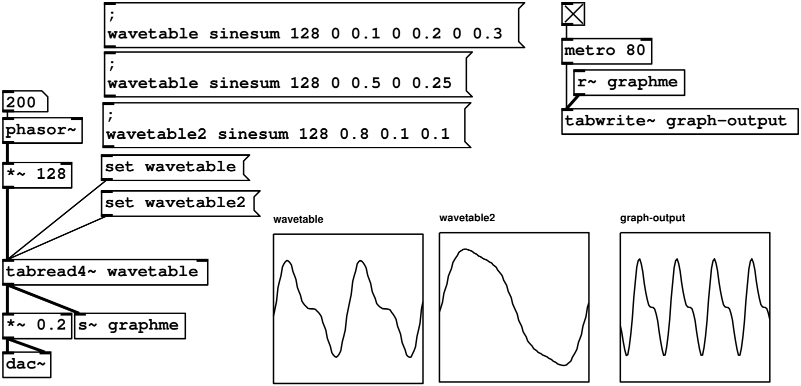


Figure: Using wavetables in Pure Data (Farnell 2010, fig. 18.2)

Basic wavetable synthesis

Limitation	Solutions
Static spectra only	Change table contents on the fly, or: Change between different tables
How to change table (contents) without clicks?	Write behind phasor index, or: Crossfade between tables

Table: Limitations of wavetable synthesis and possible solutions (Farnell 2010, p. 279)

Vector synthesis

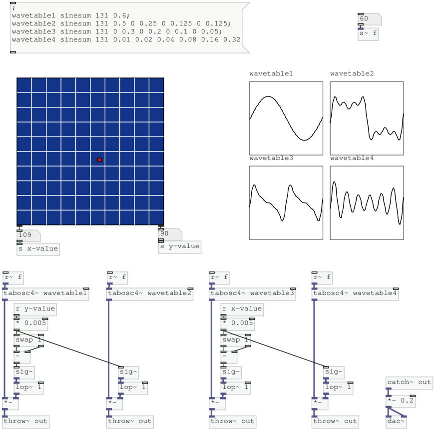


Figure: 2D vector synthesizer using the [grid] GUI object from Pd extended (Farnell 2010, fig. 18.3)

Wavescanning synthesis

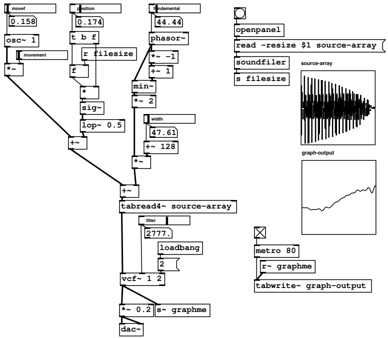


Figure: Wavescanning synthesizer (Farnell 2010, fig. 18.4)

Demo

1. Load soundfile via [openpanel]
2. Turn on DSP and toggle graph
3. Set `filter` slider > 0
4. Move position slider rapidly for ‘turntable scratch’ effect
5. Leave position in non-silent area
6. Increase `fundamental` until you hear a static sound
7. Adjust `width`, `move`, `movef` to change spectrum

21M.380 Music and Technology
Sound Design

Lecture 17: Student presentations

Massachusetts Institute of Technology
Music and Theater Arts

Wednesday, April 6, 2016

21M.380 Music and Technology
Sound Design

Lecture 18: Modulation synthesis (AM and FM)

Massachusetts Institute of Technology
Music and Theater Arts

Monday, April 11, 2016

Modulation synthesis (AM and FM)

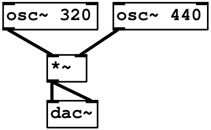
Advantages (Farnell 2010, p. 257)

- Generate complex spectra with little control data
- Computationally cheap (as few as two oscillators)
- Good for generating inharmonic spectra (brass, bells, metal)
- Simple control of spectrum's (in)harmonicity

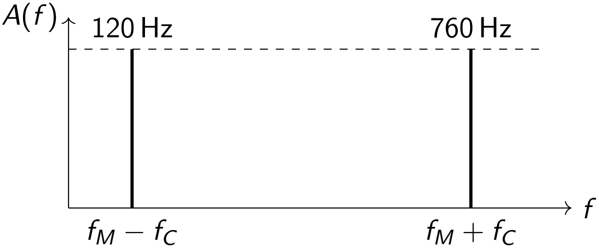
Amplitude modulation

Cosine product-to-sum rule (Farnell 2010, eq. 20.2)

$$\cos(a)\cos(b) = \frac{1}{2}\cos(a+b) + \frac{1}{2}\cos(a-b)$$



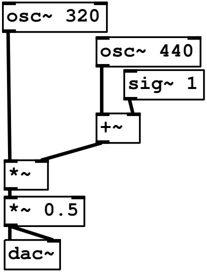
(a) $f_C = 320\,\text{Hz}$, $f_M = 440\,\text{Hz}$ (Farnell 2010, fig. 20.1)



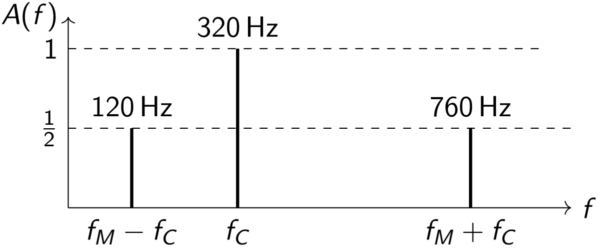
(b) Resulting spectrum (after Farnell 2010, fig. 20.2)

Figure: Multiplying two sinusoids

Ring modulation



(a) $f_C = 320\,\text{Hz}$, $f_M = 440\,\text{Hz}$ (Farnell 2010, fig. 20.3)



(b) Resulting spectrum (after Farnell 2010, fig. 20.4)

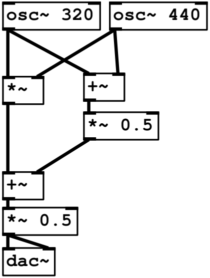
Figure: This ring-modulator includes the original carrier frequency.

Ring modulation

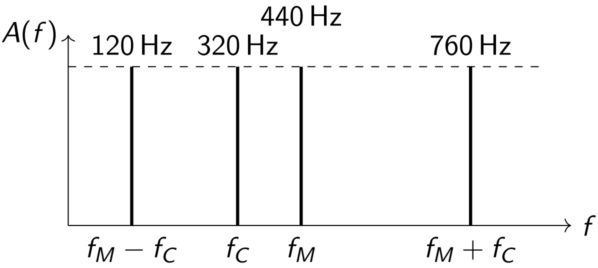
Image removed due to copyright restrictions. Photo of 'Mantra' performance arrangement: two adjacent grand pianos with lids off and several microphones; each piano supplemented by crotales and a small ring modulator.

Figure: Karlheinz Stockhausen's composition 'Mantra' (1970) features two pianos whose sound is ring-modulated by oscillators

All-band modulation



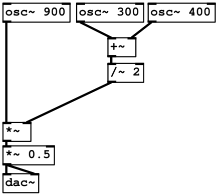
(a) $f_C = 320$ Hz , $f_M = 440$ Hz (Farnell 2010, fig. 20.5)



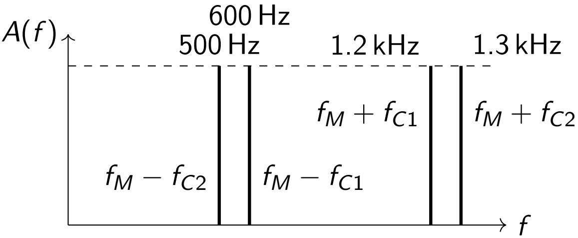
(b) Resulting spectrum (after Farnell 2010, fig. 20.6)

Figure: An all-band modulator includes both original frequencies.

Modulating a complex carrier signal



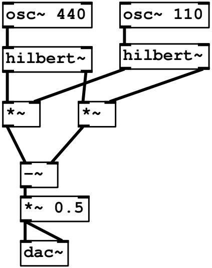
(a) $f_{C1} = 300$ Hz , $f_{C2} = 400$ Hz , $f_M = 900$ Hz (Farnell 2010, fig. 20.7)



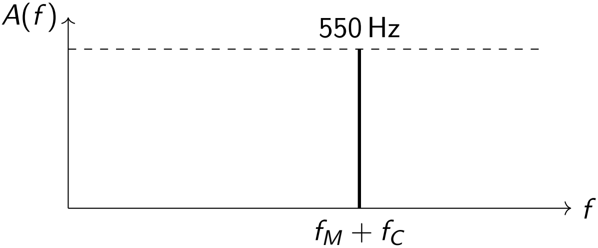
(b) Resulting spectrum (after Farnell 2010, fig. 20.8)

Figure: Modulating a carrier with two frequency components

Single-sideband modulation



(a) $f_C = 440\,\text{Hz}$, $f_M = 110\,\text{Hz}$ (Farnell 2010, fig. 20.9)



(b) Resulting spectrum (after Farnell 2010, fig. 20.10)

Figure: Using a Hilbert transform to obtain only the upper sideband

Applications

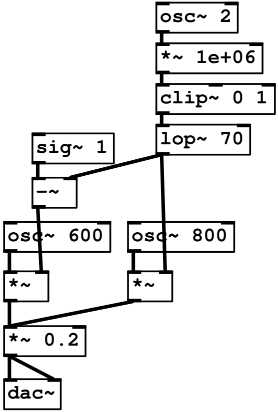


Figure: Amplitude modulation in an alarm sound (Farnell 2010, fig. 27.2)

Applications

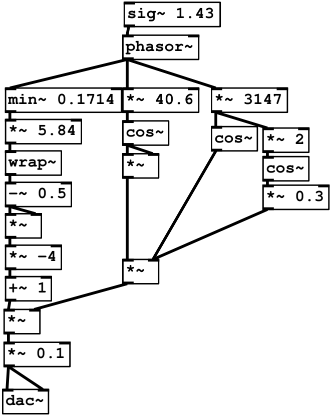


Figure: Amplitude modulation in a field cricket patch (Farnell 2010, fig. 50.7)

Frequency modulation

Figure: Yamaha DX7 (Public domain image. Source: https://commons.wikimedia.org/wiki/File:YAMAHA_DX7.jpg)

Frequency modulation

Image removed due to copyright restrictions. Photo of John Chowning.

Figure: John Chowning (⋆ 1934), father of FM synthesis (Chowning 1973) and composer of 'Stria' (1977)

Basic principles

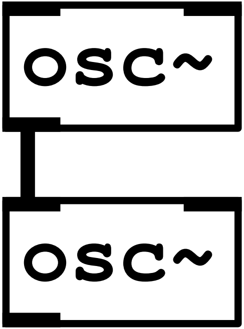
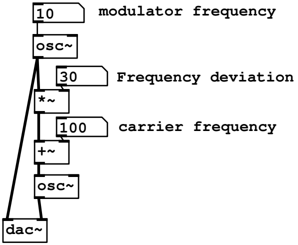
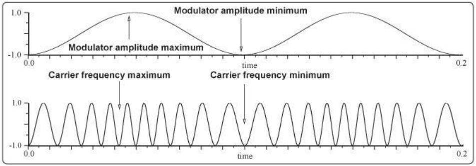


Figure: FM (Farnell 2010, fig. 20.11)

Basic principles



(a) Real FM patch (Farnell 2010, fig. 20.12)



(b) FM with a carrier of 100 Hz, modulator of 10 Hz, and an index of 30 (Farnell 2010, fig. 20.13.)

Figure: Frequency modulation with $f_C = 100\,\text{Hz}$, $f_M = 10\,\text{Hz}$ and $\Delta f = 30\,\text{Hz}$

Basic principles

Modulation index

$$i = \frac{\Delta f}{f_M}$$

Carson's rule

$$B = 2\,(\Delta f + f_M)$$

- i … modulation index
- Δf … freq. deviation (Hz)
- f_M … modulator freq. (Hz)
- B … bandwidth (Hz)

Demo

1. f_C shifts spectrum on frequency axis
2. f_M controls (in)harmonicity (distance between partials)
3. Δf controls bandwidth (number of sidebands)

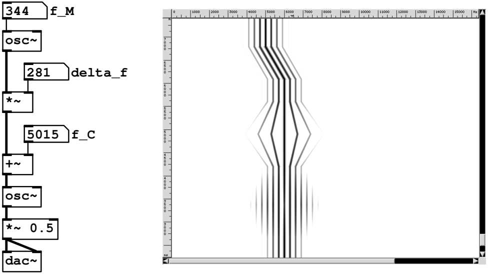


Figure: FM demo with Pd and Baudline

Theoretical foundation

Cosine sum-to-product rule (Farnell 2010, eq. 20.7)

$$\cos(a+b) = \cos(a)\cos(b) - \sin(a)\sin(b)$$

FM formula (Farnell 2010, eqs. 20.10 ff.

$$\begin{aligned}
\cos(\omega_c t + i \sin \omega_m t) \quad = \quad & J_0(i)\cos(\omega_c t) \\
& -J_1(i)\cos((\omega_c - \omega_m)t) - \cos((\omega_c + \omega_m)t) \\
& +J_2(i)\cos((\omega_c - 2\omega_m)t) + \cos((\omega_c + 2\omega_m)t) \\
& -J_3(i)\cos((\omega_c - 3\omega_m)t) - \cos((\omega_c + 3\omega_m)t) \\
& + \ldots
\end{aligned}$$

- i … modulation index ($\neq$ imaginary unit)
- J_n … Bessel functions of the first kind

Theoretical foundation

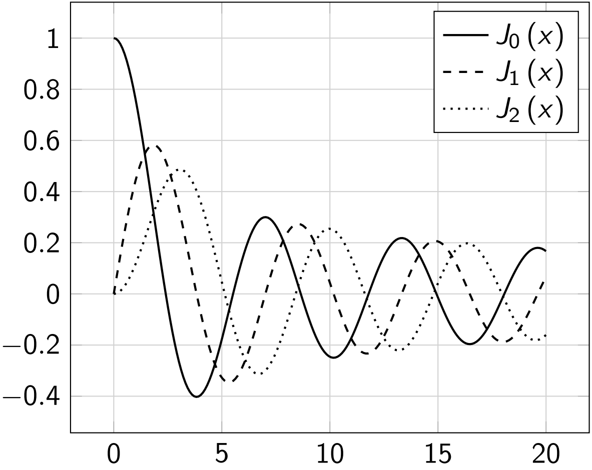
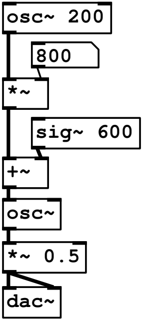


Figure: First three Bessel functions of the first kind

Negative frequencies



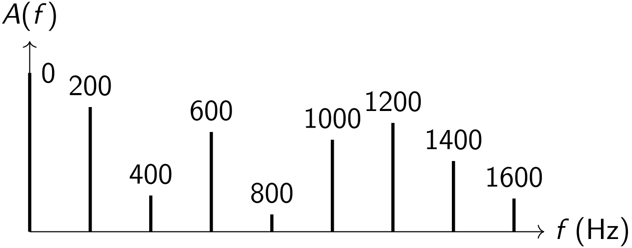
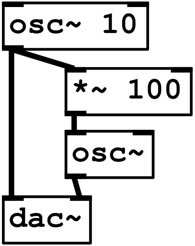
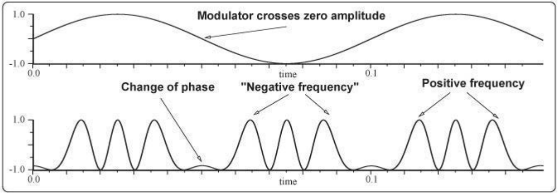


Figure: Negative frequencies being 'folded up' into the positive spectrum (Farnell 2010, fig. 20.18)

Negative frequencies



(a) Basic FM patch (Farnell 2010, fig. 20.19)



(b) Negative frequencies cause a change of phase (Farnell 2010, fig. 20.20.)

Figure: Relationship between phase changes and negative frequencies

Phase modulation

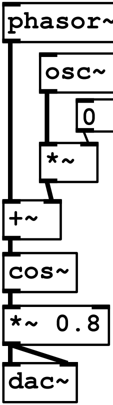


Figure: Phase modulation is equivalent to FM but makes the phase available separately, so we can synchronize it with other processes. (Farnell 2010, fig. 20.21)

Applications

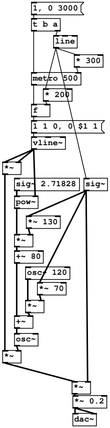


Figure: Frequency modulation in a patch simulating a bouncing ball (Farnell 2010, fig. 30.2)

Applications

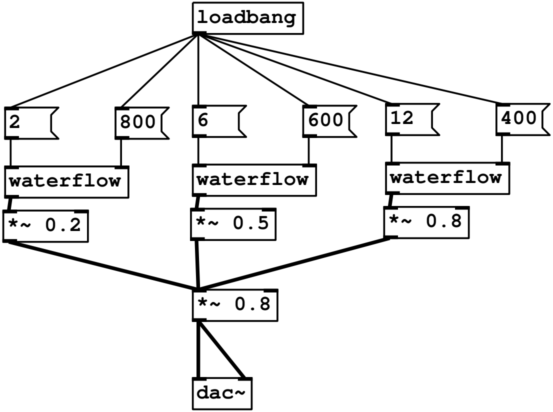


Figure: Amplitude and frequency modulation in a running water patch (cf., Farnell 2010, ch. 36)

Applications

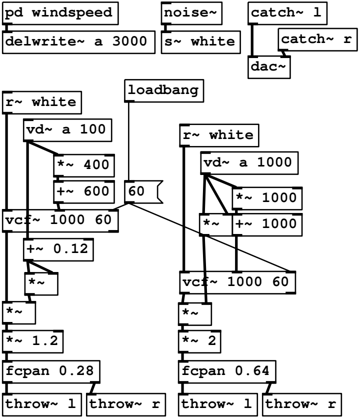


Figure: Simulation of whistling wires by means of amplitude and frequency modulation (Farnell 2010, fig. 41.12)

Applications

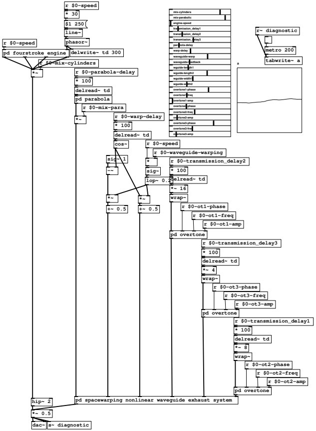


Figure: Frequency modulation in the [pd spacewarping nonlinear waveguide exhaust system] subpatch of a car engine simulation (Farnell 2010, fig. 45.8)

Applications

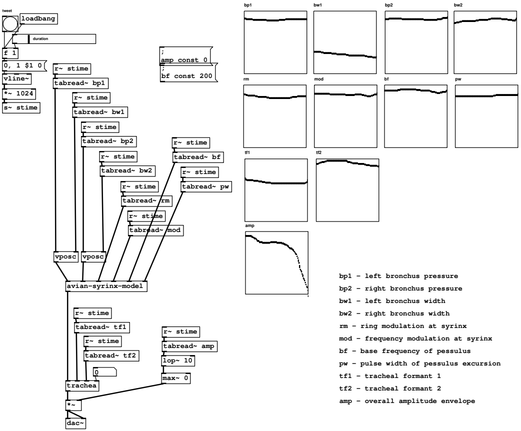


Figure: Amplitude and frequency modulation in the [avian-syrinx-model] of a bird call patch (Farnell 2010, fig. 51.7)

Applications

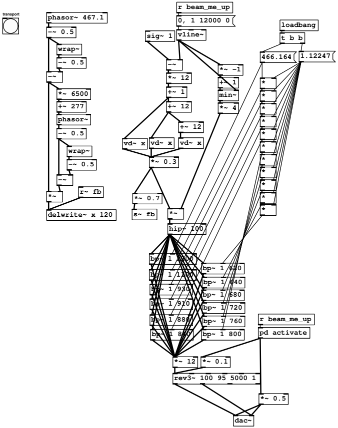


Figure: This Star Trek transporter sound simulates a cymbal by FM synthesis of triangle waves (Farnell 2010, fig. 56.2)

Applications

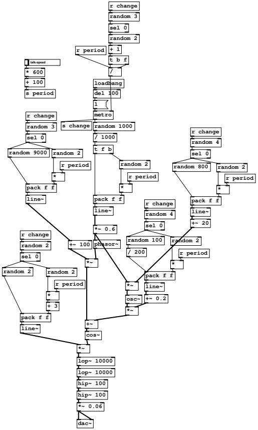


Figure: Who knew? R2D2 speaks phase modulation! (Farnell 2010, fig. 57.3)

21M.380 Music and Technology
Sound Design

Lecture 19: Method selection and implementation

Massachusetts Institute of Technology
Music and Theater Arts

Wednesday, April 13, 2016

Method selection and implementation

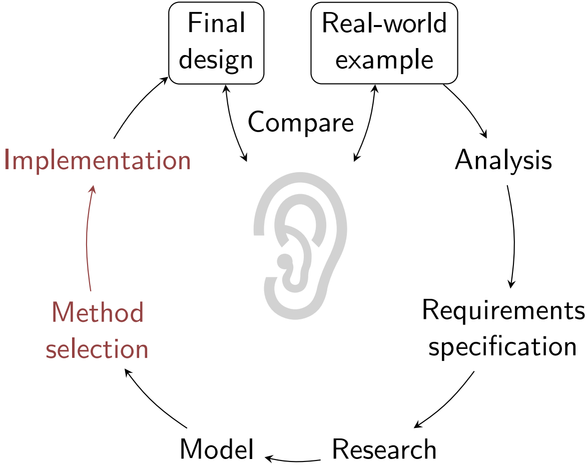


Figure: Stages of the sound design process (after Farnell 2010, figs. 16.7, 16.1)

Example: Steam train drive-by

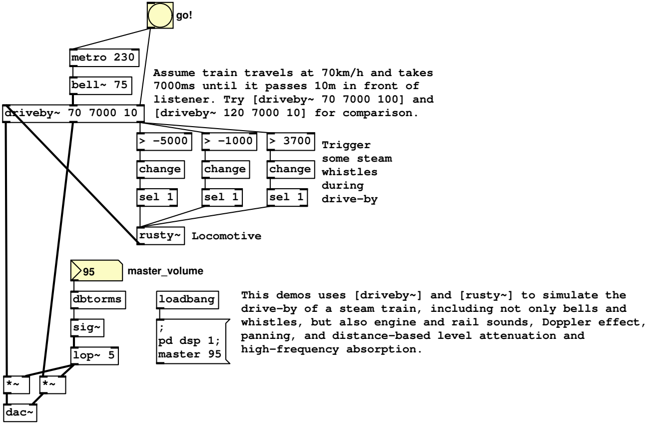


Figure: Steam train drive-by synthesized in Pd

Example: Steam train drive-by

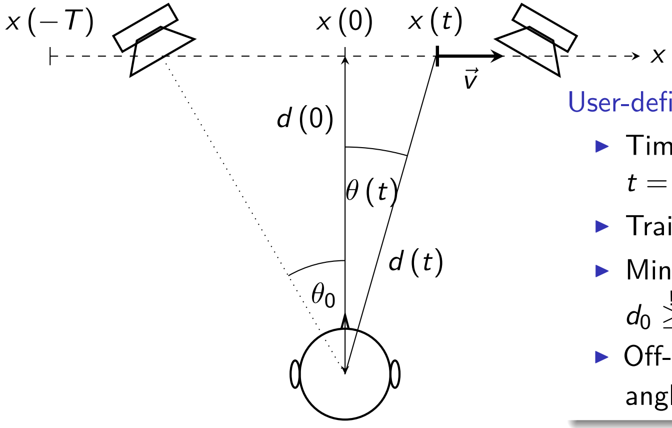


Figure: Geometry of steam train drive-by

User-definable parameters

- Time T from 'go' to $t = 0$
- Train velocity v
- Minimum distance $d_0 \overset{!}{\geq} 10\text{m}$
- Off-center loudspeaker angle θ_0

Example: Steam train drive-by

A	B	C	D
[redacted]	[redacted]	[redacted]	[redacted]
Steam whistle	Rail joints	Steam engine	Bell
Doppler	Distance attenuation	HF absorption	Panning

Table: Student groups

Method selection

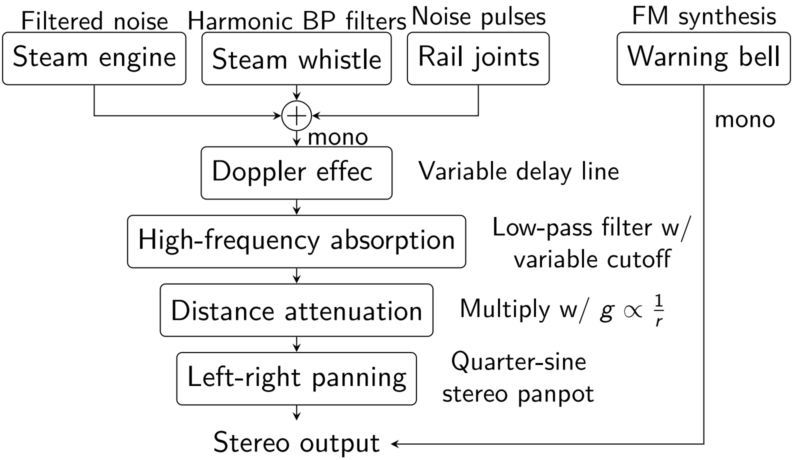
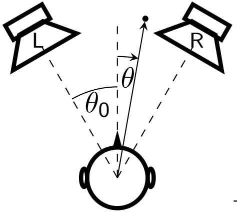
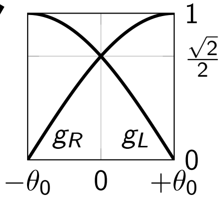


Figure: Block diagram of steam train drive-by model

Left-right panning





Quarter cosine panpot law

$$g_L = \cos\left(\frac{\pi \cdot (\theta + \theta_0)}{4 \cdot \theta_0}\right)$$

$$g_R = \cos\left(\frac{\pi \cdot |\theta - \theta_0|}{4 \cdot \theta_0}\right)$$

- g_L, g_R ... loudspeaker gains (0 to 1)
- θ ... desired phantom source direction ($-\theta_0 \leq \theta \leq +\theta_0$)
- θ_0 ... off-center loudspeaker angle (30° for standard stereo setup)

Implementation

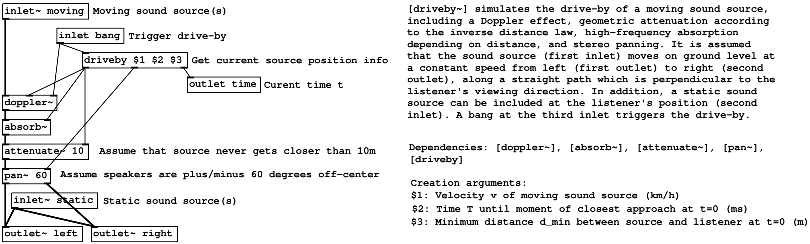


Figure: Steam train drive-by synthesized in Pd

21M.380 Music and Technology
Sound Design

Lecture 20: Steam train drive-by

Massachusetts Institute of Technology
Music and Theater Arts

Wednesday, April 20, 2016

Review EX3 assignment

A	B	C	D
[redacted]	[redacted]	[redacted]	[redacted]
Steam whistle Doppler	Rail joints Distance attenuation	Steam engine HF absorption	Bell Panning

Table: Student groups

Review EX3 assignment

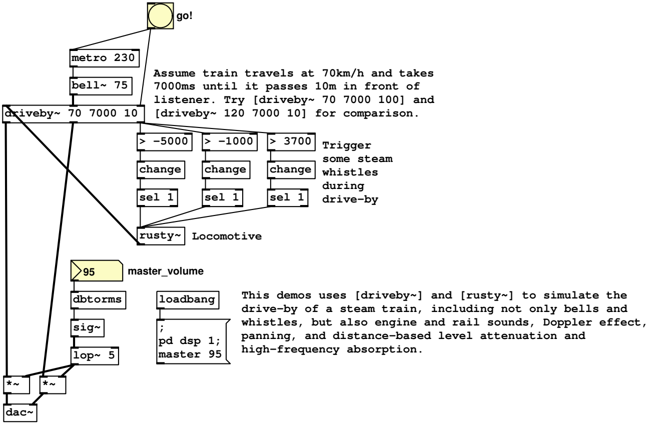


Figure: Steam train drive-by synthesized in Pd

Artificial sounds

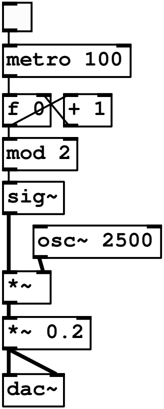


Figure: Pedestrian crossing (Farnell 2010, fig. 24.4)

Artificial sounds

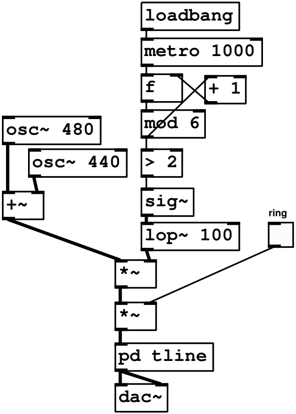


Figure: Phone tones (Farnell 2010, fig. 25.5)

Artificial sounds

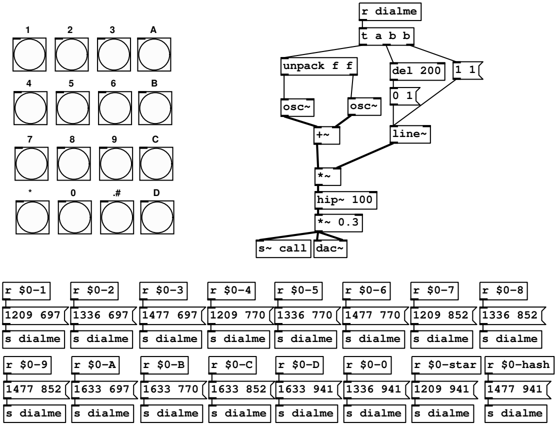


Figure: DTMF tones (Farnell 2010, fig. 26.3)

Artificial sounds

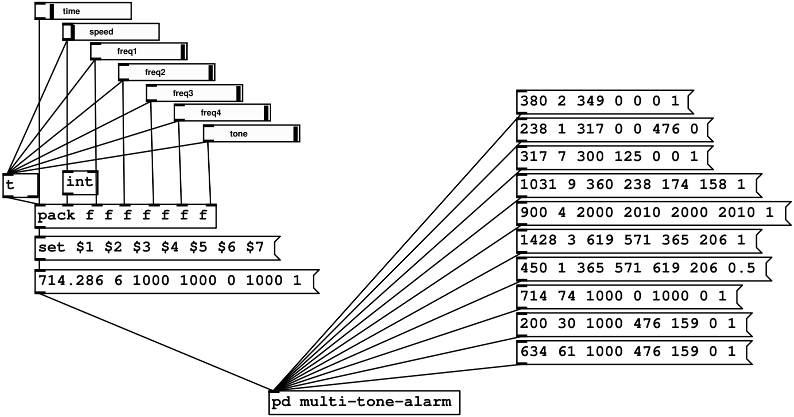


Figure: Alarm generator (Farnell 2010, fig. 27.7)

Artificial sounds

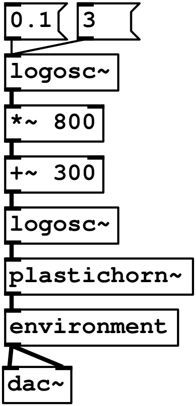


Figure: Police siren (Farnell 2010, fig. 28.9)

Idiophonics

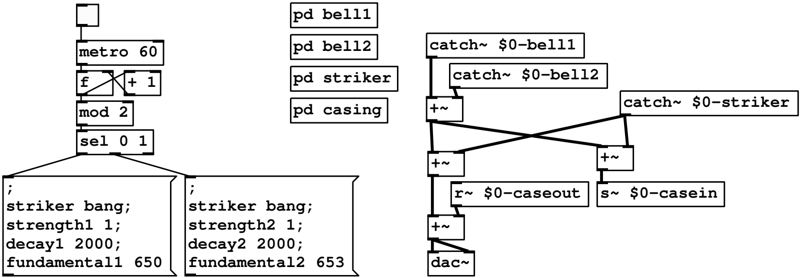


Figure: Telephone bell (Farnell 2010, fig. 29.15)

Idiophonics

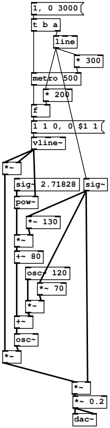


Figure: Bouncing (Farnell 2010, fig. 30.2)

Idiophonics

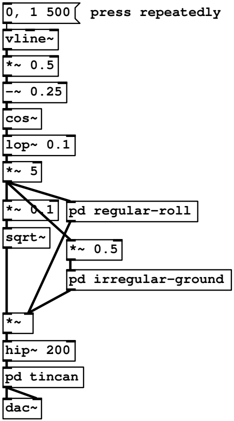


Figure: Rolling (Farnell 2010, fig. 31.8)

Idiophonics

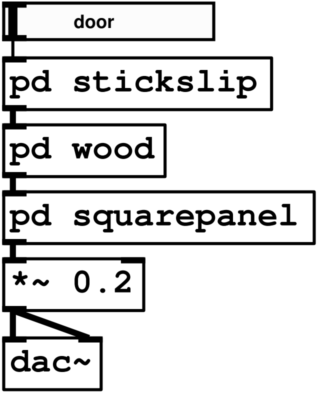


Figure: Creaking (Farnell 2010, fig. 32.6)

Idiophonics

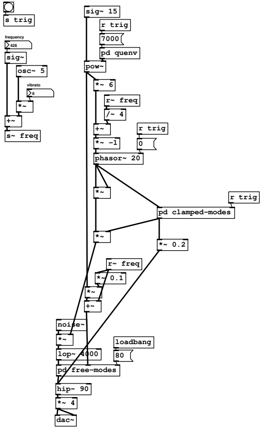


Figure: Boing (Farnell 2010, fig. 33.3)

Sounds of nature

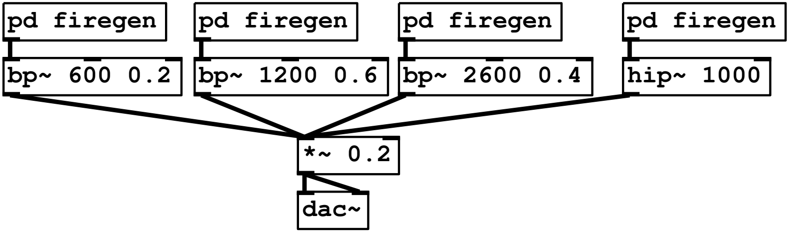


Figure: Fire (Farnell 2010, fig. 34.13)

Sounds of nature

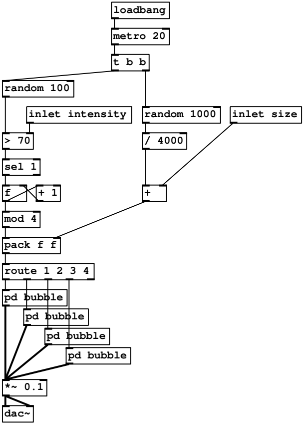


Figure: Bubbles (Farnell 2010, fig. 35.12)

Sounds of nature

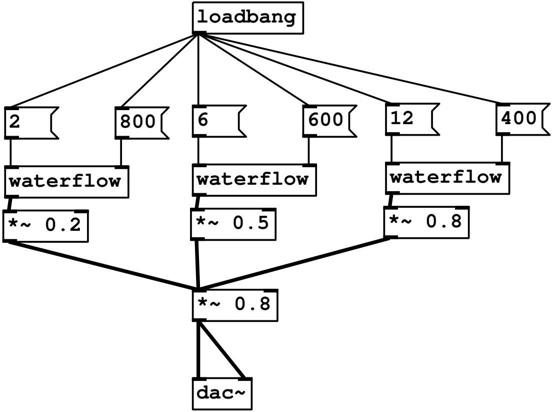


Figure: Running water (cf., Farnell 2010, ch. 36)

Sounds of nature

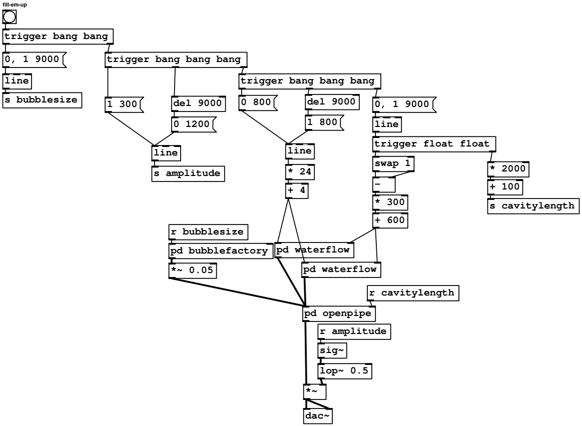


Figure: Pouring (Farnell 2010, fig. 37.3)

Sounds of nature

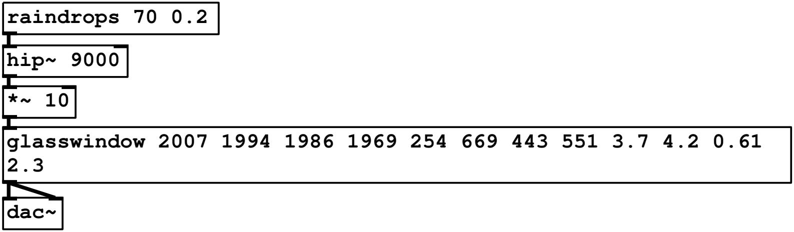


Figure: Rain (Farnell 2010, fig. 38.6)

Sounds of nature

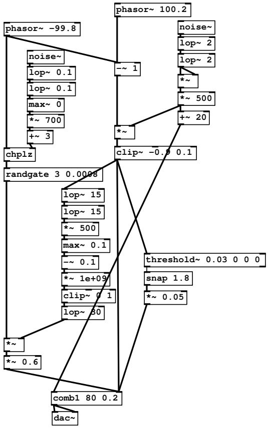


Figure: Electricity (Farnell 2010, fig. 39.8)

Sounds of nature

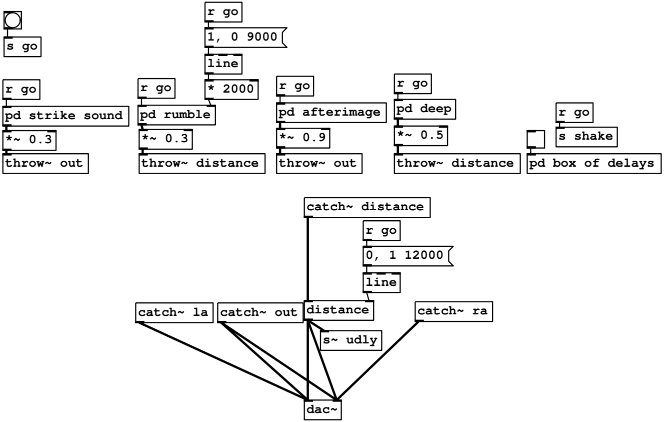


Figure: Thunder (Farnell 2010, fig. 40.13)

Sounds of nature

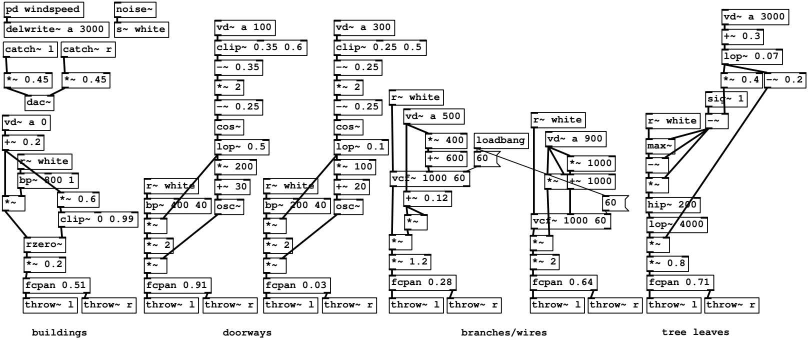


Figure: Wind (cf., Farnell 2010, ch. 41)

Sounds of machines

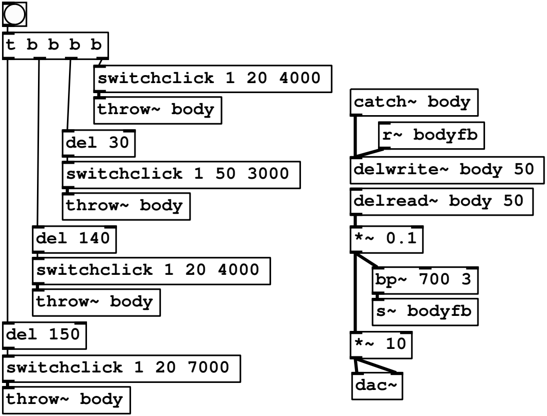


Figure: Switches (Farnell 2010, fig. 42.4)

Sounds of machines

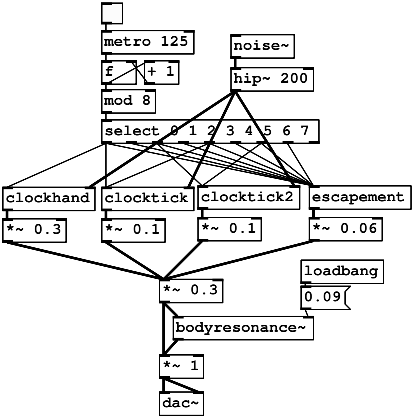


Figure: Clocks (Farnell 2010, fig. 43.12)

Sounds of machines

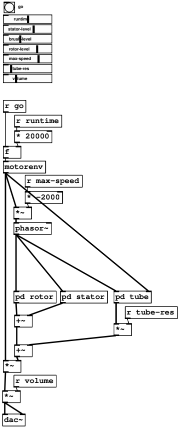


Figure: DC motors (cf., Farnell 2010, ch. 44)

Sounds of machines

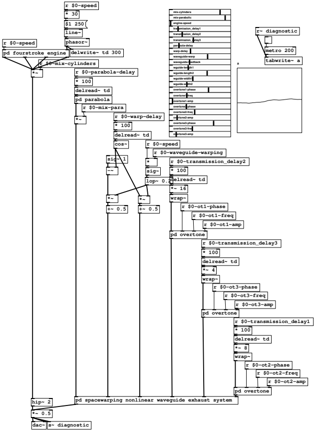


Figure: Cars (Farnell 2010, fig. 45.8)

Sounds of machines

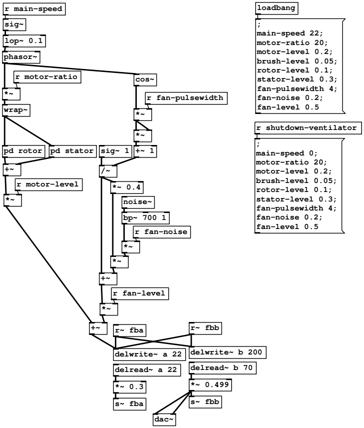


Figure: Fans (Farnell 2010, fig. 46.6)

Sounds of machines

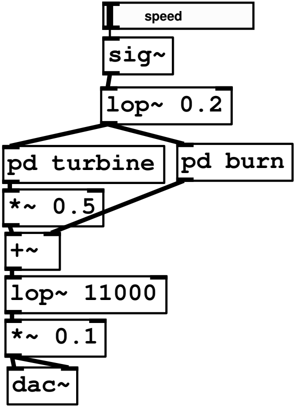


Figure: Jet engine (Farnell 2010, fig. 47.6)

Sounds of machines

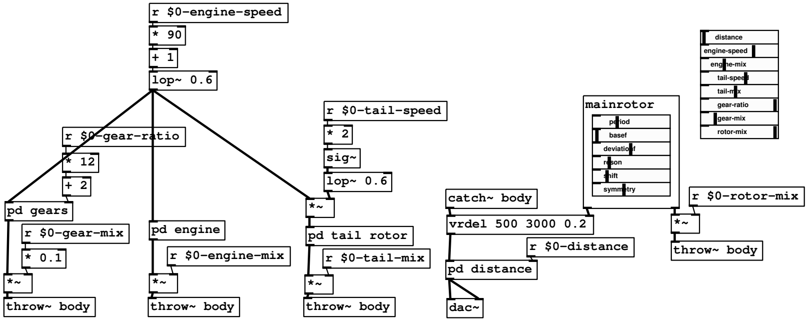


Figure: Helicopter (Farnell 2010, fig. 48.17)

Sounds of life

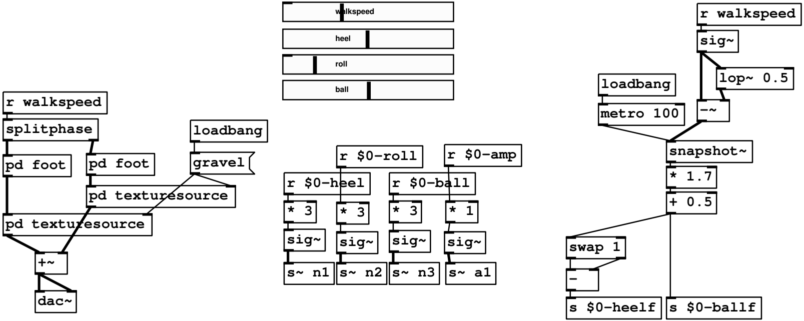


Figure: Footsteps (Farnell 2010, fig. 49.8)

Sounds of life

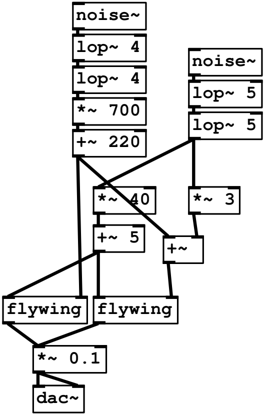


Figure: Insects (Farnell 2010, fig. 50.14)

Sounds of life

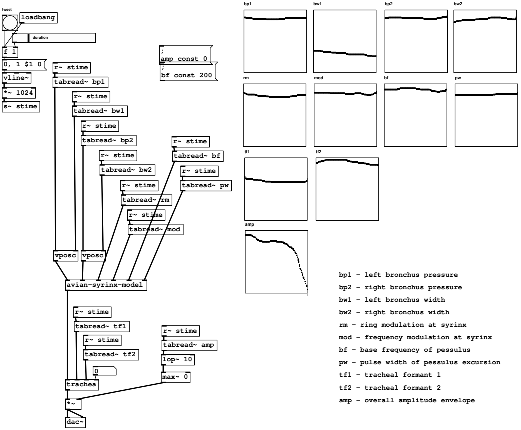


Figure: Birds (Farnell 2010, fig. 51.7)

Sounds of life

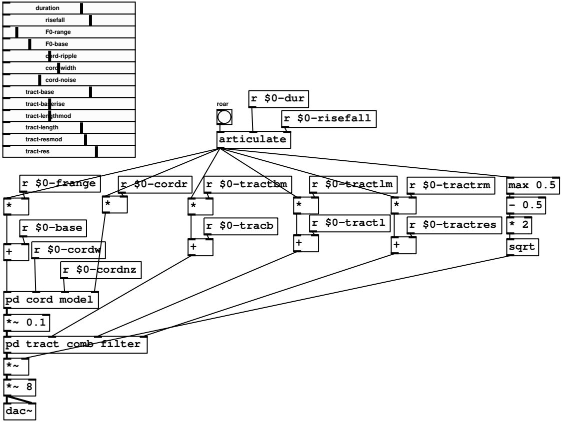


Figure: Mammals (Farnell 2010, fig. 52.5)

Sounds of mayhem

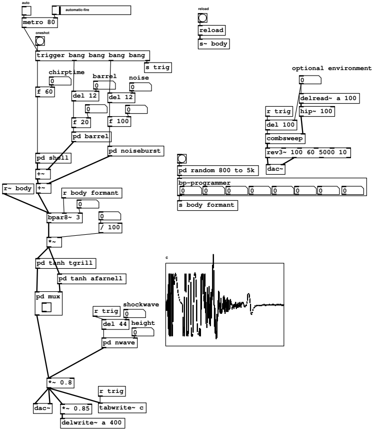


Figure: Guns (Farnell 2010, fig. 53.9)

Sounds of mayhem

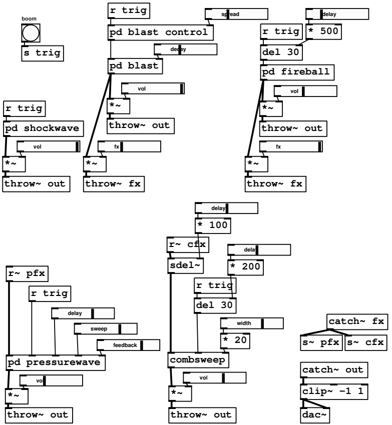


Figure: Explosions (Farnell 2010, fig. 54.9)

Sounds of mayhem

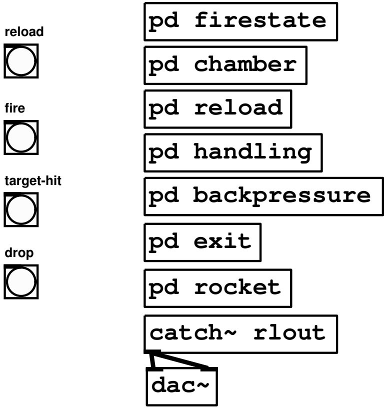


Figure: Rocket launcher (Farnell 2010, fig. 55.1)

Science fiction

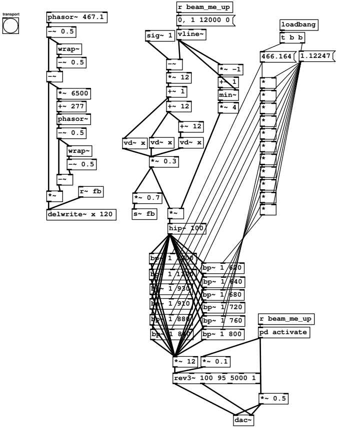


Figure: Transporter (Farnell 2010, fig. 56.2)

Science fiction

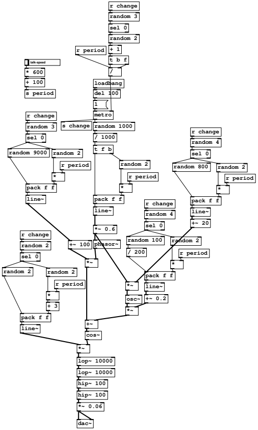


Figure: R2D2 (Farnell 2010, fig. 57.3)

Science fiction

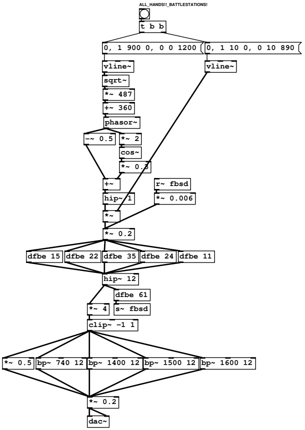


Figure: Red alert (Farnell 2010, fig. 58.7)

21M.380 Music and Technology
Sound Design

Lecture 21: Granular synthesis

Massachusetts Institute of Technology
Music and Theater Arts

Monday, April 25, 2016

General principle

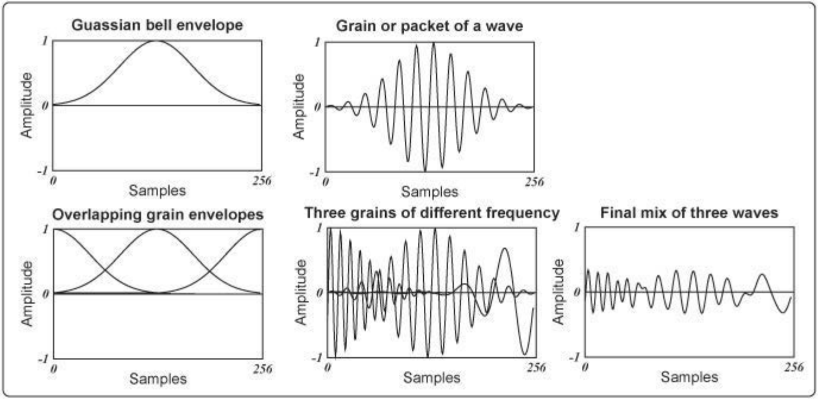
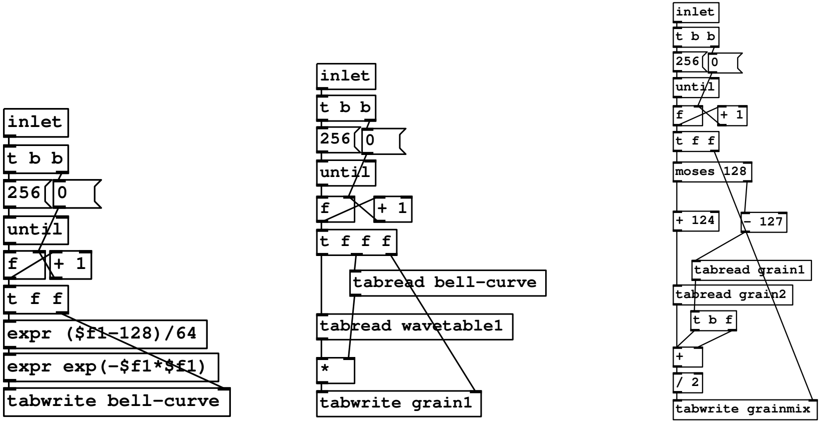


Figure: Granular synthesis of multiple sources using overlapping grains (Farnell 2010, fig. 21.1. Courtesy of MIT Press. Used with permission. `https://mitpress.mit.edu/books/designing-sound`)

Generating grains in Pd



(a) Make bell curve (b) Make grain (c) Mix grains

Figure: Operations on tables for granular synthesis (Farnell 2010, fig. 21.3)

Iannis Xenakis

Figure: Iannis Xenakis, composer of 'Concret PH' (1958) (Courtesy of The Friends of Xenakis. Used with permission.)

Barry Truax

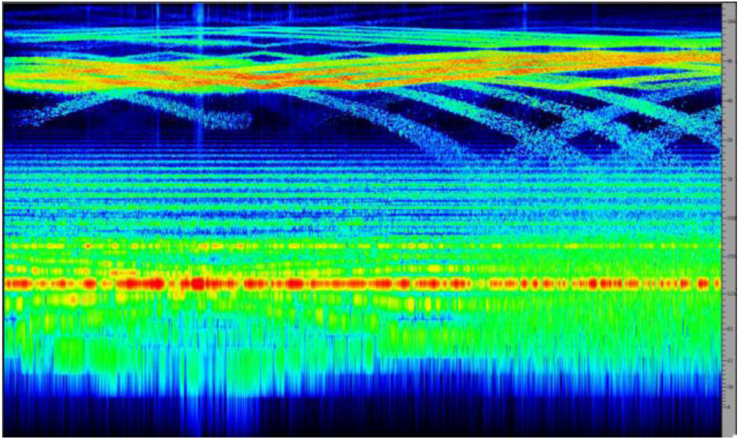


Figure: Spectrogram of Barry Truax's computer music work *Riverrun* (1987/2004), section 5 (Used with permission of the composer and Cambridge Street Publishing, CSR-DVD 0801, 2008)

Horacio Vaggione

Figure: Horacio Vaggione, composer of 'Nodal' (1997) (© Bernard Bruges-Renard.

Curtis Roads

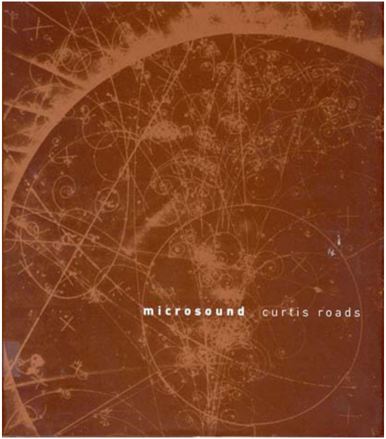


Figure: The *Microsound* book by Curtis Roads (2002), composer of 'Half-life' (1998–1999) (Courtesy of MIT Press. Used with permission. https://mitpress.mit.edu/books/microsound)

Applications

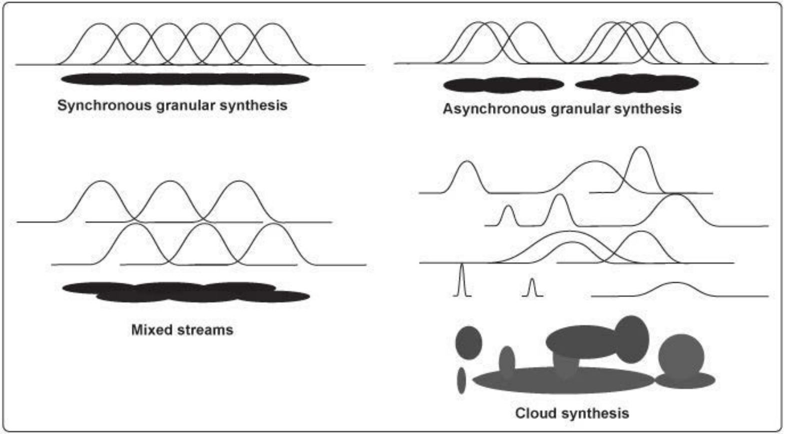


Figure: Types of granular synthesis (Farnell 2010, fig. 21.4. Courtesy of MIT Press. Used with permission. `https://mitpress.mit.edu/books/designing-sound`)

Applications

Applications of synchronous granular synthesis (Farnell 2010, pp. 307 f.)

- Time-stretching (w/o changing pitch)
- Pitch-shifting (w/o changing speed)

Applications of asynchronous granular synthesis (Farnell 2010, p. 257)

- Textures (water, fire, wind, rain, crowds of people, flocks, swarms)

Challenges (Farnell 2010, pp. 257, 305)

- Lots of control data (but can often be automated)
- Computationally expensive
- Lack of precision

Time stretching and pitch shifting

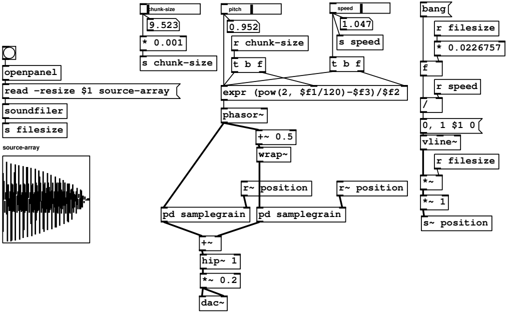


Figure: Time stretch and pitch shift using overlapping grains on opposite phases (Farnell 2010, fig. 21.8)

Demo

1. Load mono sound file via [openpanel] (try speech)
2. Turn on DSP
3. Trigger top right [bang(
4. Adjust pitch and speed
5. Re-trigger [bang(

Sound textures

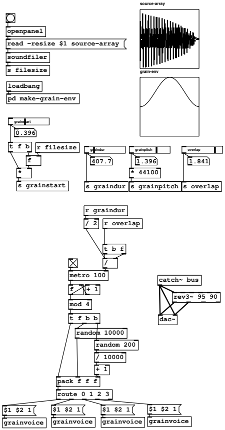


Figure: Sustained texture pad with four overlapping grain generators (Farnell 2010, fig. 21.6)

Demo

1. Load pitched mono sound file (e.g., voice, string, brass) via [openpanel]
2. Turn on DSP
3. Toggle [metro]
4. Adjust grainstart, graindur, grainpitch, overlap

Sound textures

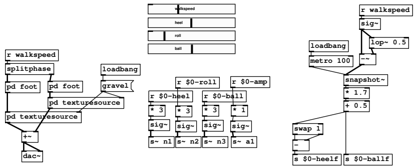


Figure: Grain generators in [pd texturesource] → [pd gravtex] → [pd gravel] (Farnell 2010, fig. 49.8)

Demo

1. Turn on DSP
2. Adjust walkspeed and roll
3. Change surface texture
4. Edit [gravel(message to [snow(, [dirt(, [wood(, or [grass(
5. Re-trigger message box

21M.380 Music and Technology
Sound Design

Lecture 22: Student presentations

Massachusetts Institute of Technology
Music and Theater Arts

Wednesday, April 27, 2016

21M.380 Music and Technology
Sound Design

Lecture 23: Quiz and student presentations

Massachusetts Institute of Technology
Music and Theater Arts

Monday, May 2, 2016

21M.380 Music and Technology
Sound Design

Lecture 24: Thunder

Massachusetts Institute of Technology
Music and Theater Arts

Wednesday, May 4, 2016

Thunder

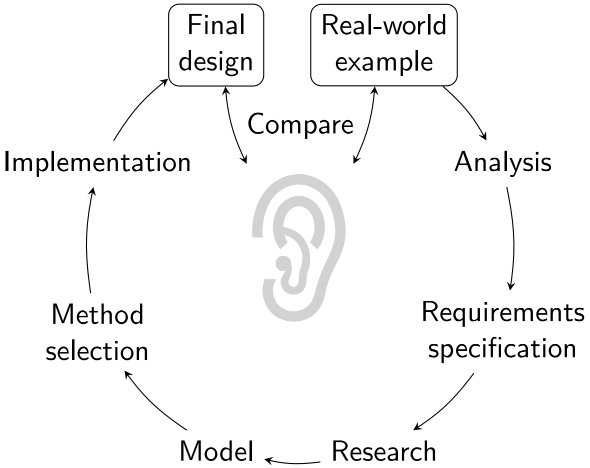


Figure: Stages of the sound design process (after Farnell 2010, figs. 16.7, 16.1)

Thunder

A	B	C	D
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	

Table: Student groups

Stereo system setup

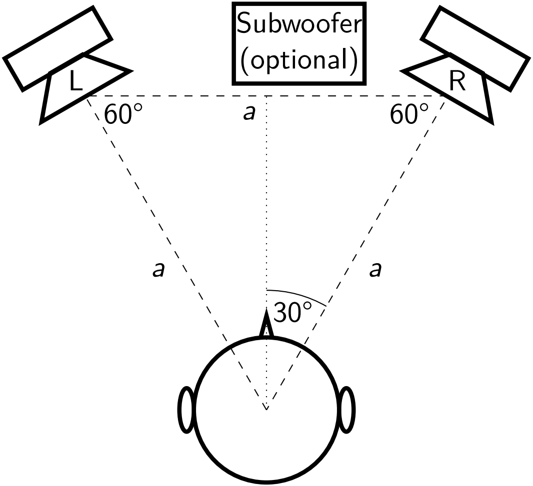


Figure: Standard stereo loudspeaker setup

Stereo system setup

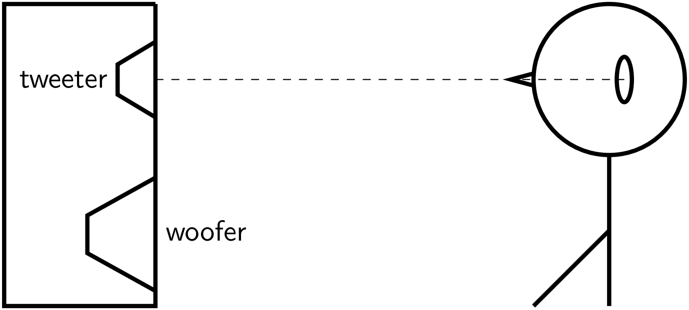


Figure: Align the tweeter with the listener's ears

Thunder strike

- Thunder strike releases 10×10^9 J of energy
- Air in path of spark heats to plasma at 30 000 °C
- Air expands very rapidly (cylindrical shockwave along bolt)
- Air cools and collapses back
- Resulting waveform is shaped like an N
- Multiple discharges decay exponentially (up to 50 strikes within 50 ms)

Multistrike discharges

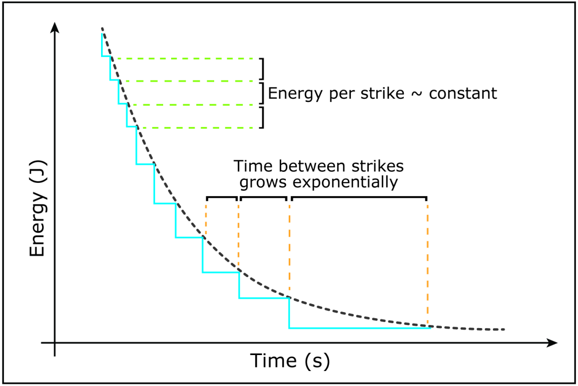


Figure: Discharge of multiple strikes (Image by MIT OpenCourseWare, after Farnell 2010, fig. 40.1.)

Tortuosity

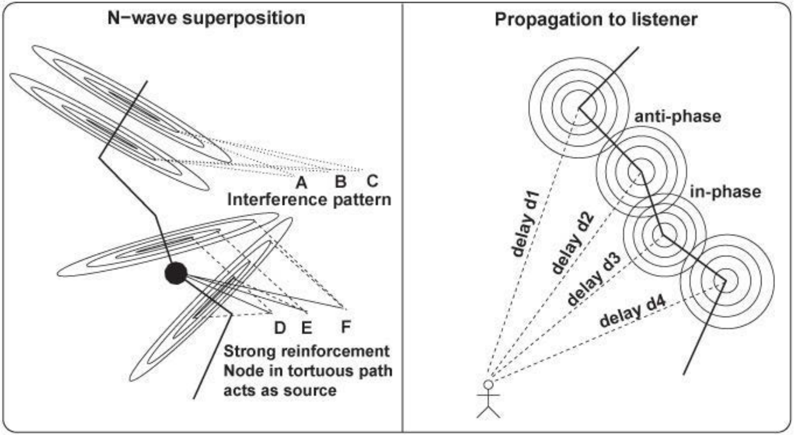


Figure: N-wave interference at the observer's position (Farnell 2010, fig. 40.2. Courtesy of MIT Press. Used with permission. `https://mitpress.mit.edu/books/designing-sound`)

Environmental factors

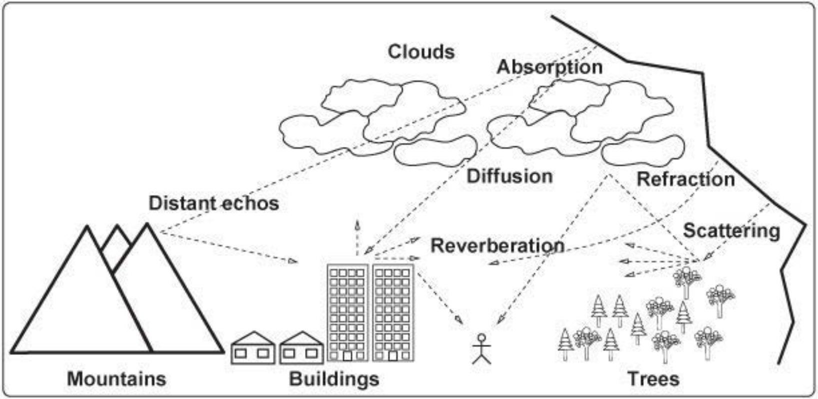
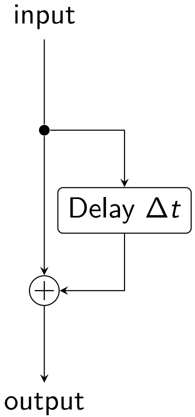
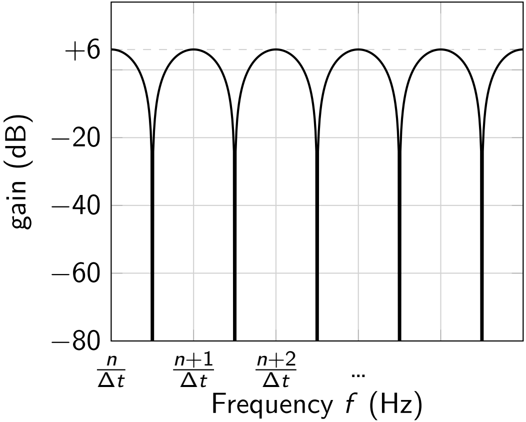


Figure: Environmental factors in thunder sound (Farnell 2010, fig. 40.4. Courtesy of MIT Press. Used with permission.
https://mitpress.mit.edu/books/designing-sound)

Comb filtering



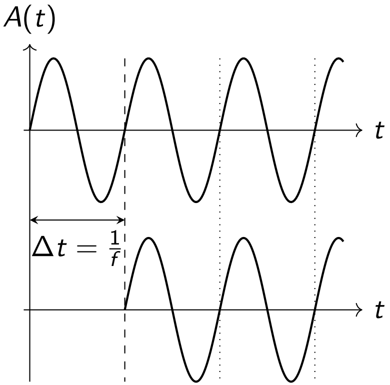
(a) Flow chart



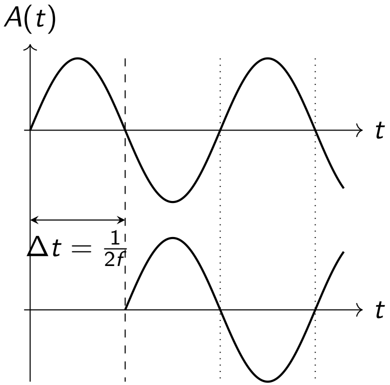
(b) Frequency response (note linear x axis)

Figure: Principle of a comb filter

Comb filtering



(a) Constructive interference



(b) Destructive interference

Figure: Mixing a signal with a delayed copy of itself results in an interference pattern that depends on frequency.

Comb filtering

$$\Delta t = \frac{\Delta d}{c} = \frac{d_2 - d_1}{c} = \frac{2\sqrt{h^2 + \left(\frac{d_1}{2}\right)^2} - d_1}{c}$$

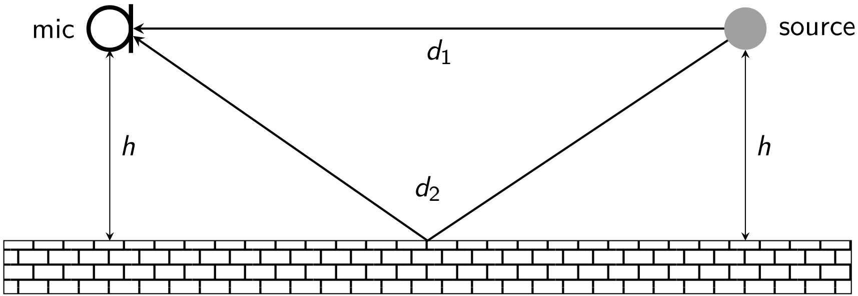


Figure: Comb filter effect caused by single reflection

Dispersion and absorption

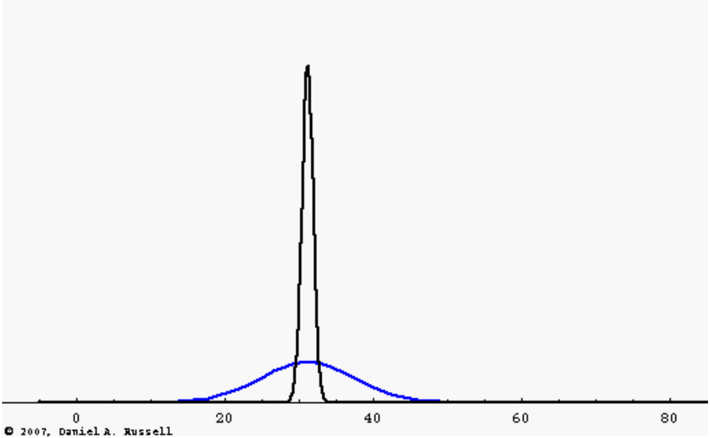


Figure: Dispersion causes the shape of a wave pulse to change as it travels

Dispersion and absorption

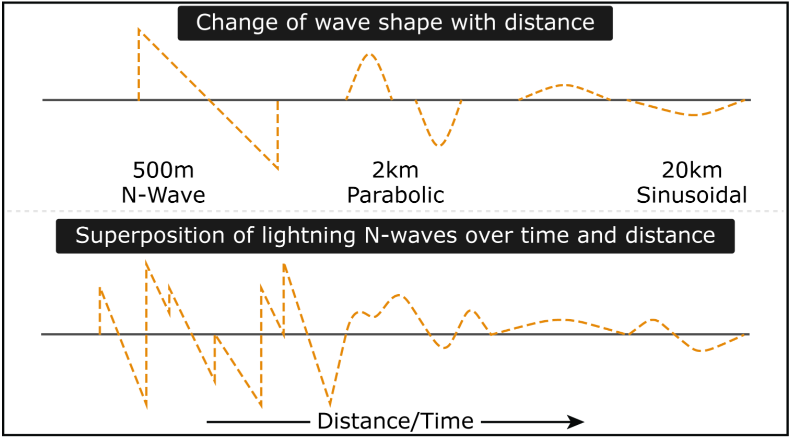
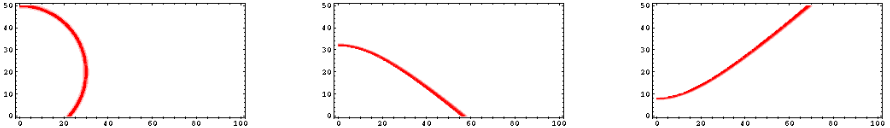


Figure: Waveform produced by N-wave superposition at a distance (Image by MIT OpenCourseWare, after Farnell 2010, fig. 40.3.)

Refraction



(a) No refraction (b) Upwards (c) Downwards

Figure: Refraction (© Daniel A. Russell. Grad. Prog. Acoustics, Penn State.)

Diffraction

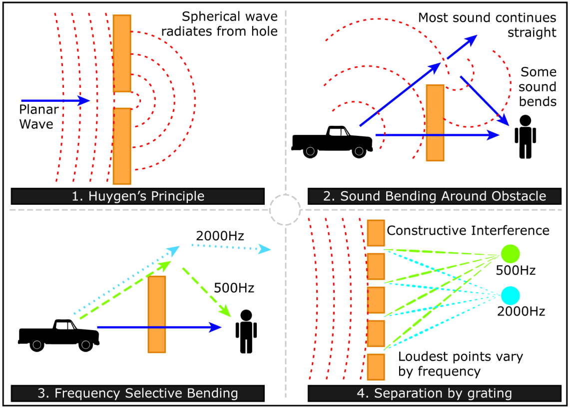


Figure: Sound diffraction effects (Image by MIT OpenCourseWare, after Farnell 2010, fig. 5.7.)

Reflections

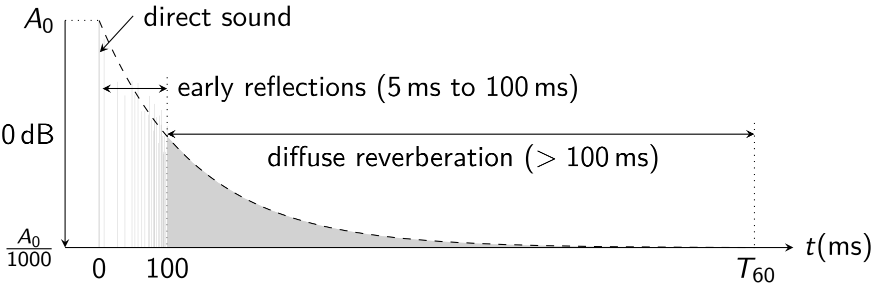


Figure: Typical impulse response of reverberation

Reflections

Material	α 125 Hz	α 500 Hz	α 2000 Hz
Acoustical tile	0.20	0.65	0.65
Brick wall (unpainted)	0.02	0.03	0.05
Heavy carpet on heavy pad	0.10	0.60	0.65
Concrete (painted)	0.01	0.01	0.02
Heavy draperies	0.15	0.55	0.70
Fiberglass blanket (7.5 cm thick)	0.60	0.95	0.80
Glazed tile	0.01	0.01	0.02
Paneling (0.30 cm thick)	0.30	0.10	0.08
Vinyl floor on concrete	0.02	0.03	0.04
Wood floor	0.06	0.06	0.06

Table: Absorption coefficient α for different materials (Hartmann 2013, p. 165)

Implementation

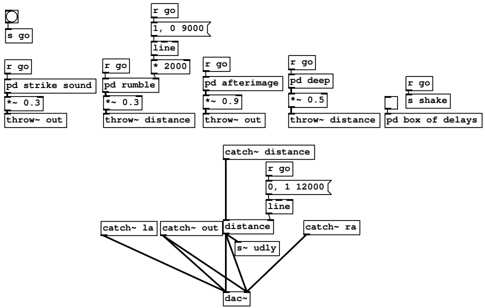


Figure: A patch to produce thunder made of several separate components (Farnell 2010, fig. 40.13)

Demo

1. Turn on DSP
2. Trigger [s go] (top left) a few times
3. Toggle [pd box of delays]
4. Re-trigger and compare

21M.380 Music and Technology
Sound Design

Lecture 25: Music synthesizers

Massachusetts Institute of Technology
Music and Theater Arts

Monday, May 9, 2016

History of music synthesizers

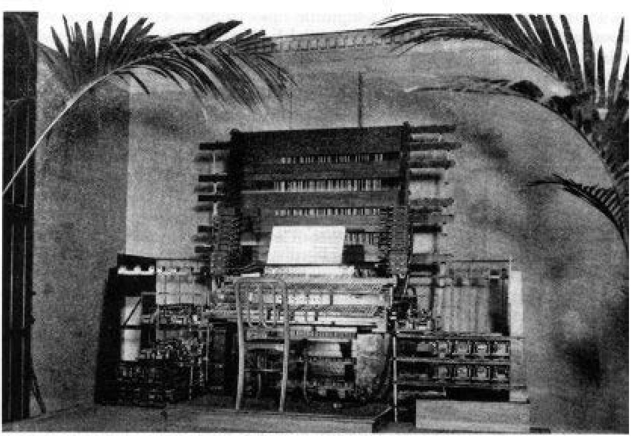


Figure: Thaddeus Cahill's 1897 Telharmonium (Ⓟ Public domain image. Source: https://en.wikipedia.org/wiki/File:Teleharmonium1897.jpg)

History of music synthesizers

Figure: Clara Rockmore on Lev Termen's theremin, patented 1928 ()

History of music synthesizers

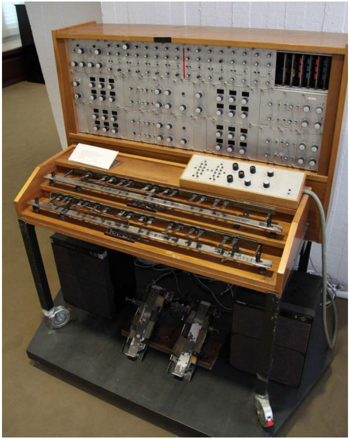


Figure: Oskar Sala's Mixtur-Trautonium (1950s), a development of Friedrich Trautwein's original Trautonium (© Wikipedia user: Morn the Gorn.

History of music synthesizers

Figure: Hugh Le Caine's 1948 Electronic Sackbut (Courtesy of David Carroll on Flickr.)

History of music synthesizers

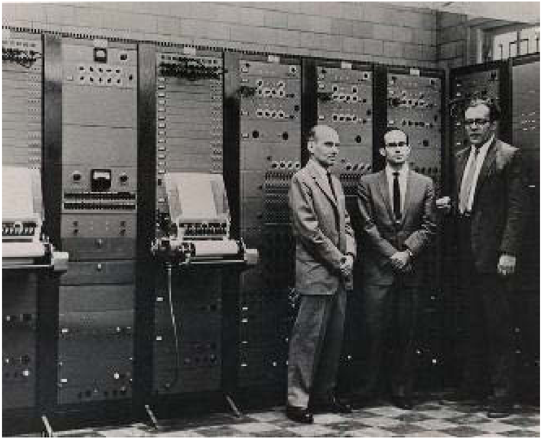


Figure: The RCA Mark II Synthesizer (1957) at the Columbia-Princeton Electronic Music Center (Courtesy of Columbia University Computer Music Center. Used with permission)

History of music synthesizers

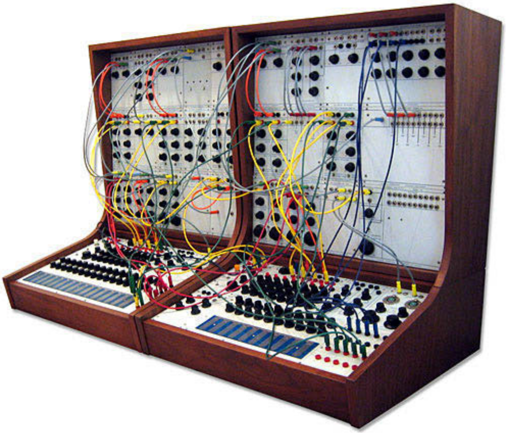


Figure: Don Buchla's 100 series synthesizer (1963) (© rick604 on Flickr. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/)

History of music synthesizers

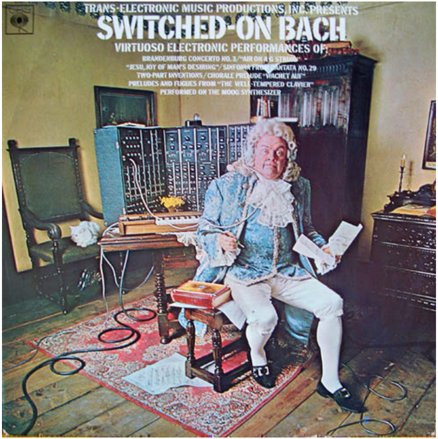


Figure: Wendy/Walter Carlos' Switched-On Bach (1968) was produced with a Moog synthesizer ()

History of music synthesizers

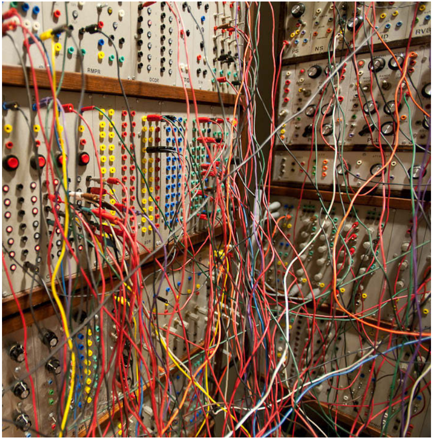


Figure: Joe Paradiso's modular synthesizer (1974–88) (© Joe Paradiso. All rights reserved. This content is excluded from our Creative Commons license. For more information, see `http://ocw.mit.edu/help/faq-fair-use/`)

History of music synthesizers

Image removed due to copyright restrictions. Photo of Annie Lennox from the video.

Figure: Eurythmics – Sweet Dreams (Are Made of This)

21M.380 Music and Technology
Sound Design

Lecture 26: Final project presentations

Massachusetts Institute of Technology
Music and Theater Arts

Wednesday, May 11, 2016

Cited readings

Ament, Vanessa Theme (2009a). *The Foley Grail. The Art of Performing Sound for Film, Games, and Animation*. 1st ed. Focal Press. 216 pp. ISBN: 978-0-240-81125-3. MIT LIBRARY: 002181600. Printed copy (incl. DVD) on course reserve at the Lewis Music Library.

– (2009b). "What we use for .... Performing the props." In: *The Foley Grail. The Art of Performing Sound for Film, Games, and Animation*. 1st ed. Focal Press. Chap. 8, pp. 89–101. ISBN: 978-0-240-81125-3. MIT LIBRARY: 002181600. Available at: MIT Learning Modules ▸ Materials.

Benson, Dave (2008). *Music: a Mathematical Offering*. URL: https://homepages.abdn.ac.uk/mth192/pages/html/music.pdf (visited on 03/07/2015).

Blauert, Jens (1996). *Spatial Hearing. The Psychophysics of Human Sound Localization*. revised second. Cambridge, MA and London: MIT Press. 508 pp. ISBN: 978-0-262-02413-6. MIT LIBRARY: 000808775.

Cited readings (cont.)

Bregman, Albert S. and Pierre Ahad (1996). *Demonstrations of Auditory Scene Analysis. The Perceptual Organization of Sound. Audio compact disk.* Montréal, Canada: Auditory Perception Laboratory, Psychology Department, McGill University.

Burk, Phil et al. (2011). *Music and computers. A theoretical and historical approach.* URL: http://music.columbia.edu/cmc/MusicAndComputers/ (visited on 04/06/2015). Archival version.

Carlyle, Angus, ed. (2007). *Autumn Leaves. Sound and the Environment in Artistic Practice.* Paris: Double Entendre. 128 pp. ISBN: 978-0-9548074-3-6. MIT LIBRARY: 002198647.

Chowning, John (1973). "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation." In: *Journal of the Audio Engineering Society* 21.7, pp. 526–34. URL: http://www.aes.org/e-lib/browse.cfm?elib=1954 (visited on 08/09/2014).

Cited readings (cont.)

Crawford, Chris (1997a). *The Art of Computer Game Design*. Electronic edition of a text originally published in 1982.

– (1997b). "The computer as a game technology." In: *The Art of Computer Game Design*. Electronic edition of a text originally published in 1982. Chap. 4, pp. 35–44. URL: http://www-rohan.sdsu.edu/~stewart/cs583/ACGD_ArtComputerGameDesign_ChrisCrawford_1982.pdf (visited on 01/19/2015).

Doornbusch, Paul (2005). *The Music of CSIRAC. Australia's First Computer Music*. Common Ground. 118 pp. ISBN: 978-1863355698. MIT LIBRARY: 001401692.

Farnell, Andy (2010). *Designing Sound*. Cambridge, MA and London: MIT Press. 688 pp. ISBN: 978-0-262-01441-0. MIT LIBRARY: 001782567. Hardcopy and electronic resource.

Cited readings (cont.)

Fildes, Jonathan (2008). *'Oldest' computer music unveiled.* URL: http://news.bbc.co.uk/2/hi/technology/7458479.stm (visited on 01/17/2015).

Film Sound Cliches (2015). *Film Sound Stereotypes and Common Logic Flaws*. URL: http://www.filmsound.org/cliche/ (visited on 01/14/2015).

Guttman, Newman (1957). *In the Silver Scale.* URL: https://www.youtube.com/watch?v=PM64-lqYyZ8. First composition synthesized by a computer. Duration: 0'17".

Hartmann, William M. (2013). *Principles of Musical Acoustics*. Undergraduate Lecture Notes in Physics. Springer.

Hiller, Lejaren and Leonard Isaacson (1956). *Illiac Suite.* URL: https://www.youtube.com/watch?v=n0njBFLQSk8. First piece to be composed (but not synthesized) by a computer.

Cited readings (cont.)

International Organization for Standardization (1996). *Acoustics—Attenuation of sound during propagation outdoors. Part 2: General method of calculation*. ISO 9613-1/2. MIT LIBRARY: 001410672. URL: http://www.persona.uk.com/barnfield/Core_docs/G/G7.pdf (visited on 04/17/2015).

Loy, Gareth (2007). *Musimathics. The Mathematical Foundations of Music*. Vol. 1. Cambridge, MA and London: MIT Press. 482 pp. MIT LIBRARY: 001379675.

Lyons, Richard G. (2004). *Understanding Digital Signal Processing*. 2nd ed. Prentice Hall. 688 pp. ISBN: 978-0131089891. MIT LIBRARY: 001289139.

MacLeod, Hugh (2004). *How to be creative*. Recommended by Farnell (2010, p. 147). URL: http://changethis.com/manifesto/6.HowToBeCreative/pdf/6.HowToBeCreative.pdf.

Cited readings (cont.)

Mathews, Max (1963). "The Digital Computer as a Musical Instrument." In: *Science* 142.3592, pp. 553–7. JSTOR: 1712380. URL: http://www.jstor.org/stable/1712380.

Mathews, Max V. (1969). *The Technology of Computer Music*. Cambridge, MA and London: MIT Press. MIT LIBRARY: 000152007.

Mathews, Max, John Kelly, and Carol Lockbaum (1961). *Daisy Bell*. Also known as *Bicycle Built for Two*. URL: https://www.youtube.com/watch?v=41U78QP8nBk (visited on 01/17/2015). Programmed on an IBM 7094.

Nave, R. (2015). *Circular Membrane*. URL: http://hyperphysics.phy-astr.gsu.edu/hbase/music/cirmem.html#c4 (visited on 03/08/2015).

Roads, Curtis (2002). *Microsound*. Cambridge, MA and London: MIT Press. 424 pp. ISBN: 978-0-262-18215-7.

Cited readings (cont.)

Smith, Julius Orion (1991). "Viewpoints on the History of Digital Synthesis." In: *Proceedings of the International Computer Music Conference*. Montréal, pp. 1–10. URL: https://ccrma.stanford.edu/~jos/kna/Historical_View_Synthesizer_Development.html (visited on 01/16/2015).

Smith, Steven W. (1997). *The Scientist and Engineer's Guide to Digital Signal Processing*. 1st ed. California Technical Pub. 640 pp. URL: http://www.dspguide.com/.

Westerkamp, Hildegard (2007). "Soundwalking." In: *Autumn Leaves. Sound and the Environment in Artistic Practice*. Ed. by Angus Carlyle. Paris: Double Entendre, pp. 49–54. ISBN: 978-0-9548074-3-6. MIT LIBRARY: 002198647. Available in a slightly different version at http://cec.concordia.ca/econtact/Soundwalk/Soundwalking.html.

21M.380 Music and Technology: Sound Design
Spring 2016