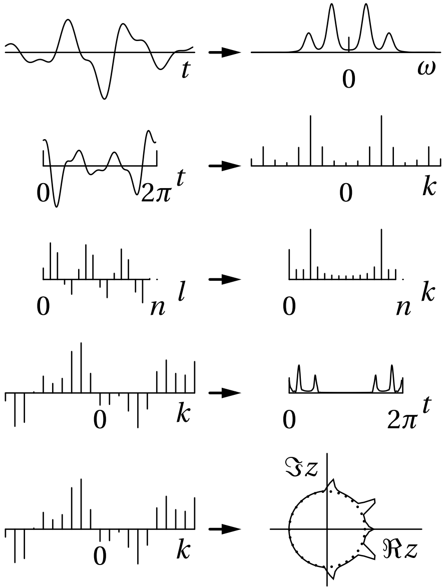
Audio Processing

1. Linear Processing (filters, equalizer, delays effects, modulation)
2. Nonlinear Processing (dynamics processing, distortion, octaver)
3. Time-Frequency Processing
   (a) Phase Vocoder Techniques
   (b) Peak Based Techniques
   (c) Linear Predictive Coding
   (d) Cepstrum
4. Time-Domain Methods
5. Spatial Effects
   (a) Sound Field Methods
   (b) Reverberation
   (c) Convolution Methods
6. Audio Coding

Introduction



continuous FT	$F(\omega) = \int_{-\infty}^{\infty} f(t)\mathrm{e}^{-\mathrm{i}\omega t}\mathrm{d}t$
Fourier series	$F[k] = \int_{0}^{2\pi} f(t)\mathrm{e}^{-\mathrm{i}kt}\mathrm{d}t$
DFT (FFT)	$F[k] = \sum_{l=0}^{n-1} f[l]\mathrm{e}^{-\mathrm{i}2\pi kl/n}$
DTFT	$F(\omega) = \sum_{k=-\infty}^{\infty} f[k]\mathrm{e}^{-\mathrm{i}\omega k}$
z-transform	$F(z) = \sum_{k=-\infty}^{\infty} f[k] z^{-k}$

Linearity of z-transform:

$$g[t] = af[t] \Rightarrow G(z) = aF(z), \quad g[t] = f_1[t] + f_2[t] \Rightarrow G(z) = F_1(z) + F_2(z).$$

Time delay of 1 $\Rightarrow$ multiplication by z^{-1}:

$$\begin{aligned} g[t] &= f[t-1] \quad \Rightarrow \\ G(z) &= \sum g[t] z^{-t} = \sum f[t-1] z^{-t} \xlongequal{s=t-1} \sum f[s] z^{-(s+1)} \\ &= z^{-1} \sum f[s] z^{-s} = z^{-1} F(z) \end{aligned}$$

FIR (finite impulse response) filters:

$$\begin{aligned} y[t] &= (h * x)[t] = h[0]x[t] + h[1]x[t-1] + \ldots + h[n]x[t-n] \quad \Rightarrow \\ Y(z) &= h[0]X(z) + h[1]z^{-1}X(z) + \ldots h[n]z^{-n}X(z) \\ &= (h[0] + h[1]z^{-1} + \ldots h[n]z^{-n})X(z) \\ &= H(z)X(z), \end{aligned}$$

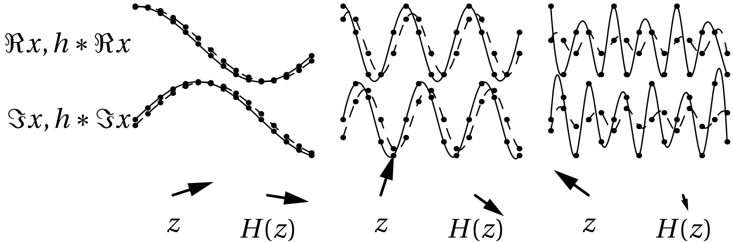
$h * x \ldots$ convolution of x and h
$H \ldots$ **transfer function**

IIR (infinite impulse response) filters:

$$\begin{aligned}
y[t] &= (h * x)[t] \\
&= h[0]x[t] + \dots h[n]x[t-n] \\
&\quad + \hat{h}[1]y[t-1] + \dots + \hat{h}[m]y[t-m] \\
y[t] - \hat{h}[1]y[t-1] - \dots - \hat{h}[m]y[t-m] &= h[0]x[t] + \dots h[n]x[t-n] \\
(1 - \hat{h}[1]z^{-1} - \dots - \hat{h}[m]z^{-m})Y(z) &= (h[0] + h[1]z^{-1} + \dots h[n]z^{-n})X(z) \\
Y(z) &= \frac{h[0] + h[1]z^{-1} + \dots h[n]z^{-n}}{1 - \hat{h}[1]z^{-1} - \dots - \hat{h}[m]z^{-m}}X(z) \\
Y(z) &= H(z)X(z)
\end{aligned}$$

(Complex) signal $x = e^{i\omega t} = z^t$, $\omega \in \{0.1\pi, 0.4\pi, 0.8\pi\}$ (solid line)

Filtering (dashed line) by the filter $h = (0.5, 0.5)$ $(H(z) = 0.5 + 0.5z^{-1}$



Assume: sampling rate 1 $\Rightarrow$ f from 0 to 0.5 (the Nyquist frequency), $\omega = 2\pi f$ from 0 to π

1 Linear Processing

control flow	signal flow
controls signal flow	controls signal
slow (every 16 to 4096 samples)	fast

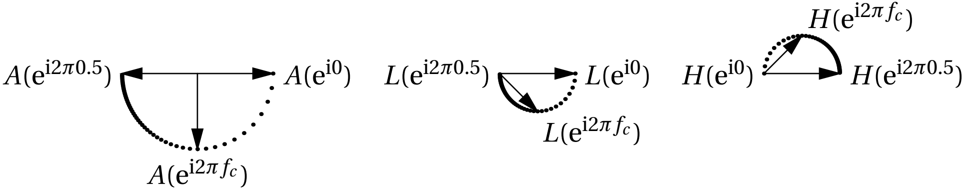
Parametric filters (easy to change properties)

Parametric allpass filter (first order):

$$y[t] = (a * x)[t] = cx[t] + x[t-1] - cy[t-1]$$

Transfer function:

$$A(z) = \frac{c + z^{-1}}{1 + cz^{-1}}$$



Transfer functions for allpass (A), lowpass (L) and highpass (H). $f_c = 0.1$

Magnitude response = 1:

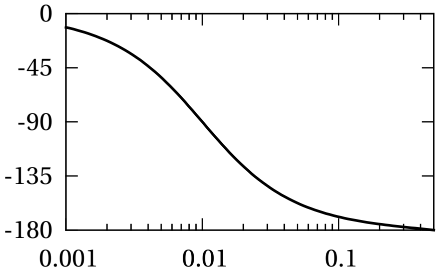
$$|A(z)| = \frac{|c + z^{-1}|}{|1 + cz^{-1}|} = \frac{|c + z^{-1}|}{|z^{-1}| \cdot |z + c|} \overset{|z|=1}{=\!=} 1$$

Phase response

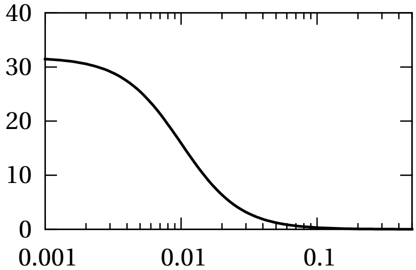
$$\varphi = \arg(A(\mathrm{e}^{\mathrm{i}\omega})) = \begin{cases} 0 & \omega = 0 \\ -90^\circ & \text{“cutoff”-frequency } \omega = 2\pi f_c,\ A(z) = A(\mathrm{e}^{-\mathrm{i}\omega}) = -\mathrm{i} \\ -180^\circ & \text{Nyquist rate } \omega = \pi \end{cases}$$

$$\begin{aligned} \frac{c + z^{-1}}{1 + cz^{-1}} &= -\mathrm{i} \\ c + z^{-1} &= -\mathrm{i} - \mathrm{i}cz^{-1} \\ c(1 + \mathrm{i}z^{-1}) &= -(\mathrm{i} + z^{-1}) \qquad \mid \cdot (1 - \mathrm{i}z) \\ c(1 + \mathrm{i}z^{-1} - \mathrm{i}z + 1) &= -(\mathrm{i} + z^{-1} + z - \mathrm{i}) \\ c(2 + 2\sin\omega) &= -2\cos\omega \\ c &= -\frac{\cos\omega}{1 + \sin\omega} = \frac{\tan(\pi f_c) - 1}{\tan(\pi f_c) + 1} \end{aligned}$$

Phase response of parametric allpass filter with $f_c = 0.01$



Phase



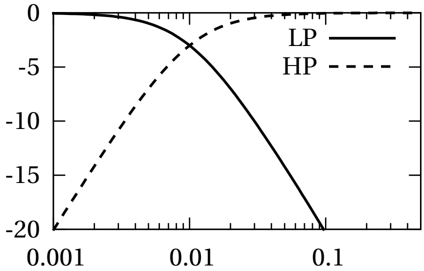
Group delay in samples

Parametric lowpass:

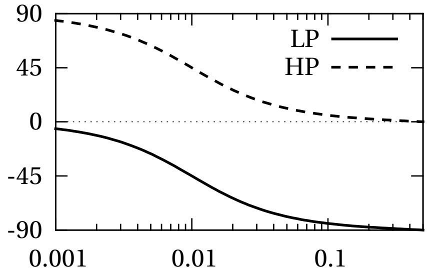
$$y = l * x = \frac{x + a * x}{2}, \qquad L(z) = \frac{1 + A(z)}{2}$$

Parametric highpass: substitute $-$ for $+$, i.e. $h * x = \frac{x - a * x}{2}$

Response of parametric lowpass and highpass filters with $f_c = 0.01$:



Magnitude response in dB



Phase

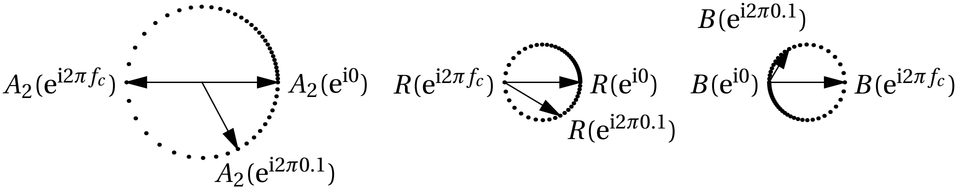
Second-order allpass filter:

$$y[t] = (a_2 * x)[t] = -d\,x[t] + c(1-d)\,x[t-1] + x[t-2] - c(1-d)\,y[t-1] + d\,y[t-2]$$

Transfer function:

$$A_2(z) = \frac{-d + c(1-d)z^{-1} + z^{-2}}{1 + c(1-d)z^{-1} - dz^{-2}}$$

Transfer functions for second-order allpass (A_2), band-reject (R) and band-pass (B) filters for $f_c = 0.2$ and $f_d = 0.15$:



Magnitude response = 1:

$$|A_2(z)| = \frac{|-d + c(1-d)z^{-1} + z^{-2}|}{|1 + c(1-d)z^{-1} - dz^{-2}|} = \frac{|-d + c(1-d)z^{-1} + z^{-2}|}{|z^{-2}| \cdot |-d + c(1-d)z + z^2|} \overset{|z|=1}{=\!=\!=} 1\,.$$

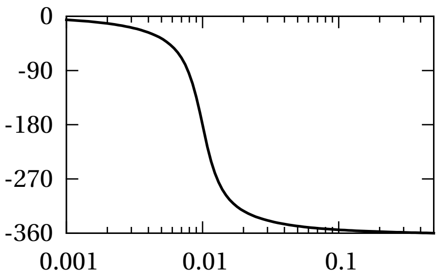
Phase $-180°$ at $\omega = \frac{f_c}{2\pi}$: $A_2(z) = A_2(\mathrm{e}^{\mathrm{i}\omega}) = -1 \Rightarrow$

$$c = -\cos\omega = -\cos 2\pi f_c$$

Parameter d controls the slope:

$$d = \frac{\tan(\pi f_d) - 1}{\tan(\pi f_d) + 1}$$

Phase response of second-order allpass filter for $f_c = 0.01$ and $f_d = 0.005$:



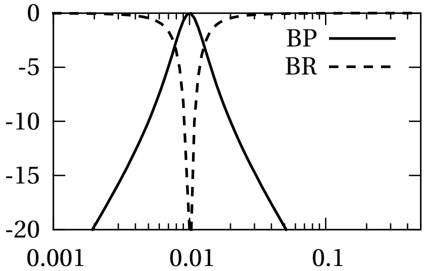
Second-order bandpass filter:

$$y = b * x = \frac{x - a_2 * x}{2}, \qquad B(z) = \frac{1 - A_2(z)}{2}$$

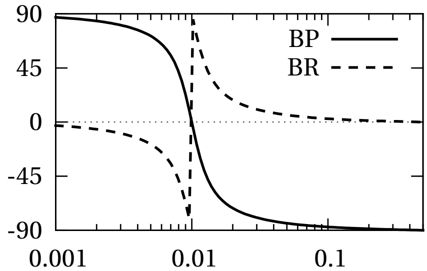
Second-order bandreject filter:

$$y = r * x = \frac{x + a_2 * x}{2}, \qquad R(z) = \frac{1 + A_2(z)}{2}$$

Response of parametric second-order bandpass and bandreject filters with $f_c = 0.01$ and $f_d = 0.005$



Magnitude response in dB



Phase

Second-order lowpass filter ($K = \tan \pi f_c$):

$$y[t] = (l_2 * x)[t] = \frac{1}{1+\sqrt{2}K+K^2}(K^2 x[t] + 2K^2 x[t-1] + K^2 x[t-2]$$
$$- 2(K^2-1)y[t-1] - (1-\sqrt{2}K+K^2)y[t-2])$$

Second-order highpass filter:

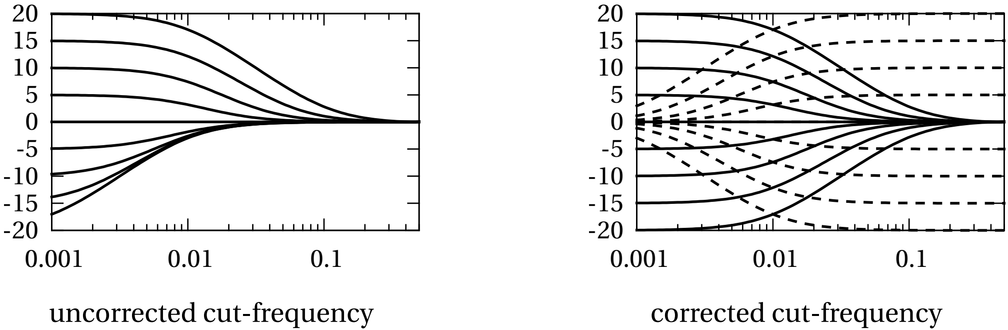
$$y[t] = (h_2 * x)[t] = \frac{1}{1+\sqrt{2}K+K^2}(x[t] - 2x[t-1] + x[t-2]$$
$$- 2(K^2-1)y[t-1] - (1-\sqrt{2}K+K^2)y[t-2])$$

Shelving filters: add low-/high-pass to original signal.

$$s_l * x = x + (v-1)\, l * x, \qquad s_h * x = x + (v-1)\, h * x,$$

$v \ldots$ amplitude factor for the passband
Gain in dB $V \quad \Rightarrow v = 10^{V/20}$

Magnitude response of low-frequency and high-frequency shelving filters for gain from −20dB to +20dB and $f_c = 0.01$



Correction to make this symmetrical for $v < 1$:

$$c = \frac{\tan(\pi f_c) - v}{\tan(\pi f_c) + v}, \qquad c = \frac{v \tan(\pi f_c) - 1}{v \tan(\pi f_c) + 1}$$

for the low-frequency and the high-frequency filter, respectively.

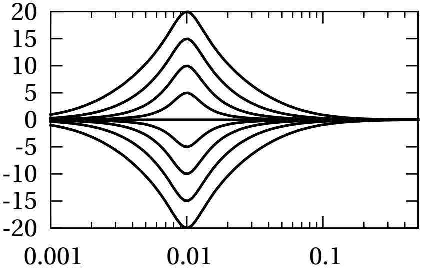
Peak filter:

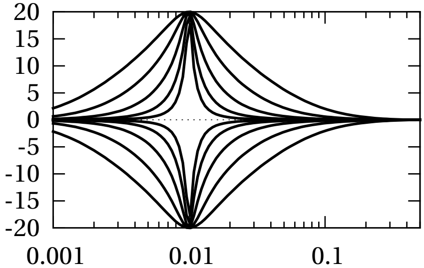
$$p * x = x + (v - 1)b * x$$

Similar correction for $v < 1$:

$$d = \frac{\tan(\pi f_d) - v}{\tan(\pi f_d) + v}$$

Magnitude response of peak filters for $f_c = 0.01$:





varying gain, $f_d = 0.005$ varying bandwidth $f_d = 0.0005, 0.001, 0.002, 0.004, 0.008$

Equalizer:

$$e = s_l(f_{cl}, V_l) * p(f_{c1}, f_{d1}, V_1) * \cdots * p(f_{cn}, f_{dn}, V_n) * s_h(f_{ch}, V_h)$$

Phaser: set of second-order bandreject filters with independently varying center frequencies
Implemented by a cascade of second-order allpass filters that are mixed with the original signal

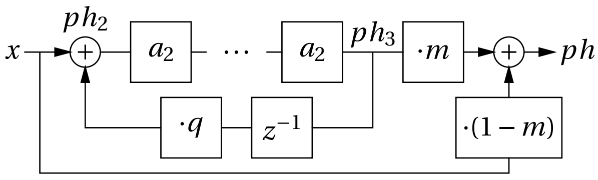
$$ph * x = (1-m)x + m \cdot a_2^{(n)} * \cdots * a_2^{(2)} * a_2^{(1)} * x$$

Extension: feedback loop

$$ph_3 * x = a_2^{(n)} * \cdots * a_2^{(2)} * a_2^{(1)} * ph_2 * x\,,$$

$$(ph_2 * x)[t] = x[t] + q \cdot (ph_3 * x)[t-1]\,,$$

$$ph * x = (1-m)x + m \cdot ph_3 * x\,.$$

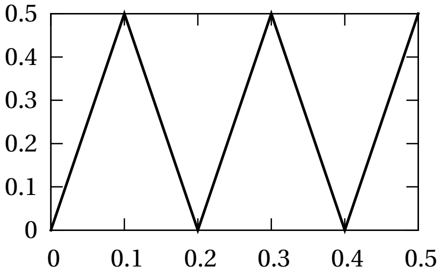


Wah-Wah effect: set of peak filters with varying center frequencies
Implemented with a single peak filter with m-tap delay ($W(z) = P(z^m)$)

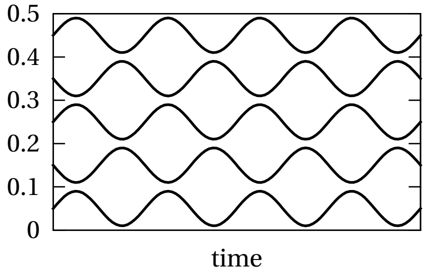
Because

$$|H(e^{i(-\omega)})| = |H(\bar{z})| = |\overline{H(z)}| = |H(z)| = |H(e^{i\omega})|,$$

and because $e^{i\omega} = e^{i(\omega \pm 2\pi)} \quad \Rightarrow \quad$ map $m\omega$ to $[0, \pi]$
$\Rightarrow$ Frequency mapping $f \mapsto g(f)$ so that $|P(e^{i2\pi m f})| = |P(e^{i2\pi g(f)})|$.



$f \mapsto g(f)$, $m = 5$



Peak frequencies, $m = 5$, controlled by LFO

Constant Q-factor: $q = \frac{f_d}{f_c} \quad \Rightarrow \quad f_d = q f_c$

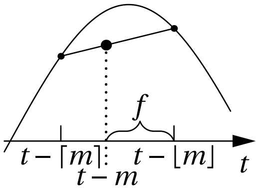
Delay effects: m-tap delay, optional mix with direct signal, optional (IIR-)feedback

Example:
vibrato effect: time-shift m varied according to a low-frequency oscillator (LFO) between 0 and 3 ms

Integer m not fine grained enough $\Rightarrow$ **fractional delays**

1. **linear interpolation**

$$y[t] = (1-f)\,x[t-\lfloor m \rfloor] + f\,x[t-\lceil m \rceil] \qquad f = m - \lfloor m \rfloor$$

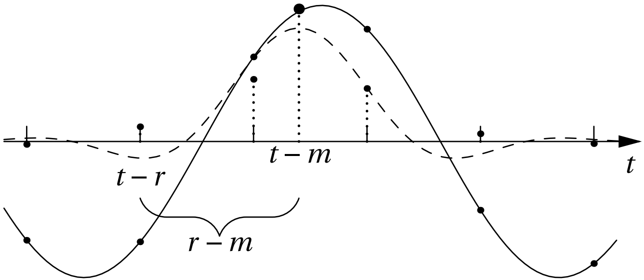


2. correct way: **sinc interpolation**

$$x(s) = \sum_{t=-\infty}^{\infty} x[t]\,\mathrm{sinc}(s-t)\,, \qquad \mathrm{sinc}(s) = \frac{\sin \pi s}{\pi s}$$

3. finite approximation: **Lanczos kernel**

$$y[t] = \sum_{|r-m|<a} x[t-r]L(r-m) \qquad L(s) = \begin{cases} \operatorname{sinc}(s)\operatorname{sinc}(\frac{s}{a}) & -a < x < a \\ 0 & \text{else} \end{cases}$$

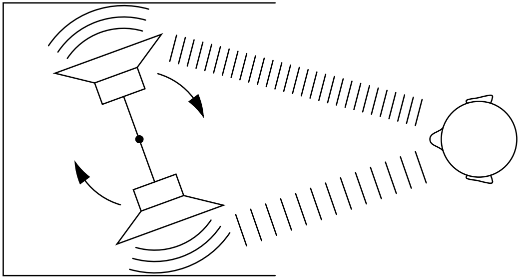


4. **allpass interpolation**

$$y[t] = (1-f)x[t-\lfloor m\rfloor] + x[t-\lceil m\rceil] - (1-f)y[t-1]$$

5. spline interpolation

Rotary speaker



$$y[t] = l(1+\sin\beta t)x[t-a(1-\sin\beta t)] + r(1-\sin\beta t)x[t-a(1+\sin\beta t)]$$

β ... rotation speed of the speakers
a ... depth of the pitch modulation
l, r ... amplitudes of the two speakers
Stereo effect: l and r unequal but symmetrical values for the left and right channel
e.g. y_l with $l = 0.7, r = 0.5$, y_r with $l = 0.5, r = 0.7$.

Comb filter: delayed signal mixed with direct signal

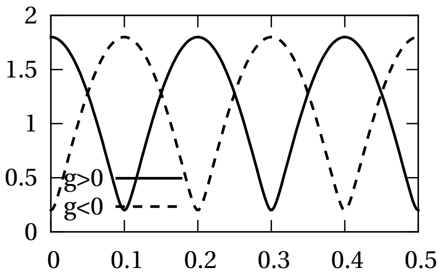
FIR comb filter:

$$y[t] = (c * x)[t] = x[t] + g\,x[t-m]\,, \qquad C(z) = 1 + g z^{-m}\,,$$

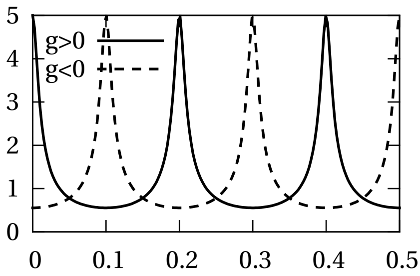
IIR comb filter:

$$y[t] = (c * x)[t] = x[t] + g\,y[t-m]\,, \qquad C(z) = \frac{1}{1 - g z^{-m}}\,,$$

Magnitude response with $m = 5$ for $g = 0.8$ and $g = -0.8$:



FIR comb filter



IIR comb filter

Problem: very high gain possible for IIR comb filter. Solution:

- retain L^∞-norm (max): multiply output by $1 - |g|$
- unmodified loudness for broadband signals: retain L^2-norm: multiply by $\sqrt{1-g^2}$.

Audio effects with delay filters:

- **slapback** effect: FIR comb filter with a delay of 10 to 25 ms (1950's rock'n'roll)
- **echo**: delays over 50 ms
- **flanger** effect: delays less than 15 ms, varied by a low-frequency oscillator (LFO)
- **chorus** effect: mixing several delayed signals with direct signal, delays independently and randomly varied with LFOs

All effects also possible with IIR comb filters.

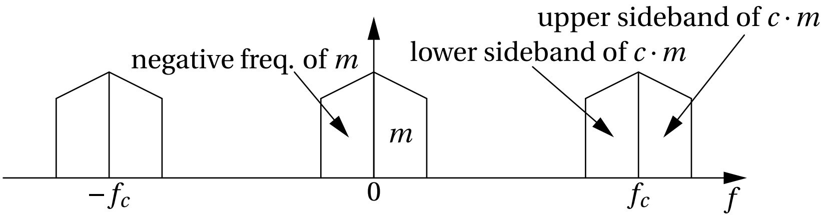
Ring modulator: multiplies a carrier signal $c[t]$ and a modulator signal $m[t]$

Complex signals: if $c[t] = \mathrm{e}^{\mathrm{i}\omega_c t}$ and $m[t] = \mathrm{e}^{\mathrm{i}\omega_m t}$, then

$$c[t]\,m[t] = \mathrm{e}^{\mathrm{i}\omega_c t}\mathrm{e}^{\mathrm{i}\omega_m t} = \mathrm{e}^{\mathrm{i}(\omega_c+\omega_m)t}$$

Real signals: mirrored negative frequencies included: $\cos x = \frac{1}{2}(\mathrm{e}^{\mathrm{i}x} + \mathrm{e}^{-\mathrm{i}x})$.
For $c[t] = \cos\omega_c t$ and $m[t] = \cos\omega_m t$:

$$\begin{aligned} c[t]\,m[t] &= \frac{1}{2}\left(\mathrm{e}^{\mathrm{i}\omega_c t} + \mathrm{e}^{-\mathrm{i}\omega_c t}\right)\frac{1}{2}\left(\mathrm{e}^{\mathrm{i}\omega_m t} + \mathrm{e}^{-\mathrm{i}\omega_m t}\right) \\ &= \frac{1}{4}\left(\mathrm{e}^{\mathrm{i}(\omega_c+\omega_m)t} + \mathrm{e}^{-\mathrm{i}(\omega_c+\omega_m)t} + \mathrm{e}^{\mathrm{i}(\omega_c-\omega_m)t} + \mathrm{e}^{-\mathrm{i}(\omega_c-\omega_m)t}\right) \\ &= \frac{1}{2}\left(\cos(\omega_c+\omega_m)t + \cos(\omega_c-\omega_m)t\right). \end{aligned}$$



Amplitude modulation: reversed roles of c and m $\Rightarrow$ **tremolo** effect

$$y[t] = (1 + \alpha m[t])x[t]$$

Getting rid of lower sideband: Reconstruct imaginary part by 90° phase shift filter

$\cos\omega t$ should become

$$\cos\left(\omega t-\frac{\pi}{2}\right)=\frac{1}{2}\left(e^{i\left(\omega t-\frac{\pi}{2}\right)}+e^{-i\left(\omega t-\frac{\pi}{2}\right)}\right)=\frac{1}{2}\left(-ie^{i\omega t}+ie^{-i\omega t}\right).$$

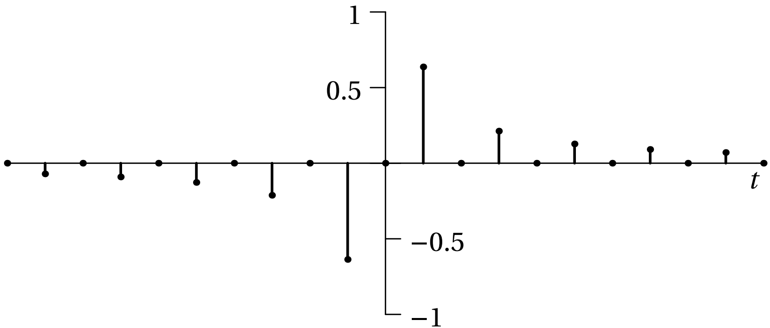
$\Rightarrow$ transfer function of the filter should be

$$H(e^{i\omega})=\begin{cases}-i & \omega>0\\ i & \omega<0.\end{cases}$$

= **Hilbert filter**

Inverse z-transform $\Rightarrow$ impulse response:

$$h[t] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(\mathrm{e}^{\mathrm{i}\omega})\mathrm{e}^{\mathrm{i}\omega t}\mathrm{d}\omega = \frac{1}{2\pi}\left(\int_{-\pi}^{0} \mathrm{i}\mathrm{e}^{\mathrm{i}\omega t}\mathrm{d}\omega - \int_{0}^{\pi} \mathrm{i}\mathrm{e}^{\mathrm{i}\omega t}\mathrm{d}\omega\right)$$
$$= \frac{1}{2\pi}\left(\mathrm{i}\frac{\mathrm{e}^{\mathrm{i}\omega t}}{\mathrm{i}t}\bigg|_{-\pi}^{0} - \mathrm{i}\frac{\mathrm{e}^{\mathrm{i}\omega t}}{\mathrm{i}t}\bigg|_{0}^{\pi}\right) = \frac{1}{2\pi t}\begin{cases} 1+1+1+1 & t \text{ odd} \\ 1-1-1+1 & t \text{ even}\end{cases}$$
$$= \begin{cases} \frac{2}{\pi t} & t \text{ odd} \\ 0 & t \text{ even}\end{cases}$$



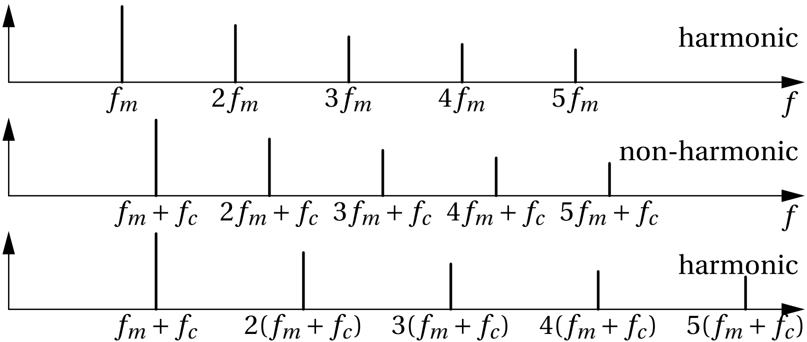
We write $\hat{x} = h * x$.

Analytic version (without negative frequencies) of c and m: $c + \mathrm{i}\hat{c}$, $m + \mathrm{i}\hat{m}$. $\Rightarrow$

$$(c + \mathrm{i}\hat{c})(m + \mathrm{i}\hat{m}) = cm - \hat{c}\hat{m} + \mathrm{i}(c\hat{m} + \hat{c}m)$$

Real part = **single sideband** modulated signal: $cm - \hat{c}\hat{m}$.

Attention: frequency shifts lead to non-harmonic sounds:



2 Nonlinear Processing

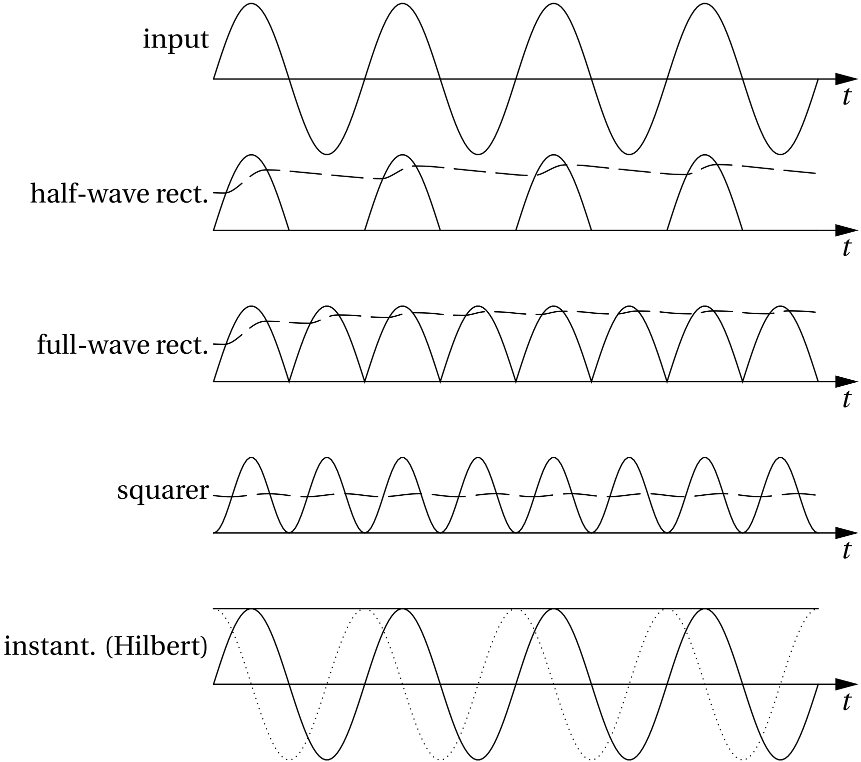
- Linear processing: $y = h * x$
- Nonlinear processing: $y = g(x)$
  - example: $y[t] = (x[t])^2$
  - example: $y[t] = (x[t])^2 + x[t-1] \cdot x[t-2]$
  - example: $y = x(l * x^2)$, low f_c
    $\Rightarrow$ slow amplitude manipulation (dynamics processing)

Dynamics processing

First step: **amplitude follower** comprised of **detector** and **averager**

Detector:

- half-wave **rectifier**: $d(x)[t] = \max(0, x[t])$.
- full-wave rectifier: $d(x)[t] = |x[t]|$.
- squarer: $d(x)[t] = x^2[t]$.
- instantaneous envelope (Hilbert transform) $d(x)[t] = x^2[t] + \hat{x}^2[t]$.


input
t
half-wave rect.
t
full-wave rect.
t
squarer
t
instant. (Hilbert)
t

Averager:

$$y[t] = a(x)[t] = (1-g)x[t] + g\,y[t-1]\,, \qquad \text{where} \qquad g = \mathrm{e}^{-\frac{1}{\tau}}$$

τ … attack and release time constant in samples.

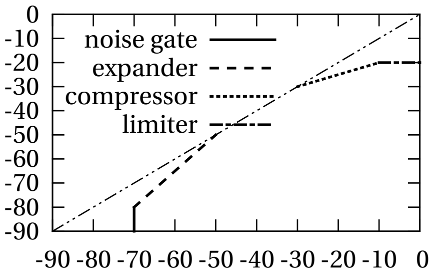
Shorter attack than release times:

$$y[t] = a(x)[t] = \begin{cases} (1-g_a)x[t] + g_a y[t-1] & y[t-1] < x[t] \\ (1-g_r)x[t] + g_r y[t-1] & y[t-1] \geq x[t] \end{cases}$$

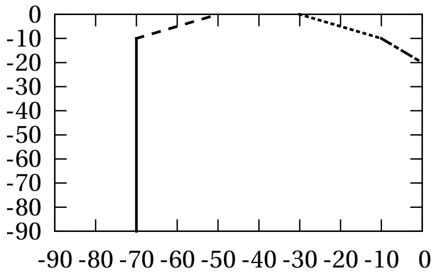
Dynamic range control:

$$y[t] = x[t-\tau] \cdot a_2(\exp(r(\log(a_1(d(x))))))[t]$$

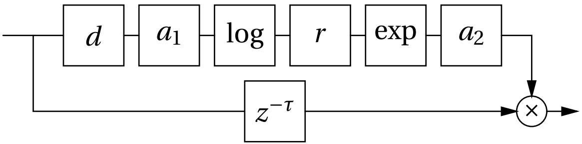
Levels and factors in dB, maximum level is 0 dB:



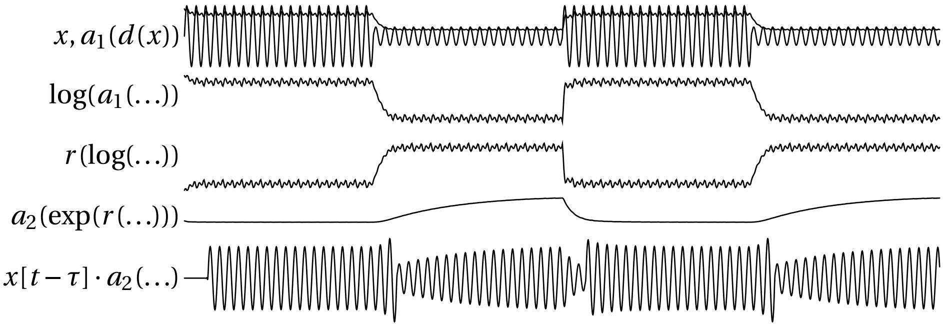
Output level over input level



Gain factor r over input level



Operator chain for dynamics processing



Compressor/limiter

- **compressor** reduces the amplitude of loud signals
- **expander** does the opposite
- **noise gate** entirely eliminates signals below a threshold
- **limiter** reduces peaks in the audio signal (rectifier as detector)
- **infinite limiter** or **clipper**: limiter with zero attack and release times: $y[t] = g(x[t])$

Typical values: $\tau_{1,a} = 5\,\mathrm{ms}$, $\tau_{1,r} = 130\,\mathrm{ms}$, $\tau_{2,a} = 1\ldots100\,\mathrm{ms}$, $\tau_{2,r} = 20\ldots5000\,\mathrm{ms}$.

$$y[t] = g(x[t])$$

Taylor expansion: $g(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots$

Impact on frequency spectrum of a single oscillation:

$$\cos^n(\omega t + \varphi) = \frac{1}{2^n} \sum_{k=0}^{n} \binom{n}{k} \cos((n-2k)(\omega t + \varphi))$$

$\Rightarrow$ new frequencies $\omega, 2\omega, 3\omega, \dots$

Total harmonic distortion:

$$\mathrm{THD} = \sqrt{\frac{A_2^2 + A_3^2 + A_4^2 + \dots}{A_1^2 + A_2^2 + A_3^2 + \dots}},$$

$A_k \dots$ amplitude of frequency $k\omega$

More than one frequency in the input signal:

$$(\cos\omega_1 t + \cos\omega_2 t)^n = \sum_{k=0}^{n} \binom{n}{k} \cos^k \omega_1 t \cos^{n-k} \omega_2 t$$

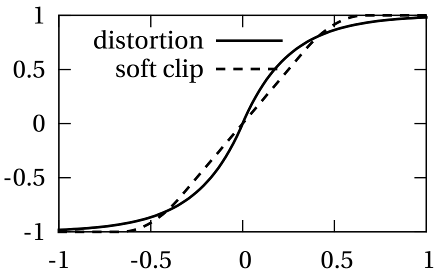
New frequencies: $a\omega_1 + b\omega_2$ for integers a and b

Soft clipping:

$$g(x) = \mathrm{sign}(x) \cdot \begin{cases} 2|x| & 0 \le |x| \le \frac{1}{3} \\ \frac{3-(2-3|x|)^2}{3} & \frac{1}{3} \le |x| \le \frac{2}{3} \\ 1 & \frac{2}{3} \le |x| \le 1 \,. \end{cases}$$

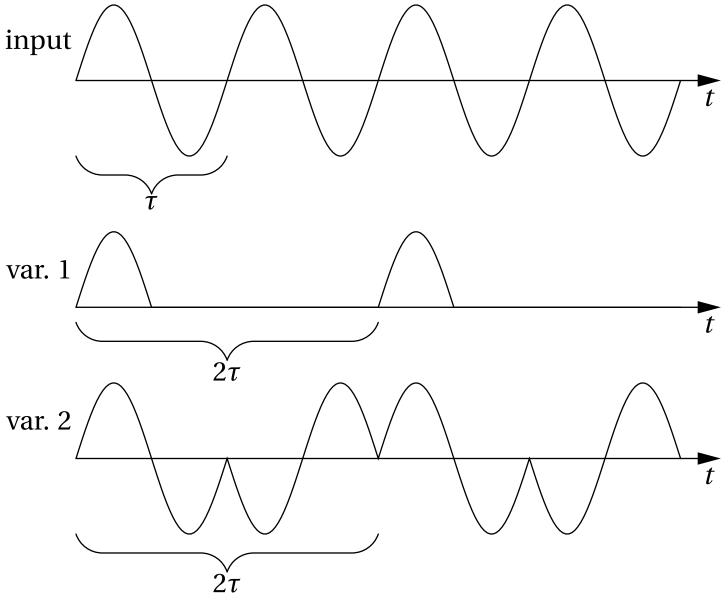
Distortion:

$$g(x) = \mathrm{sign}(x)(1 - \mathrm{e}^{-a|x|}) \qquad a \ldots \text{amount of distortion}$$



overdrive . . . small amount of distortion (“warmer” sound)
distortion . . . clearly audible distortion
fuzz . . . heavy distortion (mutual interaction between several notes results in noise)
exciter . . . light distortion to increase harmonics of a sound (brighter and clearer sound)
enhancer . . . like exciter, also uses equalization to shape the harmonic content

Octaver:
Full-wave rectifier $g(x) = |x|$ sine-wave with wave-length τ into a $\frac{\tau}{2}$-periodic signal
$\Rightarrow$ upwards octave shift
Downwards octave shift:

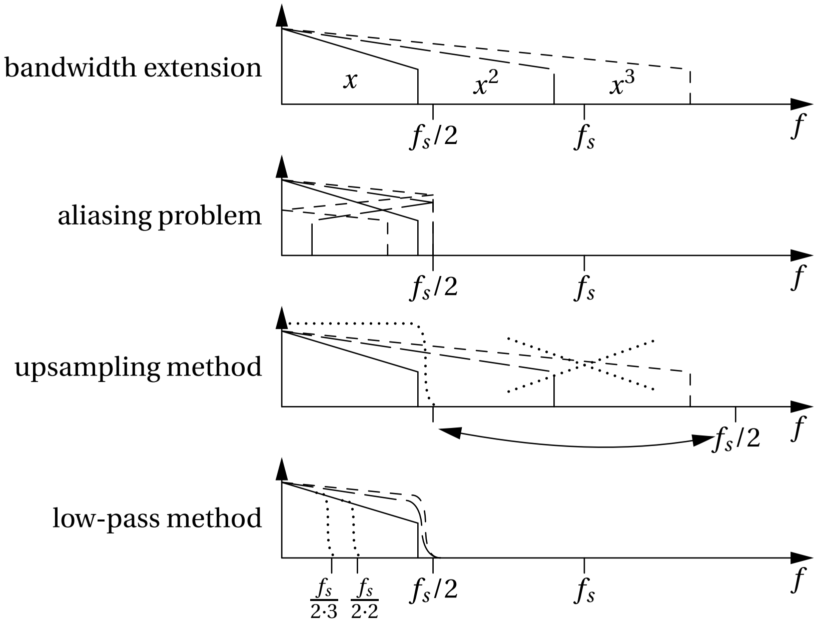


Problem: Distortion $\Rightarrow$ bandwidth extension ($x^n \Rightarrow n\omega$) $\Rightarrow$ aliasing

Solution 1: upsample signal by n using interpolation $\Rightarrow$ new frequencies from distortion are below the new Nyquist-frequency, afterwards down-sampling (with low-pass filtering)

Solution 2: split $g(x)$ into $a_1x + a_2x^2 + a_3x^3 + \ldots$, split x into n channels, each low-pass filtered by l_k with a cutoff frequency of $\frac{f_s}{2k}$

$$y[t] = a_1x + a_2(l_2 * x)^2 + a_3(l_3 * x)^3 + \ldots$$



bandwidth extension
aliasing problem
upsampling method
low-pass method
x
x²
x³
fs/2
fs
f
fs/(2·3)
fs/(2·2)

3 Time-Frequency Processing

Sinusoidal+residual model:

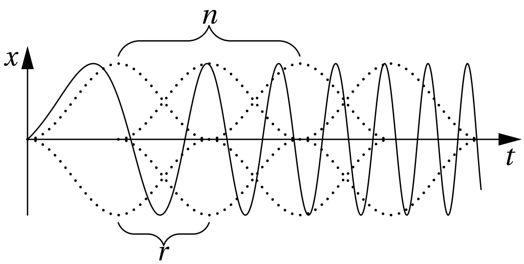
$$x[t] = \sum_k a_k[t] \cos(\varphi_k[t]) + e[t] \,.$$

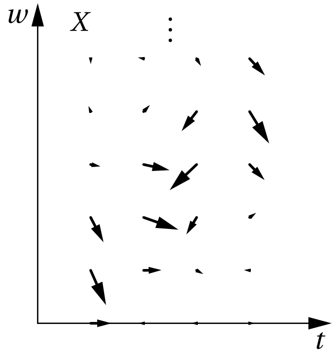
$a_k[t]$ ... amplitude of the k-th sinusoid
$e[t]$ ... residual signal
$\varphi_k[t]$ ... instantaneous phase of the k-th, which cumulates the instantaneous frequency $\omega_k[t]$:

$$\varphi_k[t] = \sum_{s=0}^{t} \omega_k[s] \,.$$

3.1 Phase Vocoder Techniques

Short-time Fourier transform (**STFT**):





n ... window or frame size

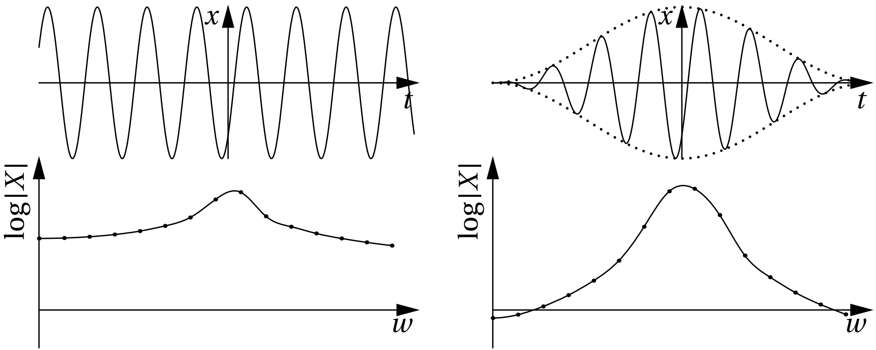
small window ⇔ bad frequency resolution

large window ⇔ bad time resolution and higher latency

r ... **hop size** (distance between the centers of consecutive windows)

overlap: $1 - r/n$

Windowing ($h[t]$):



STFT:

$$X[t,w] = \sum_{s=-n/2}^{n/2-1} h[s]x[rt+s]e^{-i2\pi ws/n}$$

w … frequency bands/bins (integer, as opposed to ω)
t … coarser time-resolution ($t+1$ means time shift of r)
$X[t,w] = |X[t,w]|e^{i\varphi[t,w]}$ … amplitude $|X[t,w]|$, phase $\varphi[t,w]$

Re-synthesis (inverse Fourier-transform, overlap-add method):

$$x[t] = \sum_{s:-\frac{n}{2}\le t-rs<\frac{n}{2}} h_s[t-rs]\sum_w X[s,w]e^{i2\pi w(t-rs)}$$

h_s … synthesis window:
    reverses analysis window h
    in overlap regions the sum of the resulting windows has to be 1 (summing condition):

$$\sum_s h[t-rs]h_s[t-rs] = 1$$

Example: Hann window

$h[t] = \frac{A}{2}(1 + \cos 2\pi t/n)$, hop size $r = n/4$

$h_s = h \quad \Rightarrow \quad \sum_s (h[t - rs])^2 = 1$

$$h^2[t] + h^2[t-n/4] + h^2[t-n/2] + h^2[t-3n/4]$$

$$= \frac{A^2}{4}(1+\cos 2\pi t/n)^2 + \frac{A^2}{4}(1+\cos 2\pi(t/n-1/2))^2 + \ldots + \frac{A^2}{4}(1+\cos 2\pi(t/n-3/4))^2$$

$$= \frac{A^2}{4}(1+\cos)^2 + \frac{A^2}{4}(1-\sin)^2 + \frac{A^2}{4}(1-\cos)^2 + \frac{A^2}{4}(1+\sin)^2$$

$$= \frac{A^2}{4}(1+2\cos+\cos^2+1-2\sin+\sin^2+1-2\cos+\cos^2+1+2\sin+\sin^2)$$

$$= \frac{A^2}{4}(4+2(\cos^2+\sin^2)) = \frac{3A^2}{2} \overset{A=\sqrt{2/3}}{=\!=\!=\!=} 1$$

Phase vocoder = STFT + modifications + inverse STFT

Time stretching: use a different hop size r_s for synthesis
Problem: phases do not match
Solution: **phase unwrapping**:
$\varphi[t, w]$ … instantaneous phase of $X[t, w]$, so that

$$X[t, w] = A[t, w]\mathrm{e}^{\mathrm{i}\varphi[t,w]}$$

If frequency would be exactly w, then the projected phase of $X[t+1, w]$ is

$$\varphi_p[t+1, w] = \varphi[t, w] + 2\pi w r/n \quad \overset{\bmod 2\pi}{=} \quad \varphi[t+1, w]$$

Otherwise: unwrapped phase $\varphi_u[t+1, w]$:

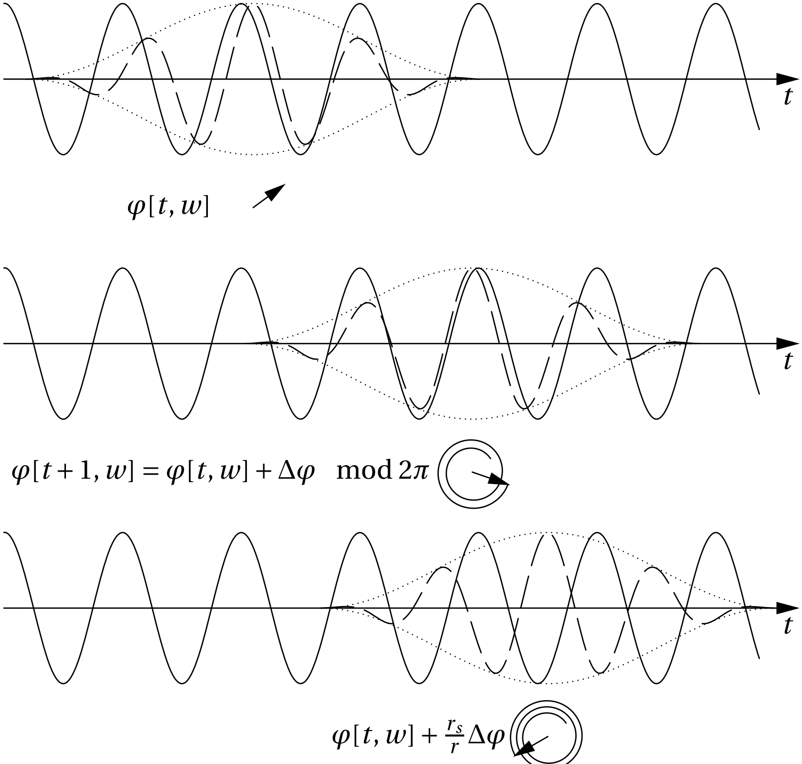
$$\varphi_u[t+1, w] = \varphi[t+1, w] \mod 2\pi, \qquad -\pi \le \varphi_u[t+1, w] - \varphi_p[t+1, w] \le \pi$$

This can be achieved by

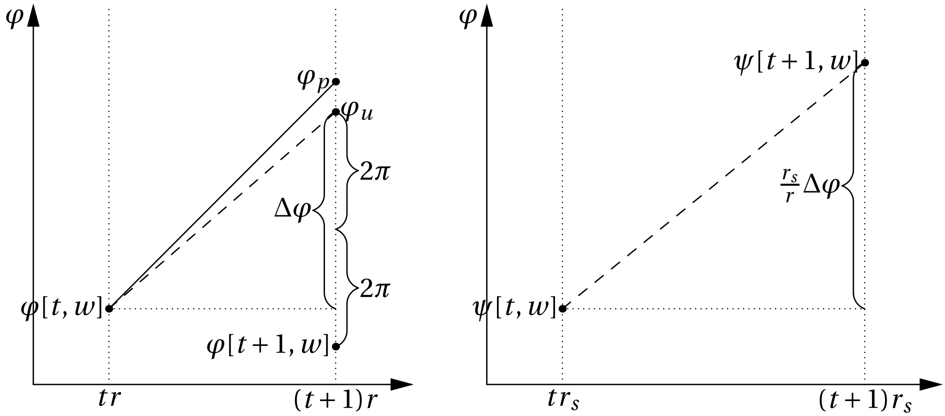
$$\varphi_u[t+1, w] = \varphi[t+1, w] + \mathrm{round}((\varphi_p[t+1, w] - \varphi[t+1, w])/2\pi) \cdot 2\pi$$

Total phase rotation between t and $t+1$ in frequency bin w:

$$\Delta\varphi[t+1, w] = \varphi_u[t+1, w] - \varphi[t, w]$$



t
φ[t, w]
t
φ[t + 1, w] = φ[t, w] + Δφ mod 2π
t
φ[t, w] + (r_s / r) Δφ



φ
φ_p
φ_u
2π
$\Delta\varphi$
2π
$\varphi[t,w]$
$\varphi[t+1,w]$
tr
$(t+1)r$
φ
$\psi[t+1,w]$
$\frac{r_s}{r}\Delta\varphi$
$\psi[t,w]$
tr_s
$(t+1)r_s$

Time stretching, finally:

$$Y[t,w] = \sum_{s=-n/2}^{n/2-1} h[s]\,y[r_s t + s]\,\mathrm{e}^{-\mathrm{i}2\pi w s/n} = A[t,w]\,\mathrm{e}^{\mathrm{i}\psi[t,w]}$$

$$\psi[t+1,w] = \psi[t,w] + \frac{r_s}{r}\Delta\varphi[t+1,w]$$

Pitch shifting by time stretching ($r_s = \alpha r$): resampling after time stretching $y[t] = x[\alpha t]$

Problem: Frequency transients and consonants are smeared in time.
Solution: Separate stable from transient components (stable = unchanging phase change):

$$\varphi[t,w] - \varphi[t-1,w] \approx \varphi[t-1,w] - \varphi[t-2,w] \mod 2\pi$$

More precisely:

$$|\varphi[t,w] - 2\varphi[t-1,w] + \varphi[t-2,w]| < d \mod 2\pi$$

where "$|x| < d \mod 2\pi$" means: the smallest $|x + k \cdot 2\pi|$ is smaller than d.

Stable frequency bins: time stretching
Transient bins: drop or use to construct residual signal
Or: do not stretch parts without stable bins

Mutation (**morphing**, **cross-synthesis**, vocoder effect): Use phase of X_1 and magnitude of X_2:

$$Y[t,w] = \frac{X_1[t,w]}{|X_1[t,w]|}|X_2[t,w]|$$

Robotization: Set all phases to zero in each frame and each bin.

Whisperization: randomize the phase

Denoising: attenuate frequency bins with low magnitude, keep high magnitudes unchanged.

$$Y[t,w] = X[t,w]\frac{|X[t,w]|}{|X[t,w]|+c_w}$$

$c_w \ldots$ controls amount and level of attenuation.

3.2 Peak Based Techniques

- Phase vocoder: represent frequency by frequency bin and phase (bin-number only exact up to f_s/N)
- Peak based: represent frequency by exact peak

Peak detection: fit a parabola to the maximum and the two neighboring bins (in logarithmic representation of the magnitudes)

$a_w = 10\log_{10}|X[t, w_0 + w]|_2^2 \quad (w_0 \ldots$ bin of local maximum$)$

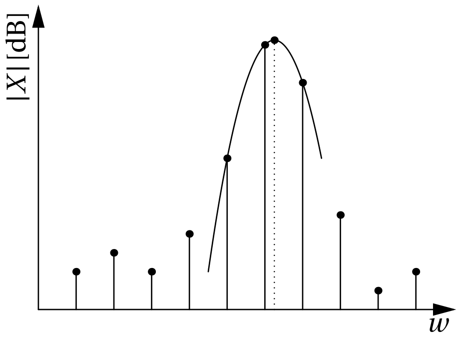
Parabola $p(w) = \alpha w^2 + \beta w + \gamma$ so that $p(w) = a_w$ for $w \in \{-1, 0, 1\}$

$\Rightarrow \alpha - \beta + \gamma = a_{-1}, \gamma = a_0, \alpha + \beta + \gamma = a_1$

$\Rightarrow \alpha = \frac{1}{2}(a_1 - 2a_0 + a_{-1}), \beta = \frac{1}{2}(a_1 - a_{-1})$

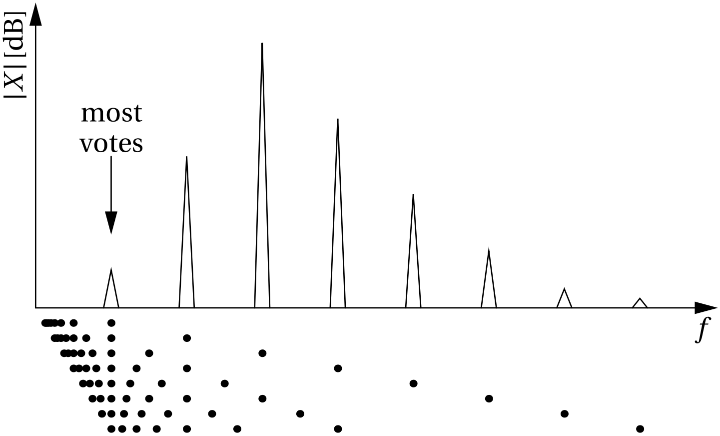
Peak of $p(w)$ where $p'(w) = 0 \quad \Rightarrow \quad 2\alpha w + \beta = 0 \quad \Rightarrow$

$$w = -\frac{\beta}{2\alpha} = \frac{a_{-1} - a_1}{2(a_{-1} - 2a_0 + a_1)}.$$



Pitch detection: find the fundamental frequency (integer multiples: harmonics/partials)

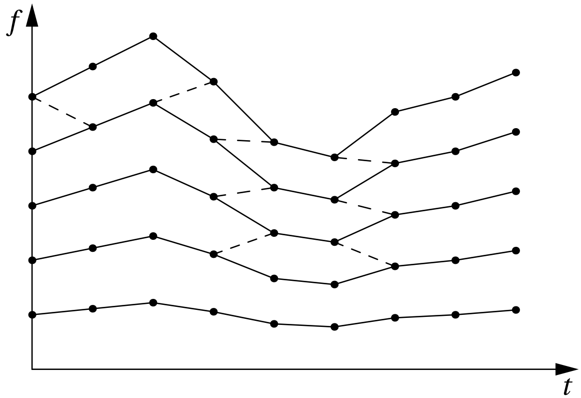
Heuristics: Each peak casts a (weighted) vote to itself and its integer fractions:



Peak continuation: associate corresponding peaks of subsequent frames

Simple way: choose peak that is closest in frequency (may be wrong in case of transients)

Better way: “guides” – updated to match peaks and fundamental frequency – can be created, killed, turned on/off temporarily



Convert *tracks* representation back to sound (**synthesis**):

- oscillator
- inverse Fourier transform

Oscillator (analog, differential equation):

$$x''(t) = -ax(t)$$

Discretization:

$$x''(t) \approx x[t+1] - 2x[t] + x[t-1]$$

$\Rightarrow$ (**digital resonator**):

$$x[t+1] = (2-a)x[t] - x[t-1] =: (r * x)[t+1]$$

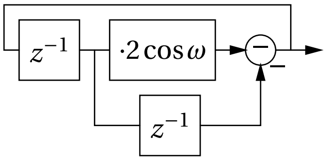
Transfer function:

$$R(z) = \frac{1}{1-(2-a)z^{-1}+z^{-2}}$$

Pole of $R(z)$ is resonance frequency (denominator = 0):

$$\begin{aligned}(2-a)z^{-1} &= 1+z^{-2}\\ (2-a) &= z+z^{-1} = 2\cos\omega\end{aligned}$$

Initialize by calculating $x[0]$ and $x[1]$ directly



Problem: changes in oscillation energy during frequency changes:

$$E[t] = ax[t]x[t-1] + (x[t] - x[t-1])^2$$

$$\begin{aligned}
E[t+1] &= ax[t+1]x[t] + (x[t+1] - x[t])^2 \\
&= a((2-a)x[t] - x[t-1])x[t] + ((2-a)x[t] - x[t-1] - x[t])^2 \\
&= a(2-a)x[t]^2 - ax[t]x[t-1] + (x[t] - x[t-1] - ax[t])^2 \\
&= a(2-a)x[t]^2 - ax[t]x[t-1] + (x[t] - x[t-1])^2 - 2ax[t](x[t] - x[t-1]) + a^2x[t]^2 \\
&= a(2-a)x[t]^2 - ax[t]x[t-1] + (x[t] - x[t-1])^2 - a(2-a)x[t]^2 + 2ax[t]x[t-1] \\
&= ax[t]x[t-1] + (x[t] - x[t-1])^2 = E[t] .
\end{aligned}$$

Frequency change ($a \mapsto a_2$):

- at signal maximum: $E[t] \approx ax[t]x[t-1] \approx ax[t]^2 \Rightarrow$ changed energy ($\cdot a_2/a$), same amplitude
- at zero crossing: $E[t] \approx (x[t] - x[t-1])^2 \Rightarrow$ same energy, changed amplitude

This has to be compensated or, better, the signal has to be initialized again.

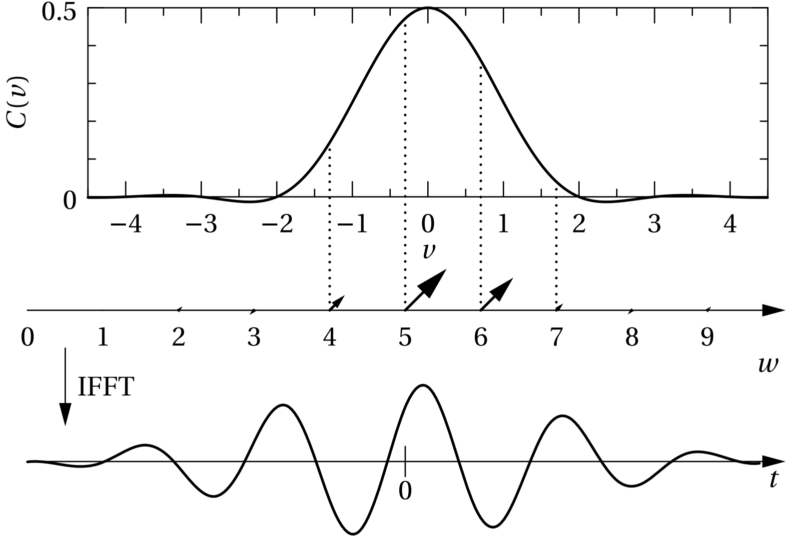
Synthesis by inverse Fourier transform: add spectral pattern of sinusoid to frequency bins
Determine coefficients by forward transform of pure sine wave. Redundancies:

- amplitudes adjusted by multiplying coefficients (consider only normed amplitude)
- phase adjusted by multiplication with $\mathrm{e}^{\mathrm{i}\varphi}$ (consider only normed phase)
- all coefficients have same phase (ignore phases)
- coefficients for two frequencies with an integer bin-distance are the same, just shifted by a certain number of bins (consider only frequencies between bin 0 and 1)
- coefficients far from the center frequency are negligibly small (consider only small number of bins)

$$C_f[w] = \sum_{s=-n/2}^{n/2-1} h[s]\mathrm{e}^{\mathrm{i}2\pi f s}\mathrm{e}^{-\mathrm{i}2\pi w s/n} = \sum_{s=-n/2}^{n/2-1} h[s]\mathrm{e}^{-\mathrm{i}2\pi(w-nf)s/n},$$

$w = -b, \ldots, b$, $b \ldots$ approximation bandwidth, $nf \in [0,1)$, or better $nf \in [-0.5, 0.5)$
Combine w and f into $v = w - nf$ $\quad\Rightarrow\quad$ zero-padded Fourier transform of window $h[s]$

$$C(v) = \sum_{s=-n/2}^{n/2-1} h[s]\mathrm{e}^{-\mathrm{i}2\pi v s/n}$$



Spectral motif $C(v)$ for Hann window, used for IFFT synthesis ($nf = 5.3$, $\varphi = \pi/4$)

Copy/add $AC(w - nf)e^{i\varphi}$ into bin w.

Performance comparison:

- one sinusoid:
  - Resonator: $O(1)$ operations per sample
  - inverse FFT: $O(n \log n)$ per frame $\quad\Rightarrow\quad$ $O(\log n)/(1-\text{overlap})$ per sample
- k sinusoids:
  - Resonator: $O(k)$
  - inverse FFT: $O(bk/n) + O(\log n)/(1-\text{overlap})$

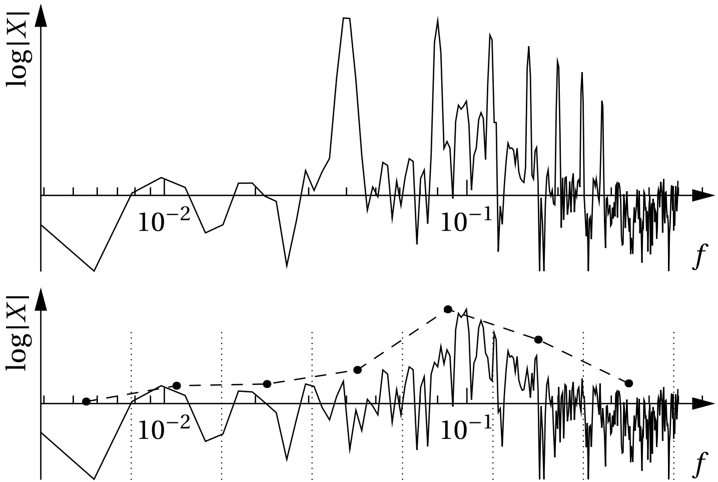
Problem with overlap-add IFFT synthesis: change in frequency $\Rightarrow$ interferences in overlaps
Possible solution: no overlap:

- inverse window $h_s[s] = h[s]^{-1}$
- truncate border (approximation errors mostly near border)
- phases must be exact (avoid phase jumps at border)

Residual signal: subtract re-synthesized signal from the original signal

- in time domain: shorter frames (time resolution more important)
- in frequency domain: no additional FFT needed

Residual signal: stochastic signal (only spectral shape important, no phase information)
Curve fitting on the magnitude spectrum (straight-line segment approximation):

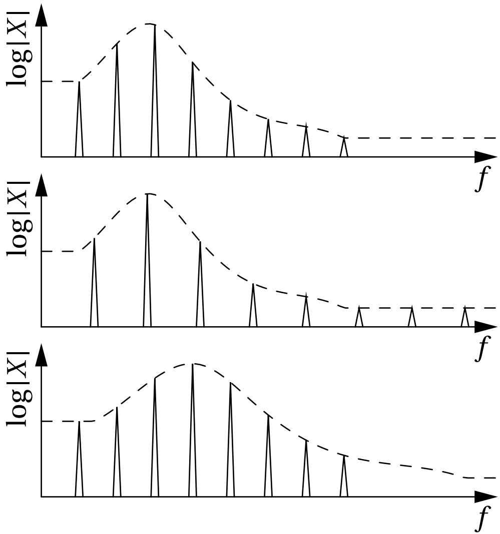


Synthesis of the residual signal:

- convolution of white noise with impulse response of the magnitude spectrum, or
- fill each frequency bin with a complex value: magnitude from the measured magnitude spectrum, random phase.

Applications of peak based methods

- filter with arbitrary resolution
- **Pitch shifting, timbre preservation**



- **Spectral shape shift**

- **Time stretching** (same hop-size but repeat/drop frames)
  avoid smoothing of attack transients: analysis and synthesis frame rates can be set equal for a short time.

- **Pitch correction** (Auto-Tune):

  - detect pitch
  - quantify towards nearest of the 12 semitones
  - sinusoids pitch-scaled by the same factor

- **Gender change**: pitch scaling, move spectral shape along with the pitch for female voice

- **Hoarseness**: increase magnitude of the residual signal

3.3 Linear Predictive Coding

Linear predictive coding (**LPC**):
Prediction filter p: $x[t] \approx (p * x)[t]$
Residual $e[t] = x[t] - (p * x)[t]$

$$(p * x)[t] = p[1]x[t-1] + p[2]x[t-2] + \ldots + p[m]x[t-m]$$

Re-synthesize: $x[t] = (p * x)[t] + e[t]$

If residual $e[t]$ not known exactly ($\tilde{e}[t]$):

$$y[t] = (p * y)[t] + \tilde{e}[t]$$

(all-pole IIR filter)

How to find optimum filter coefficients $p[k]$?
Minimize:

$$E := \sum_t e^2[t] = \sum_t (x[t] - p[1]x[t-1] - p[2]x[t-2] - \ldots - p[m]x[t-m])^2$$

Deriving this with respect to all $p[k]$, setting zero:

$$0 = \frac{\mathrm{d}E}{\mathrm{d}p[k]} = \sum_t 2e[t]\frac{\mathrm{d}e[t]}{\mathrm{d}p[k]} = 2\sum_t e[t]x[t-k] = 2\sum_t \left(x[t] - \sum_j p[j]x[t-j]\right) x[t-k]$$

$$\Leftrightarrow \quad \sum_j p[j] \sum_t x[t-j]x[t-k] = \sum_t x[t]x[t-k]$$

Involves the autocorrelation of x. More stable with windowing:

$$r_{xx}[s] := \sum_t w[t]x[t]w[t-s]x[t-s]$$

$\Rightarrow$

$$\sum_j p[j]r_{xx}[k-j] = r_{xx}[k],$$

$\Rightarrow$ equation system with Toeplitz matrix (constant diagonals $M_{k,k-i} = r_{xx}[k-(k-i)] = r_{xx}[i]$)

Levinson-Durbin recursion:
$T^{(n)}$ ... upper left $n \times n$-sub-matrix of $M_{k,j} = r_{xx}[k-j]$
$p^{(n)}$ ... solution vector of $T^{(n)} p^{(n)} = y^{(n)}$ where $y^{(n)} = r_{xx}[1 \ldots n]$

$$T^{(n+1)} \begin{pmatrix} p^{(n)} \\ 0 \end{pmatrix} = \begin{pmatrix} y^{(n)} \\ \epsilon \end{pmatrix} \tag{1}$$

ϵ should be $r_{xx}[n+1]$
Help vector $b^{(n)}$ which satisfies $T^{(n)} b^{(n)} = (0, \ldots, 0, 1)$

$$T^{(n+1)} p^{(n+1)} = T^{(n+1)} \left(\begin{pmatrix} p^{(n)} \\ 0 \end{pmatrix} + (r_{xx}[n+1] - \epsilon) b^{(n+1)} \right) = y^{(n+1)} \tag{2}$$

Find $b^{(n)}$: find also $f^{(n)}$ satisfying $T^{(n)} f^{(n)} = (1, 0, \ldots, 0)$

$$T^{(n+1)} \begin{pmatrix} f^{(n)} \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ \epsilon_f \end{pmatrix}, \qquad T^{(n+1)} \begin{pmatrix} 0 \\ b^{(n)} \end{pmatrix} = \begin{pmatrix} \epsilon_b \\ 0 \\ \vdots \\ 1 \end{pmatrix} \tag{3}$$

Find α and β so that

$$T^{(n+1)} f^{(n+1)} = T^{(n+1)} \left(\alpha \begin{pmatrix} f^{(n)} \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ b^{(n)} \end{pmatrix} \right) = \alpha \begin{pmatrix} 1 \\ 0 \\ \vdots \\ \epsilon_f \end{pmatrix} + \beta \begin{pmatrix} \epsilon_b \\ 0 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \end{pmatrix}, \tag{4}$$

which can be found by solving

$$\alpha + \beta\epsilon_b = 1, \quad \alpha\epsilon_f + \beta = 0 \qquad \Rightarrow \qquad \alpha = \frac{1}{1-\epsilon_b\epsilon_f}, \quad \beta = -\epsilon_f\alpha \tag{5}$$

Same for $b^{(n+1)}$.
For symmetric Toeplitz matrices: b is just f reversed, and $\epsilon_f = \epsilon_b$.

$\Rightarrow$ Recursion from $n+1=1$ to m (length of filter p)
Complexity: $O(m^2)$ (normal equation solving: $O(m^3)$)

Example.

$$x = (1, 2, 1, -1, -2, -1)$$

$$r_{xx}[0] = 1^2 + 2^2 + \dots, r_{xx}[1] = 1 \cdot 2 + 2 \cdot 1 + \dots, \quad r_{xx} = (12, 7, -2, -6, -4, -1)$$

To solve for $m = 3$:

$$\begin{pmatrix} 12 & 7 & -2 \\ 7 & 12 & 7 \\ -2 & 7 & 12 \end{pmatrix} \begin{pmatrix} p[1] \\ p[2] \\ p[3] \end{pmatrix} = \begin{pmatrix} 7 \\ -2 \\ -6 \end{pmatrix}$$

Iteration $n = 0$

$$p^{(1)} = (7/12) = \left(\tfrac{7}{12}\right), \quad f^{(1)} = b^{(1)} = \left(\tfrac{1}{12}\right)$$

Iteration $n = 1$

$$\epsilon_f = \epsilon_b = \frac{1}{12} \cdot 7 = \frac{7}{12} \qquad \Leftarrow (3)$$

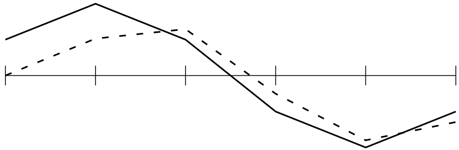
$$\alpha = \frac{1}{1 - \frac{7}{12} \cdot \frac{7}{12}} = \frac{144}{95}, \qquad \beta = -\frac{7}{12} \cdot \frac{144}{95} = -\frac{84}{95} \qquad \Leftarrow (5)$$

$$f^{(2)} = \frac{144}{95} \begin{pmatrix} \frac{1}{12} \\ 0 \end{pmatrix} + \left(-\frac{84}{95}\right) \begin{pmatrix} 0 \\ \frac{1}{12} \end{pmatrix} = \begin{pmatrix} \frac{12}{95} \\ -\frac{7}{95} \end{pmatrix}, \qquad b^{(2)} = \begin{pmatrix} -\frac{7}{95} \\ \frac{12}{95} \end{pmatrix} \qquad \Leftarrow (4)$$

$$\epsilon = \frac{7}{12} \cdot 7 = \frac{49}{12} \qquad \Leftarrow (1)$$

$$p^{(2)} = \begin{pmatrix} \frac{7}{12} \\ 0 \end{pmatrix} + \left(-2 - \frac{49}{12}\right) \begin{pmatrix} -\frac{7}{95} \\ \frac{12}{95} \end{pmatrix} = \begin{pmatrix} \frac{98}{95} \\ -\frac{73}{95} \end{pmatrix} \qquad \Leftarrow (2)$$

$x[t], (p^{(2)} * x)[t]$:



Iteration $n = 2$

$$\epsilon_f = \epsilon_b = \frac{12}{95} \cdot (-2) + \left(-\frac{7}{95}\right) \cdot 7 = -\frac{73}{95} \qquad \Leftarrow (3)$$

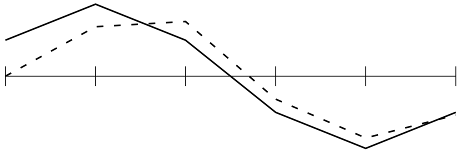
$$\alpha = \frac{1}{1 - (-\frac{73}{95}) \cdot (-\frac{73}{95})} = \frac{9025}{3696}, \qquad \beta = -\left(-\frac{73}{95}\right) \cdot \frac{9025}{3696} = \frac{6935}{3696} \qquad \Leftarrow (5)$$

$$f^{(3)} = \frac{9025}{3696} \begin{pmatrix} \frac{12}{95} \\ -\frac{7}{95} \\ 0 \end{pmatrix} + \frac{6935}{3696} \begin{pmatrix} 0 \\ -\frac{7}{95} \\ \frac{12}{95} \end{pmatrix} = \begin{pmatrix} \frac{95}{308} \\ -\frac{7}{22} \\ \frac{73}{308} \end{pmatrix}, \qquad b^{(3)} = \begin{pmatrix} \frac{73}{308} \\ -\frac{7}{22} \\ \frac{95}{308} \end{pmatrix} \qquad \Leftarrow (4)$$

$$\epsilon = \frac{98}{95} \cdot (-2) + \left(-\frac{73}{95}\right) \cdot 7 = \left(-\frac{707}{95}\right) \qquad \Leftarrow (1)$$

$$p^{(3)} = \begin{pmatrix} \frac{98}{95} \\ -\frac{73}{95} \\ 0 \end{pmatrix} + \left(-6 - \left(-\frac{707}{95}\right)\right) \begin{pmatrix} \frac{73}{308} \\ -\frac{7}{22} \\ \frac{95}{308} \end{pmatrix} = \begin{pmatrix} \frac{423}{308} \\ -\frac{27}{22} \\ \frac{137}{308} \end{pmatrix} \qquad \Leftarrow (2)$$

$x[t], (p^{(3)} * x)[t]$:



Two possibilities to apply the predictor p:

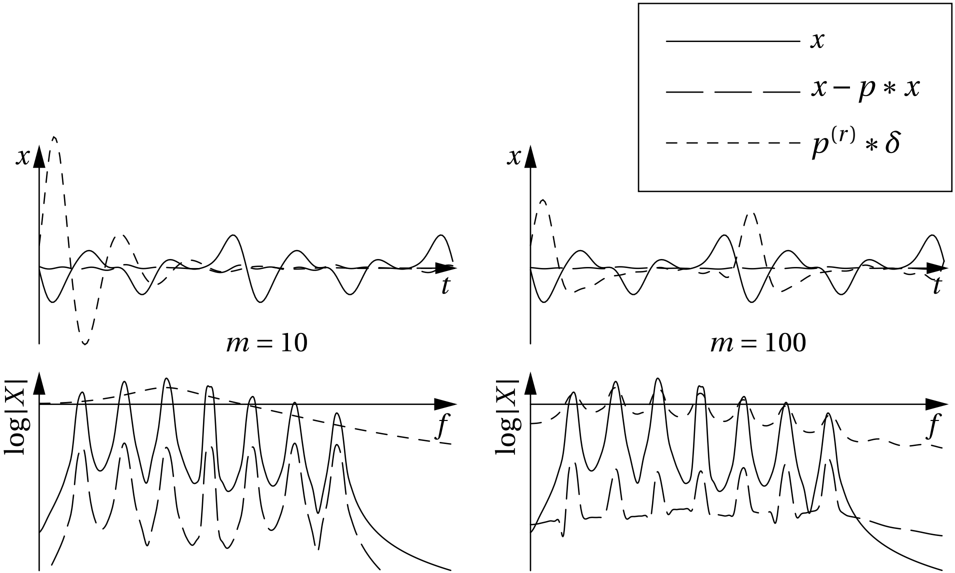
- FIR filter $p * x$
- recursive IIR filter $p^{(r)} * x$:

$$y[t] = (p^{(r)} * x)[t] := x[t] + (p * y)[t] = x[t] + p_1 y[t-1] + \ldots + p_m y[t-m]$$

$x \ldots$ "excitation" of $p^{(r)}$

Excited with the prediction residual $\quad\Rightarrow\quad$ original signal is reconstructed:

$$y = p^{(r)} * (x - p * x) = x - p * x + p * y \quad\Rightarrow\quad y - p * y = x - p * x \quad\Rightarrow\quad y = x$$



Residual is "whitened" (peaks at same level)
Predictor represents spectral shape ($p^{(r)} * \delta$)

Sound **mutation**:

$$y = p_2^{(r)} * (x_1 - p_1 * x_1) \, .$$

LPC-method widely used in speech analysis, synthesis and compression.

3.4 Cepstrum

Cepstrum (anagram of spectrum): smoothing of the magnitude spectrum by a Fourier method
real cepstrum:

$$c[t,s] := \frac{1}{n} \sum_{w=-n/2}^{n/2-1} \log|X[t,w]| \mathrm{e}^{\mathrm{i}2\pi ws/n}$$

$s \ldots$ “quefrency”

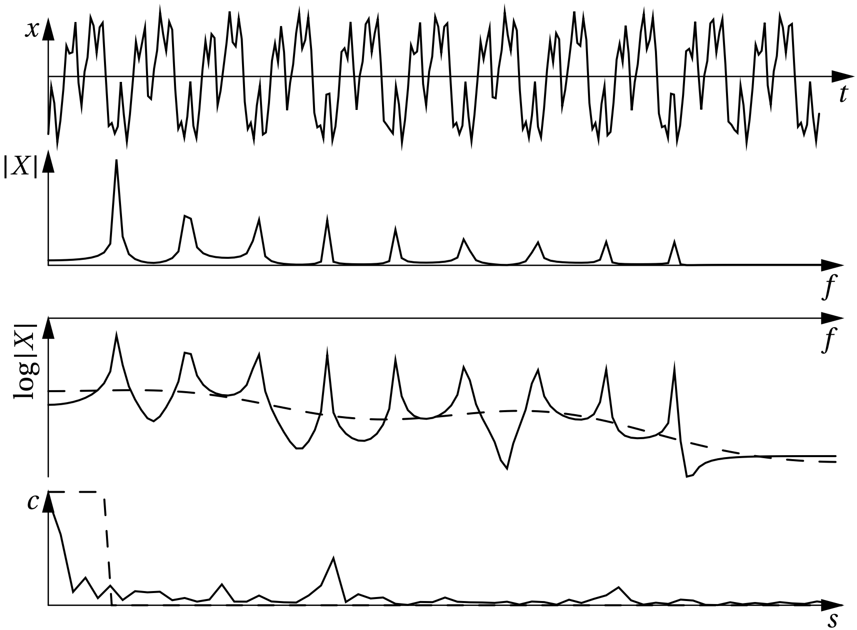
Low-pass filtering in the s-domain:

$$l[s] = \begin{cases} 1 & -s_c \leq s < s_c \\ 0 & \text{else}, \end{cases}$$

$s_c \ldots$ cutoff quefrency

Forward Fourier transform ⇒ smoothed spectrum in the logarithmic domain (dB):

$$C_l[t,w] = \sum_{s=-n/2}^{n/2-1} c[t,s]\, l[s] \mathrm{e}^{-\mathrm{i}2\pi ws/n}$$


x
t
|X|
f
log|X|
f
c
s

High-pass window $h[s] = 1 - l[s] \quad \Rightarrow \quad$ complementary source envelope

$$\log|X[t,w]| = C_l[t,w] + C_h[t,w]$$

$$X[t,w] = \exp(C_l[t,w])\exp(C_h[t,w])\mathrm{e}^{\mathrm{i}\varphi[t,w]}$$

Source-filter separation:

- $\exp(C_l[t,w])$ ... filter or spectral envelope
- $\exp(C_h[t,w])\mathrm{e}^{\mathrm{i}\varphi[t,w]}$ ... source signal

Sound **mutation** (again):

$$\begin{aligned} Y[t,w] &= \exp(C_l^{(1)}[t,w])\exp(C_h^{(2)}[t,w])\mathrm{e}^{\mathrm{i}\varphi^{(2)}[t,w]} \\ &= X^{(2)}[t,w]\exp(-C_l^{(2)}[t,w])\exp(C_l^{(1)}[t,w]) \end{aligned}$$

Formant changing:

$$\begin{aligned} Y[t,w] &= X[t,w]\exp(-C_l[t,w])\exp(C_l[t,w/k]) \\ &= X[t,w]\exp(C_l[t,w/k] - C_l[t,w]) \end{aligned}$$

k ... scale factor.
Similar: pitch shifting with timbre preservation

Pitch detection by cepstrum:

Regular intervals of harmonics ⇒ peak at period of fundamental frequency in s-domain

Also peaks for integer multiples ⇒ choose leftmost peak

4 Time-Domain Methods

Time stretching in the time domain: shifting overlapping short segments
Overlapping segments:

$$x_k[t] = x[kr + t] \qquad \text{for} \quad t = 0, \dots, n-1$$

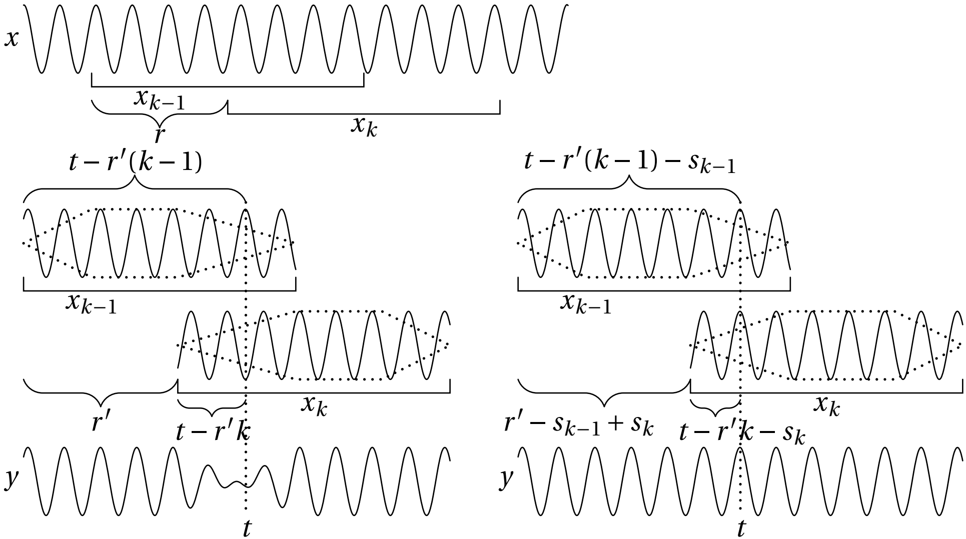
k … index of the segment
r … hop-size
n … segment length
Change hop-size to r' $\quad \Rightarrow$ phase mismatches $\quad \Rightarrow$ amplitude fluctuations
Solution: adjust by additional shift s_k:

$$y[t] = \sum_k x_k[t - kr' - s_k]\, w_k[t - kr' - s_k]$$

w_k … fade-in/fade-out window
Best fitting shifts s_k by cross-correlation:

$$c[s] = \sum_t x_{k-1}[t + r' - s_{k-1}]\, x_k[t - s] \qquad s_k = \arg\max_s c[s]$$

… **SOLA** (synchronous overlap-add)
More extreme scaling: repeat/omit segments (source segment $k(l)$ for destination segment l)



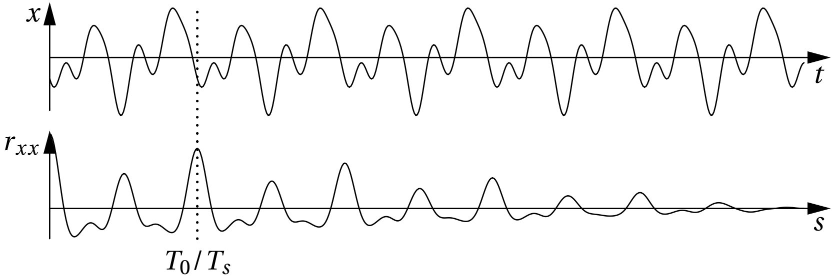
x
x_{k-1}
r
x_k
$t - r'(k-1)$
x_{k-1}
x_k
r'
$t - r'k$
y
t
$t - r'(k-1) - s_{k-1}$
x_{k-1}
x_k
$r' - s_{k-1} + s_k$
$t - r'k - s_k$
y
t

If pitch is known: **PSOLA** (pitch-synchronous overlap-add)

$r' - r + s_k - s_{k-1}$ must be a multiple of the pitch period τ:

$$s_k = \mathrm{round}\left(\frac{r' - r - s_{k-1}}{\tau}\right)\tau - (r' - r) + s_{k-1}$$

Pitch detection by auto-correlation:
$r_{xx}[s]$: peak at a lag of $s = T_0 / T_s$
$T_0 \ldots$ period of the signal ($T_0 = 1/f_0$)
$T_s \ldots$ sampling interval ($T_s = 1/f_s$)
$\Rightarrow \quad s = f_s / f_0$.



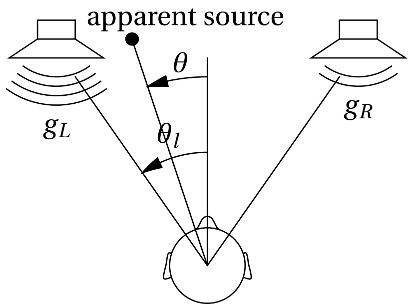
partial amplitudes $(0.4, 0.8, 0.4, 0.6, 0.1, 0.2, 0.1) \Rightarrow$ false peak at $0.5 \cdot T_0 / T_s$ (strong even partials)

Problems:
– lag is integer $\Rightarrow$ detected fundamental frequencies must not be too high
– fundamental frequency is not the only peak:
  – integer multiples (T_s-periodic $\Rightarrow$ also kT_s-periodic)
  – integer fractions (harmonics have smaller periods)

5 Spatial Effects

5.1 Sound Field Methods

Panorama:



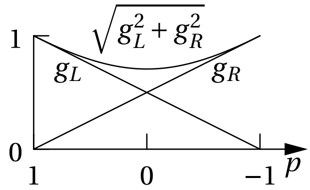
Apparent source direction

$$p := \frac{\tan\theta}{\tan\theta_l} = \frac{g_L - g_R}{g_L + g_R}$$

Linear interpolation (linear panning): “hole” in the center

Reason: $\sqrt{E(gx)} = \sqrt{g^2 E(x)} = g\sqrt{E(x)}$, but

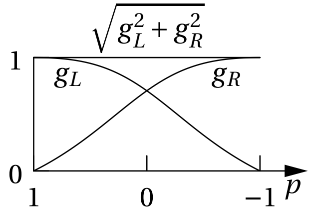
$$\sqrt{E(g_L x) + E(g_R x)} = \sqrt{g_L^2 E(x) + g_R^2 E(x)} = \sqrt{g_L^2 + g_R^2}\sqrt{E(x)}$$



Better:

$$g_L = \frac{1+p}{\sqrt{2(1+p^2)}}, \quad g_R = \frac{1-p}{\sqrt{2(1+p^2)}}$$

$\Rightarrow$ “overall gain” $\sqrt{g_L^2 + g_R^2} = 1$



True for broadband signals and low frequencies
Higher frequencies: different panning

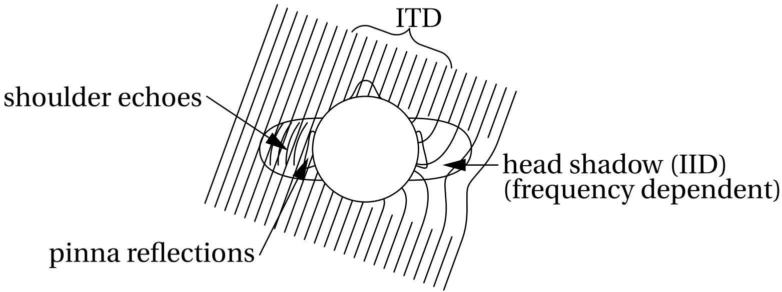
Precedence effect: short delay of up to 1 ms between speakers
⇒ sound appears nearer to speaker that emits sound first
effect strongly depends on the type of sound being played and the frequency

Inter-aural differences (in headphones):

- Inter-aural intensity difference (IID)
  basically a panorama effect
  depends on the frequency (less diffraction of higher frequencies ⇒ more head shadow)

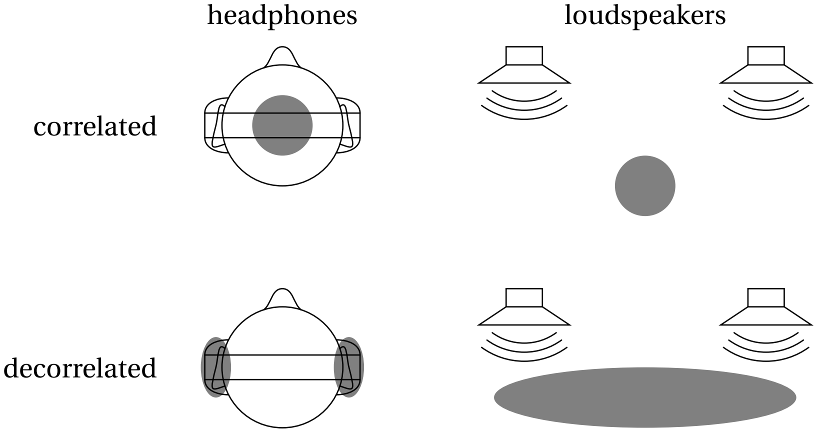
- Inter-aural time difference (ITD)
  time delay between the two channels
  depends on the frequency (below 1 kHz difference is greater, constant otherwise)

IID and ITD both depend on angle of the sound source



IID + ITD + shoulder echoes + pinna reflections: **head related transfer function** (**HRTF**)
measured by artificial dummy heads at different angles
approximated by IIR filters of an order of about 10

or: approximate head by a sphere:
– calculate the IID as a first-order IIR filter
– ITD implemented by delay
– shoulder echoes by single echo (angle-dependent delay)
– pinna reflections: short series of short-time echoes (very short angle-dependent delays)



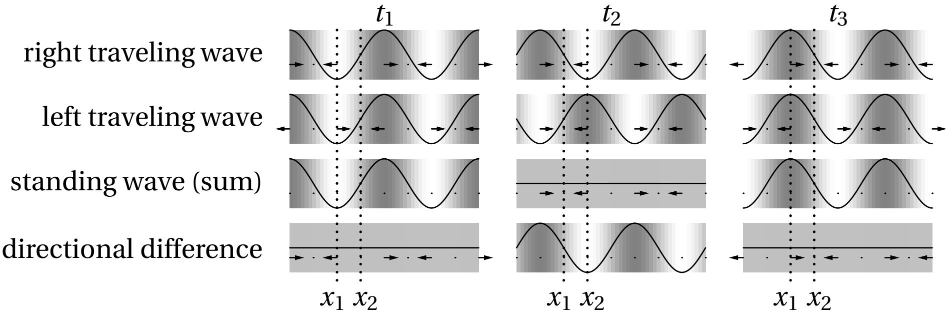
Correlation coefficient:

$$r(\tau) = \frac{\int x_L(t)\,x_R(t+\tau)\,\mathrm{d}t}{\sqrt{\int x_L^2(t)\,\mathrm{d}t \int x_R^2(t)\,\mathrm{d}t}}$$

Sound externalization: push apparent sound source out of head

Method: **decorrelation**: complex reverberation or convolution with uncorrelated white noise

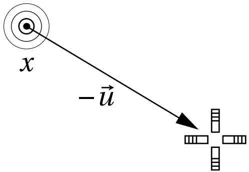
Traveling and standing waves:

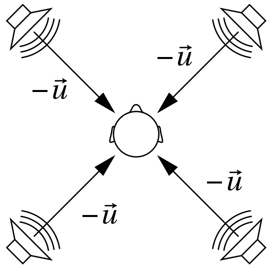


Animation

Capture 3D audio: sound field recording
Simple: place microphones and loudspeakers in same directions

Better: **Ambisonics**
– non-directional sound pressure component W
– three directional components X, Y, and Z





$$W = \text{front} + \text{back} + \text{left} + \text{right} + \text{up} + \text{down}$$

$$X = \text{front} - \text{back}$$

$$Y = \text{left} - \text{right}$$

$$Z = \text{up} - \text{down}$$

$$(W, X, Y, Z) = (\sqrt{2}/2, \vec{u}) \cdot x$$

Loudspeaker at direction $\vec{u}$:

$$\frac{1}{2}(G_1 W + G_2 (X, Y, Z)^\top \vec{u})$$

G_1, G_2 depend on the theory (there are several), frequency-dependent (filters)

Disadvantage: “sweet spots”

$\Rightarrow$ Higher-order versions of Ambisonics (higher derivatives) $\Rightarrow$ wider sweet spots

If elevation component is not needed $\Rightarrow$ ignore Z channel

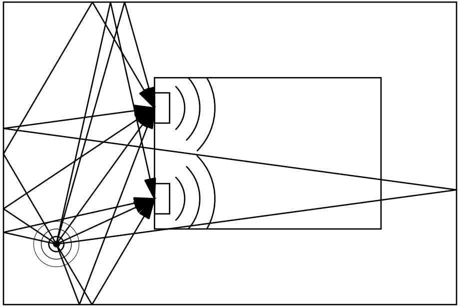
5.2 Reverberation

Apparent distance of sound from the listener, room size:
– direct sound
– reflections from walls
– ratio of direct to reverberating sound
    – direct sound loses energy with distance
    – reverberating sound fills room continuously

Direct sound delay T_d, reflection delay T_r $\quad \Rightarrow \quad$ cue for position

Problem: additional reverberation in room of listener

Robust method: **room-within-a-room** model.



– virtual holes in wall at loudspeaker positions

– delay according to the path length l from source to hole (delay = l/c, c ... speed of sound)

– paths may include reflections of the outer room

– gain set to $1/l$ (l in meters) (reason: spherical sound waves)

– gain limited to 1 to avoid infinite (or too high) gains

– attenuate if sound direction is opposite to speaker direction

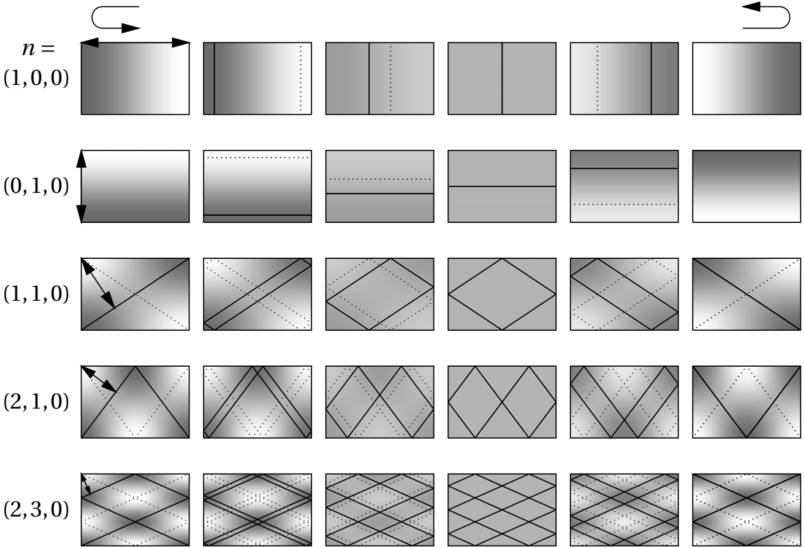
Problem: sound path calculation for multiple reflections computationally demanding
However: sound waves become increasingly planar and aligned with room geometry

Normal modes: standing waves in room

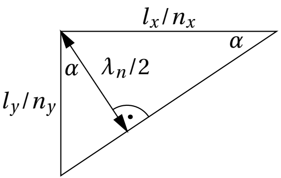
For room of size (l_x, l_y, l_z):
mode number vector (n_x, n_y, n_z) $(n_i = 0, 1, \ldots)$ corresponding to wave length

$$\lambda_n = 2\left(\left(\frac{n_x}{l_x}\right)^2 + \left(\frac{n_y}{l_y}\right)^2 + \left(\frac{n_z}{l_z}\right)^2\right)^{-\frac{1}{2}}$$

Impulse response of room: resonances at frequencies $f_n = c/\lambda_n$
For irreducible triplets n: fundamental frequency + multiples $\Rightarrow$ harmonic frequencies
$\Rightarrow$ implemented by comb filters



Animations:
(1,0,0)
(0,1,0)
(1,1,0)
(2,1,0)
(2,3,0)



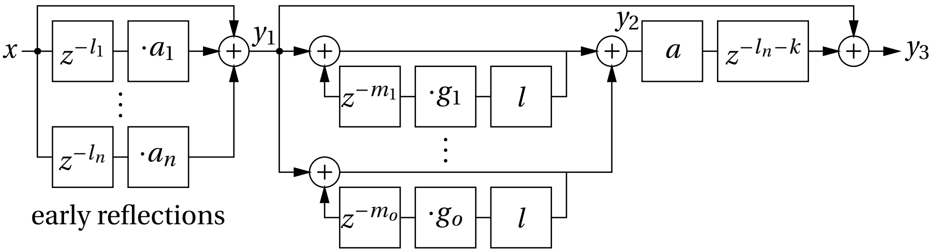
$$\frac{\lambda_n/2}{l_y/n_y} = \frac{l_x/n_x}{\sqrt{(l_y/n_y)^2+(l_x/n_x)^2}}$$

$$\Rightarrow \quad \lambda_n = 2\frac{1}{\sqrt{(n_x/l_x)^2+(n_y/l_y)^2}}$$

Reverberation without "coloration" (flat magnitude response): delay-based all-pass filter:

$$y[t] = (a * x) = cx[t] + x[t-m] - cy[t-m]$$

Combination of techniques: **Moorer's reverberator**.



Early reflections (delays l_i based on the sound trajectories):

$$y_1[t] = x[t] + a_1 x[t-l_1] + \ldots + a_n x[t-l_n]$$

IIR comb filters with a low-pass filter in the loop:

$$y[t] = (c * x)[t] = x[t] + g(l * y)[t-m]$$

applied in parallel for late reflections:

$$y_2[t] = c_1 * y_1 + c_2 * y_1 + \ldots + c_o * y_1$$

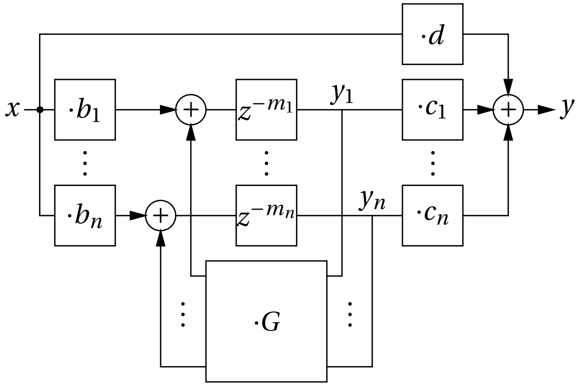
(m_i are based on wavelengths of room modes, low-pass filter simulates the behavior of the walls)
fed into all-pass filter, delayed and mixed together:

$$y_3[t] = y_1[t] + (a * y_2)[t - l_n - k]$$

Generalization of recursive comb filter $y[t] = x[t] + g \cdot y[t-m]$: **feedback delay network** (**FDN**)
g substituted by a matrix G:

$$\vec{y}[t] = x[t-\vec{m}]\vec{b} + G\vec{y}[t-\vec{m}] \qquad \text{and} \qquad y[t] = d\,x[t] + \vec{c}^{\mathsf{T}}\vec{y}[t]$$

($\vec{y}[t-\vec{m}]$ means: each component of $\vec{y}$ is delayed by a different delay m_i)



If G is a diagonal matrix $\Rightarrow$ set of parallel comb filters as in Moorer's reverberator
Non-diagonal elements of G: interaction between the room's normal modes (due to diffusive elements)

Taking the z-transform:

$$\vec{Y}(z) = \mathrm{diag}\left(z^{-\vec{m}}\right)\left(\vec{b}X(z) + G\vec{Y}(z)\right),$$

$$\left(\mathrm{diag}\left(z^{\vec{m}}\right) - G\right)\vec{Y}(z) = \vec{b}X(z),$$

$$H(z) = \frac{Y(z)}{X(z)} = d + \vec{c}^{\mathsf{T}}\left(\mathrm{diag}\left(z^{\vec{m}}\right) - G\right)^{-1}\vec{b}$$

Poles: $\det(\mathrm{diag}(z^{\vec{m}}) - G) = 0$

– should be inside unit circle to achieve a stable system)

– should have same absolute value (modes will decay at the same rate ⇒ no “coloration”)

– first lossless prototype (poles on unit circle, e.g. G unitary matrix)

– attenuation coefficients α^{m_i} in feedback loops

– make higher frequencies decay faster (attenuation coefficients now lowpass filters)

– Feedback matrices of special form (fast implementation, e.g. circular Toeplitz matrices ⇒ Fourier methods)

5.3 Convolution Methods

Real room reverberation: convolve the input signal with **room impulse response**

How to determine room impulse response?
Simple: emit impulse (at source position), record result (at listener position)
Problem: large signal peak, little sound energy

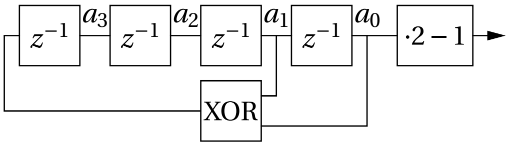
Crest factor:

$$C = \frac{\text{peak}|x|}{\text{RMS}(x)}$$

Solution: **maximum length sequences** (**MLS**) (pseudo-random binary (bit) sequences, generated by linear feedback shift registers)

Example (shift register of size 4 (a_3, a_2, a_1, a_0):

$$a_3[t] = a_0[t-1] \text{ XOR } a_1[t-1], \qquad a_k[t] = a_{k+1}[t-1] \quad \text{for} \quad k = 0, 1, 2.$$



For initial values 0001 for a, the result is

$$a_0[t] = 100010011010111\,100010011010111\,100010011010111\ldots$$

Properties of MLS:

- shift register size $m \quad \Rightarrow \quad$ sequence length $2^m - 1$
- half of the runs: length 1, quarter: length 2, eighth: length 3, ...
- $\approx$ half of bits are 1
- 0 substituted by $-1 \quad \Rightarrow$ crest factor 1 (= minimum)
- correlation property: auto-correlation $\approx$ impulses at intervals of $2^m - 1$

$$(a \star a)[k] = \sum_{t=0}^{2^m-2} a[t]a[t-k] \approx \begin{cases} 2^m - 1 & k = 0 \mod 2^m - 1 \\ 0 & \text{else} \end{cases}$$

So, $a \star a \propto \delta$ (apart from the repetition).

Extract room impulse response h from MLS response $y = h * a$:

$$y \star a = h * a \star a = h * \delta = h$$

Problem: direct convolution of impulse response with input signal computationally costly
Solution: convolution theorem (used on blocks):

$$\mathrm{FFT}^{-1}\Big(\mathrm{FFT}(x[0],\ldots,x[n-1]) \odot \mathrm{FFT}(\underbrace{h[0],\ldots,h[m-1],\ldots,0}_{\text{length } n})\Big)$$
$$= (x[0]\,h[0] + x[n-1]\,h[1] + x[n-2]\,h[2] + \ldots, \ldots)$$

$\odot$ ... pointwise multiplication

Problem: result is circular convolution

Solution: Zero-padding to length $n+m-1$:

$$\mathrm{FFT}^{-1}\left(\underbrace{\mathrm{FFT}(x[0],\ldots,x[n-1],\ldots,0)}_{\text{length } n+m-1} \odot \mathrm{FFT}(\underbrace{h[0],\ldots,h[m-1],\ldots,0}_{\text{length } n+m-1})\right)$$

$$= (x[0]\,h[0], x[1]\,h[0] + x[0]\,h[1], \ldots, x[n-1]\,h[0] + \ldots + x[n-m+1]\,h[m-1],$$

$$x[n-1]\,h[1] + \ldots + x[n-m]\,h[m-1], \ldots, x[n-1]\,h[m-1]) \,.$$

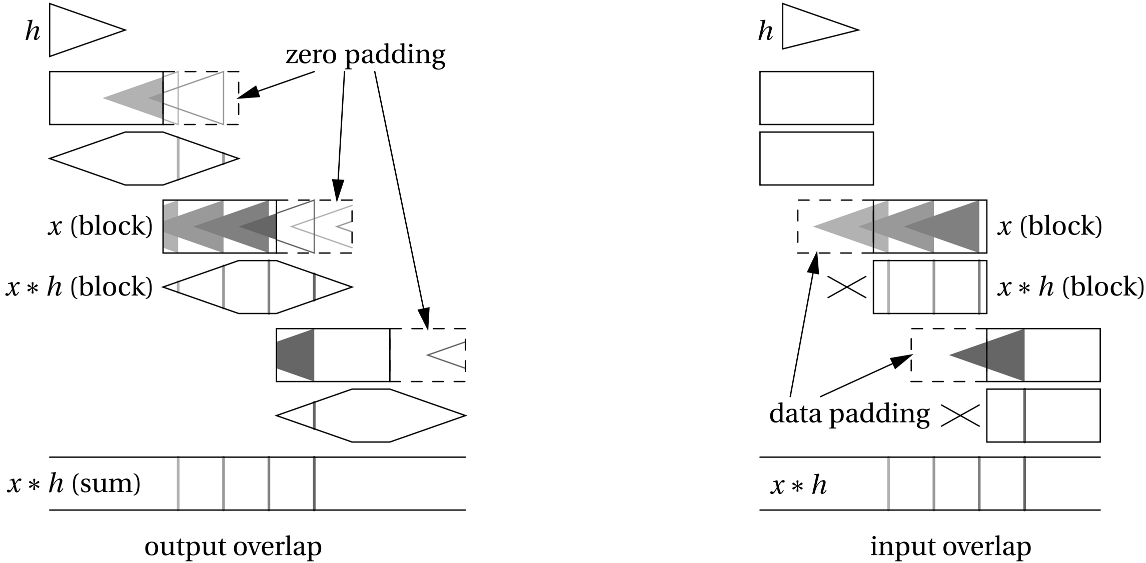
The result has to be overlap-added:

$$
\begin{array}{ll}
x[0]\,h[0] & \\
x[1]\,h[0] + x[0]\,h[1] & \\
\vdots & \\
x[n-1]\,h[0] + \ldots + x[n-m+1]\,h[m-1] & \\
x[n-1]\,h[1] + \ldots + x[n-m]\,h[m-1] & +x[n]\,h[0] \\
\vdots & \vdots \\
x[n-1]\,h[m-1] & +x[n+m-2]\,h[0] + \ldots + x[n]\,h[m-2] \\
 & \vdots \\
 & +x[2n-1]\,h[m-1]\,,
\end{array}
$$

left column: $h * x[0,\ldots,n-1]$, right column: $h * x[n,\ldots,2n-1]$.

Another possibility: input blocks of size $n+m-1$ overlap, discard $m-1$ samples of the result

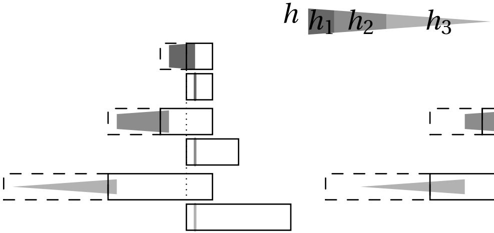
$$
\begin{aligned}
&\mathrm{FFT}^{-1}\Big(\mathrm{FFT}(x[-m+1],\ldots,x[n-1]) \odot \mathrm{FFT}(\underbrace{h[0],\ldots,h[m-1],\ldots,0}_{\text{length } n+m-1})\Big) \\
&\quad = (\cancel{x[-m+1]\,h[0] + x[n-1]\,h[1] + \ldots, \ldots, x[-1]\,h[0] + \ldots + x[n-1]\,h[m-1]}, \\
&\qquad x[0]\,h[0] + \ldots + x[-m+1]\,h[m-1], \ldots, x[n-1]\,h[0] + \ldots + x[n-m]\,h[m-1])\,.
\end{aligned}
$$



h
zero padding
x (block)
x ∗ h (block)
x ∗ h (sum)
output overlap
h
x (block)
x ∗ h (block)
data padding
x ∗ h
input overlap

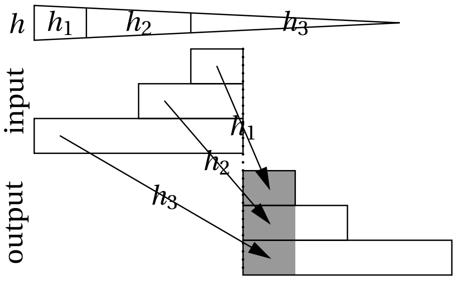
Problem: latency introduced by the block size

Solution: split the impulse response h into blocks $h_1, h_2, h_3, \ldots$ (increasing power-of-two sizes)



animation

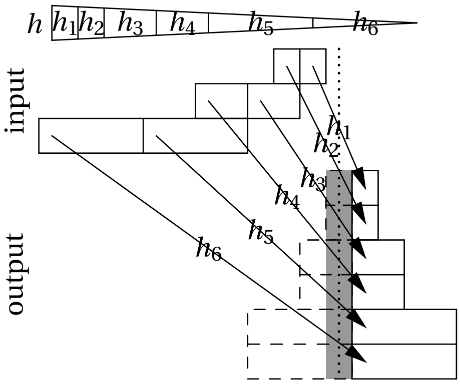
Overlap of input and output of the size of h_1 $\Rightarrow$ introduce some latency



ignoring computation time

animation

Practically: block computation time ≈ block time



considering computation time

animation

Zero-latency: prepend block h_0 ($1\times$ or $2\times$ size of h_1), direct convolution

In reality, I/O is blocked anyway, though.

6 Audio Coding

6.1 Lossless Audio Coding

Simplest approach: **silence compression**:
– runs of zero values: runlength-coding
– almost silent parts set to zero (actually lossy)

Better: linear prediction (**linear predictive coding**):
– optimized filter (**Levinson-Durbin recursion**) predicts samples
– encode prediction error

Prediction error has two-sided geometric distribution:

$$p_k = P(x[t] - (p * x)[t] = k) \propto s^{-|k|}$$

Efficiently encoded with **Rice codes**, or Golomb-Rice codes:
– parameter M ($\propto$ variance of the distribution), power of two
– divide k by M $\Rightarrow$ quotient q, remainder r:

$$k = Mq + r$$

– q encoded as unary code (q ones followed by a zero)
– r encoded as $\log_2(M)$ bits

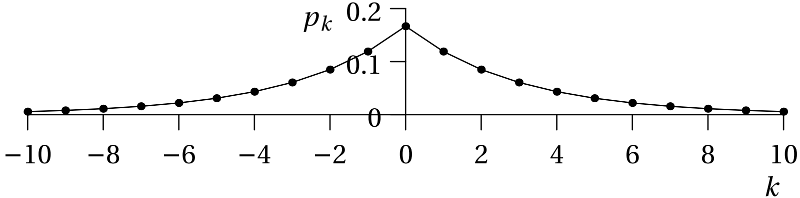
Example ($M = 4$):

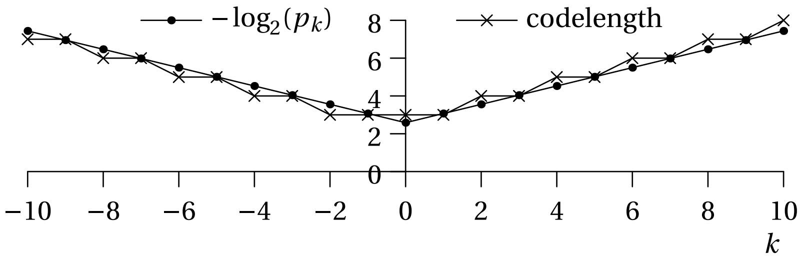
k	code	k	code	k	code	k	code
0	000	4	1000	8	11000	12	111000
1	001	5	1001	9	11001	13	111001
2	010	6	1010	10	11010	14	111010
3	011	7	1011	11	11011	15	111011

Only suitable for positive k
Signed k: $k \mapsto 2k$ for $k \geq 0$, $k \mapsto 2|k| - 1$ for $k < 0$

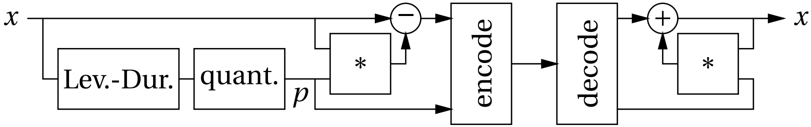
Example: two-sided geometric distribution $p_k = \frac{1}{6} \cdot 1.4^{-|k|}$

Self-information $-\log_2(p_k)$ compared to the codelengths for $M = 4$:

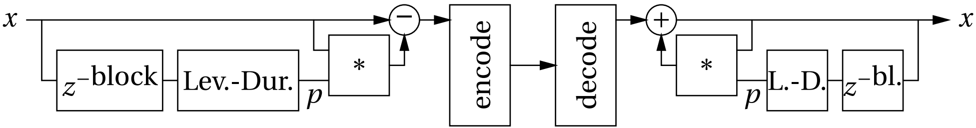




Forward-adaptive prediction:



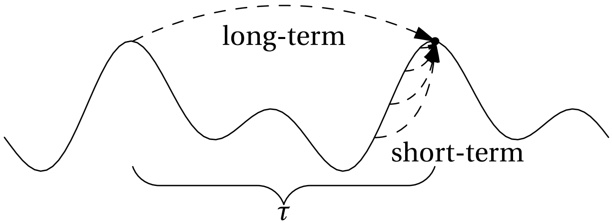
Backward-adaptive prediction:



Disadvantage: coefficients not optimized for current block

Advantages: coefficients not encoded, longer filters possible, non-quantized coefficients

Long-term prediction and short-term prediction:



τ: optimal period (similar to pitch detection)

One to five values around $t-\tau$ for prediction

Short-term and long-term prediction can be combined

Standards: FLAC (Free Lossless Audio Codec), MPEG-ALS
– many optimization details

6.2 Lossy Audio Coding

Early simple approaches: μ-law and A-law encoding (logarithmic quantization)

Approaches with linear prediction:

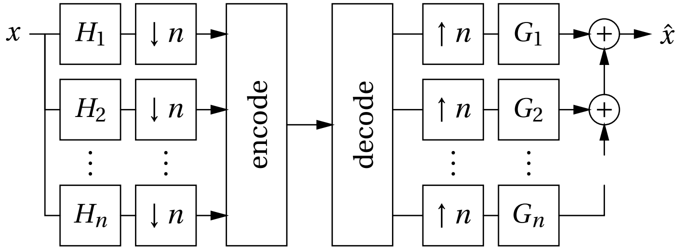
- DPCM (differential pulse code modulation) and ADPCM (adaptive DPCM): only quantized prediction errors encoded
- Pure linear predictive coding: only prediction filter coefficients encoded
- CELP (code excited linear predictor): both encoded

Advanced approach: transform coding (transform of block, quantize and encode coefficients)

Problem: High-frequency artifacts at block borders

Windows and overlapping cannot be used (increase of data size)

Solution 1: filter banks (instead of blocked transform)



H_i … bandpass filters with different center frequencies

$\downarrow n$ … downsampling by a factor of n

$\uparrow n$ … upsampling (insertion of $n-1$ zeros after each element)

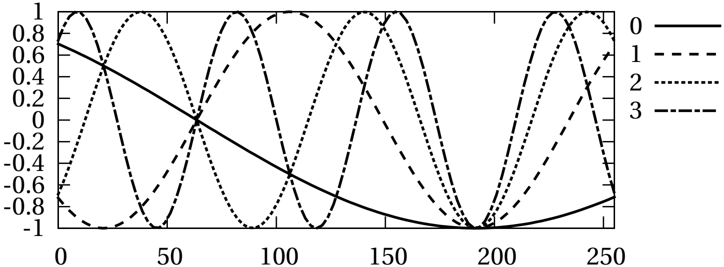
G_i reconstruction filters (H_i and G_i fulfill a “perfect reconstruction” constraint)

Used in MPEG audio level 1–2

Solution 2: **modified discrete cosine transform** (**MDCT**):

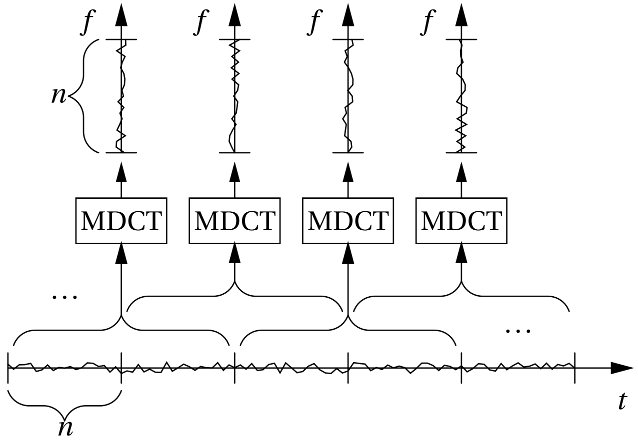
$$X[w,t] = \sum_{s=0}^{2n-1} x[nt+s]\cos\left(\frac{\pi}{n}\left(s+\frac{1}{2}+\frac{n}{2}\right)\left(w+\frac{1}{2}\right)\right)$$

$n \ldots$ hop-size $\qquad 2n \ldots$ block size $\qquad w = 0, \ldots, n-1$



First four basis functions of the MDCT for $n = 128$

Block of size $2n$ produces only n MDCT coefficients, but 50% overlap of blocks

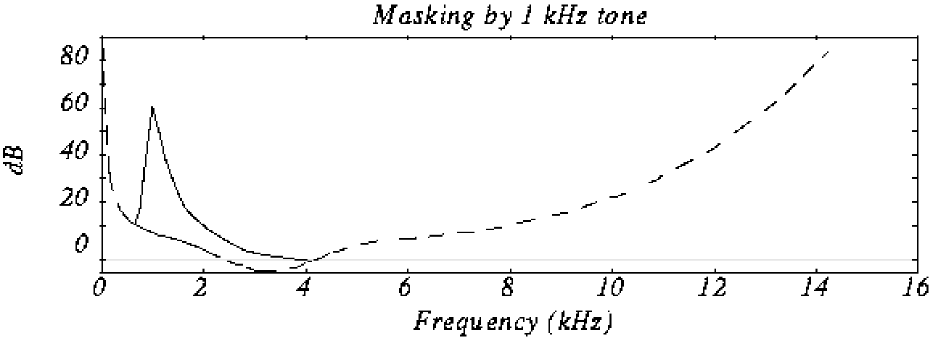


MDCT blocks can be windowed (has to satisfy $w[s]^2 + w[s+n]^2 = 1$)
Used in MPEG audio layer 3 (MP3, in addition to filter banks), MPEG-AAC (advanced audio coding), Vorbis.

Transformed data: quantized and encoded (entropy coders: Huffman, arithmetic coding)

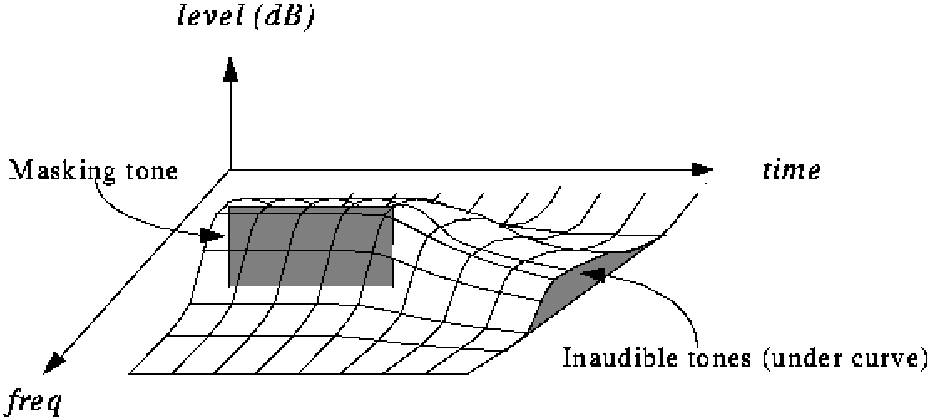
Improvement: adaptively choosing quantization factors on a coefficient basis
⇒ **psychoacoustics**

1. Frequency masking:



⇒ quantize so that quantization is below masking threshold

2. Temporal masking:



Used in all state-of-the-art lossy audio codecs: MP3, AAC, Vorbis

Disadvantages of major audio codecs:
– latency (due to blocked processing ⇒ unusable for interactive audio)
– bad compression performance for very low bit-rate and speech coding
(predictive techniques still better)
– heavily patent covered techniques

Solution: Opus codec
– frequency-domain techniques for higher bit-rates
– can switch to predictive coding dynamically
– uses small block sizes (less latency) (special techniques to overcome low frequency resolution)

Problem for low bit-rates: high frequencies usually dropped entirely
Solution: **spectral band replication**
– synthesizes higher frequency bands by extrapolating frequency content in lower bands
– harmonic signals supplemented with more harmonic frequencies in higher bands
– low-frequency noise with high-frequency noise
– may be guided by low-bit-rate side information encoded by the encoder
– result: only approximation, but sounds “nice”, improves comprehensibility of speech

The End