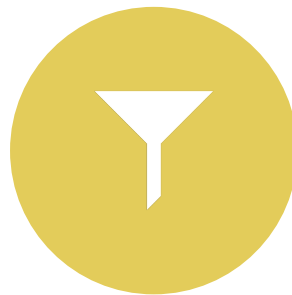


AUDIO FILTERS AND EFFECTS

AIST2010 Lecture 6



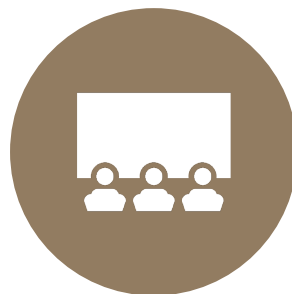
Noises



Filters



Effects



Spatialization

OUTLINE

EFFECTS AND FILTERS

Subtractive synthesis

- From a sound of rich spectrum, obtain the target sound by pruning

➤ *Sound processing*

- Frequency domain
- Time domain

Removal of unwanted components

Enhancement of preferred components

NOISES AS A RICH SPECTRUM

White noise: theoretical flat frequency response

- Random generator in CSound
 - `xr rand xamp [, iseed] [, isel] [, ioffset]`
 - If `iseed > 1`, system clock will be used as the random seed

Pink noise: equal energy in each octave (i.e. log scale)

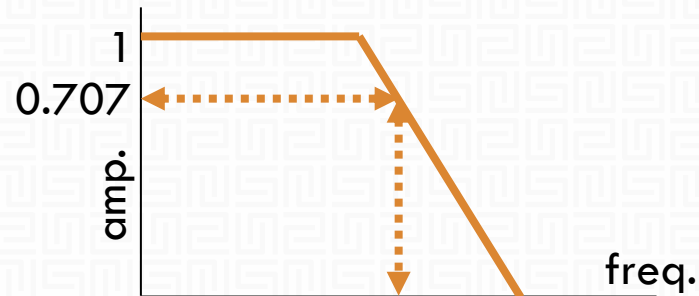
- `ar pinkish xamp [, ...]`

More CSound opcodes for noises

- <http://www.csounds.com/manual/html/SiggenNoise.html>

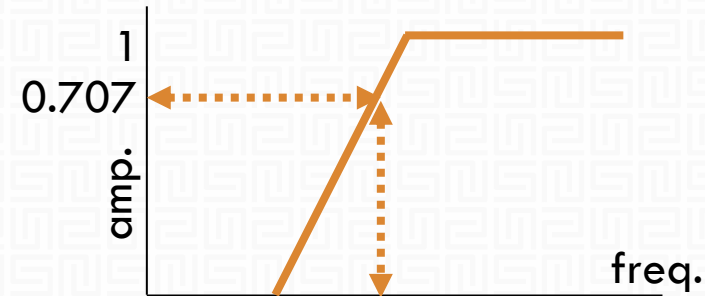
FILTERS

Filters are audio modifiers in the frequency domain



Half-power point

- Low-pass filter: removing high frequencies
 - `ar tone asig, khp [, iskip]`
 - **khp** — half-power point, sometimes referred to as the cut-off frequency



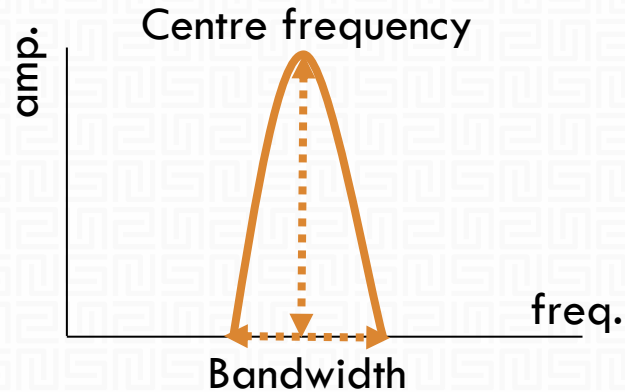
Half-power point

- High-pass filter: removing low frequencies
 - `ar atone asig, khp [, iskip]`

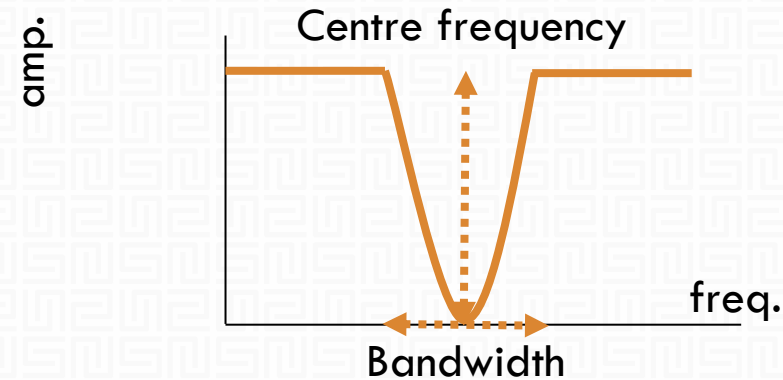
*First-order filters only: to improve the **roll-off**, repeat the filter!*

- Bright sound vs. dark sound

FILTERS

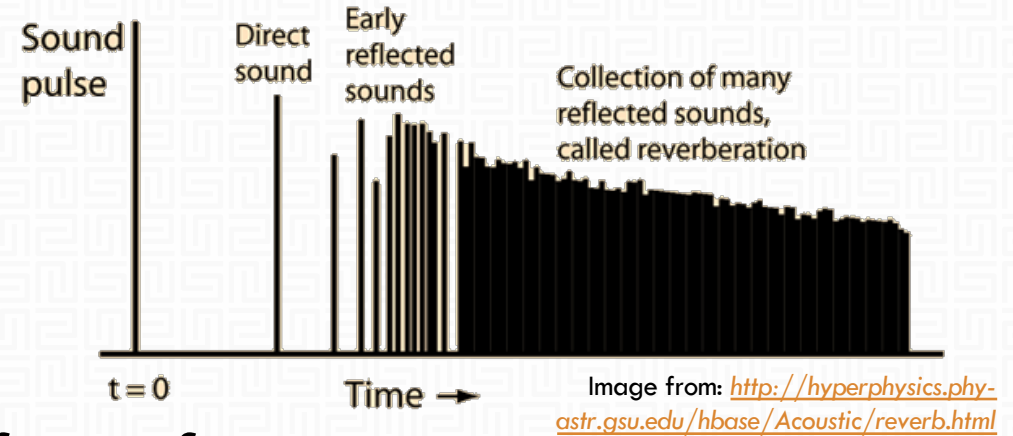


- Band-pass (*resonant*) filter: suppress anything outside the centre frequency
 - `ares reson asig, xcf, xbw`
 `[, iscl] [, iskip]`
 - `xcf` — centre frequency (peak response)
 - `xbw` — difference between upper and lower half-power points
 - `iscl` — use 1 for scaling max amplitude



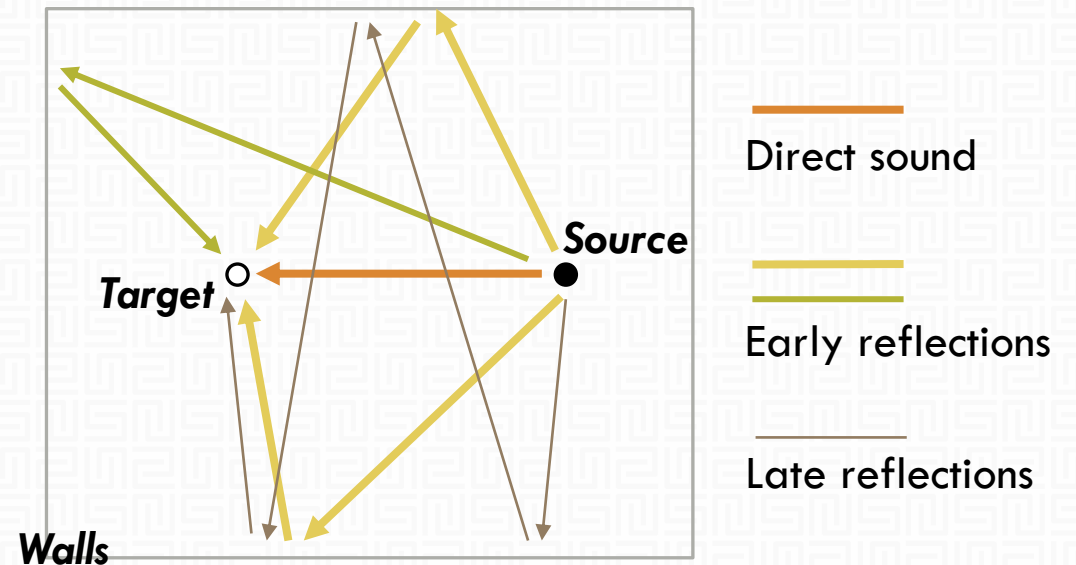
- Band-stop (*notch*) filter: suppress anything around the centre frequency
 - `ares areson asig, xcf, xbw`
 `[, iscl] [, iskip]`
 - `iscl` — use 1 for scaling max amplitude
 - Useful for removing electricity hum at 50 Hz

EFFECTS



Reverberation: the (psycho)acoustic effect of a room

- A set of echoes (reflected sound)
- Reverberation time RT_{60} = the time for the sound to fade away (by **60dB**)
- Wet sound vs. dry sound
- CSound reverb opcodes
 - alpass
 - comb
 - freeverb



EFFECTS

Compressor/Limiter: Dynamic range reduction

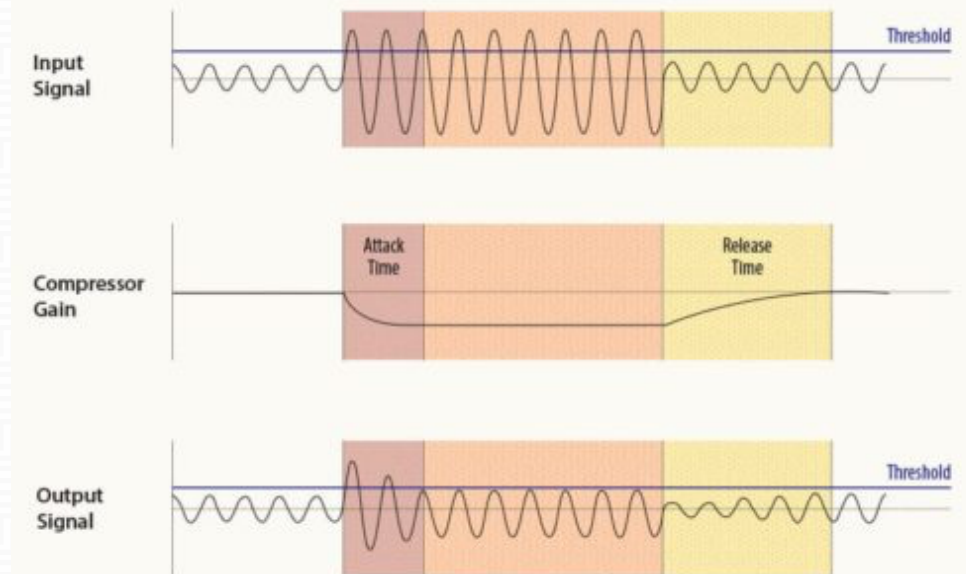
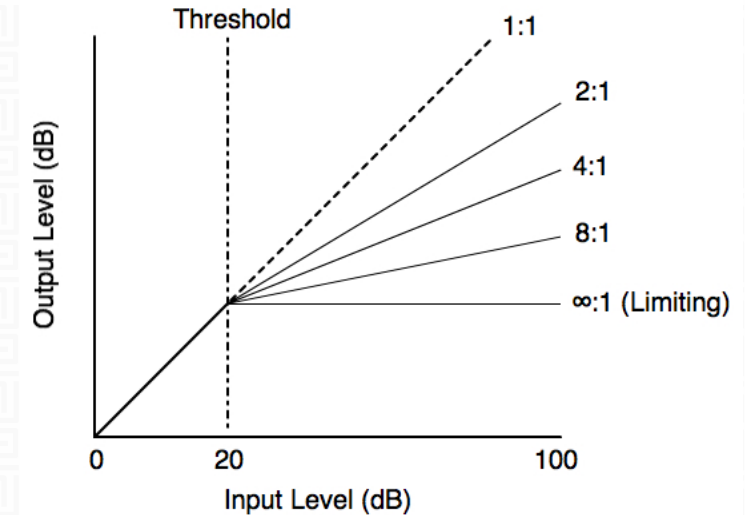
- Sometimes we prefer to avoid too much difference in dynamic range, so that both soft and loud parts of the music can be easily enjoyed

Compressor in CSound for compressing, limiting, expanding, ducking, or gating a signal

- ar **compress** aasig, acsig, kthresh, \ kloknee, khiknee, kratio, katt, krel, ilook
 - Measured in decibels

Read more:

- <https://www.gysnoizerecordings.com/music-production/audio-compression-basics/>



MORE EFFECTS

Distort: waveshaping and optional clipping

- **ar distort** asig, kdist, ifn \
 [, ihp, istor]
 - kdist — amount of distortion between 0 and 1
 - ifn — waveshaping function to modify shape and spectrum

Flanger: “chorus” and “flanger”

- **ar flanger** asig, adel, kfeedback
 [, imaxd]
 - adel — delay in seconds

More special effects in CSound:

- <http://www.csounds.com/manual/html/SigmoidSpeciale.html>

Effects may also be applied to:

- file input (diskin)
- microphone input (in)

EFFECT INSTRUMENTS IN CSOUND

Reverb or other acoustic effects are typically applied using an extra CSound instrument

- Allowing polyphony
- Similar to a conventional studio

```
garvb init 0 ; global audio-rate var, init to 0
instr 1 ;sound generation
  kfreq cpsmidib 2 ; use midi for sounds
  iamp ampmidi 0dbfs
  aout oscili iamp, kfreq
  ;out aout ;; not direct output
  garvb = garvb + aout ; add to global variable
endin
instr 99 ; freeverb instrument
  kroomsize init 0.85 ; room size
  kHFDamp init 0.5 ; high freq. damping
  arvbL,arvbR freeverb garvb, garvb, \
                                kroomsize, kHFDamp
  outs arvbL, arvbR ; output freeverb results
  clear garvb ; reset the global signal
endin
```

SPATIALIZATION

A sound source can be put at different positions of the stereo field

- Or even more channels – easily supported by CSound
- Amplitude difference, phase difference

Distribute to two channels

- `a1,a2 pan2 asig, xp [, imode]`
 - `xp` — 0=hard left, 1=hard right

*Output to two channels: **outs***

MORE FUN BY EXPERIMENTS

There are too many interesting opcodes in CSound for effects and filters

- More are added in every new version

Have fun by reading these examples

- <http://floss.booktype.pro/csound/signal-processing-i/>

QUICK HISTORY OF AUDIO EFFECTS...



Some more videos:

<https://www.waves.com/tag/sound-basics-with-stella>

Video from: <https://youtu.be/NJ2WzLg5rDc>

LECTURE REVIEW

We have discussed:

- Generating noises in CSound
- Filters in the frequency domain
- Effects in the time domain
- Spatialization

READ FURTHER

Chapter 12.1, “Source-Modifier Methods”, **CSound**.

Chapter 13, “Time-Domain Processing”, **CSound**.

Useful websites:

- <http://blog.dubspot.com/understanding-audio-effects-an-overview/>
- <http://write.flossmanuals.net/csound/signal-processing-i/>
- <http://write.flossmanuals.net/csound/signal-processing-ii/>