Lecture 1: Signals

What is a signal?

$\mathbb{R}$ ← continous in amplitude

$x(t) \rightarrow X(nT_s) : x[n]$; $n \in \{-\infty, \dots, -2, -1, 0, 1, 2, \dots, \infty\}$

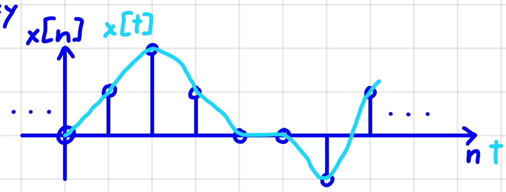
sampling on time points

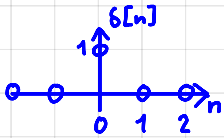
$\in$

continous time signal

$\frac{1}{T_s} = f_s$ sampling frequency

discrete (in time)



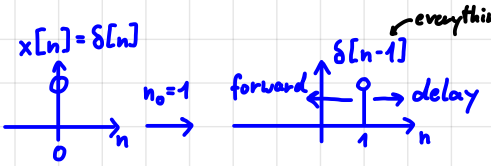


Impulse: $\delta[n] = \begin{cases} 1 ; & n = 0 \\ 0 ; & \text{otherwise} \end{cases}$

Time shifts:

$x[n] \rightarrow x[n-n_0]$

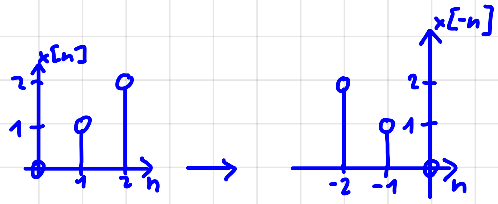
everything happens one time step later
⇒ shifted to the right



Time reversal:

$x[n] \rightarrow x[-n]$

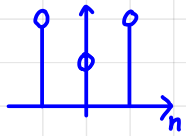
Verwendung bei linearer/diskreter convolution



Scaling:

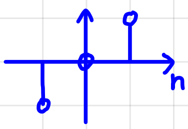
$x[n] \rightarrow \alpha \cdot x[n] ; \alpha \in \mathbb{R}$

Even signals: $x[n] = x[-n]$



symetric allong the y-axes

Odd signals: $x[-n] = -x[n]$



symetric allong the diagonal

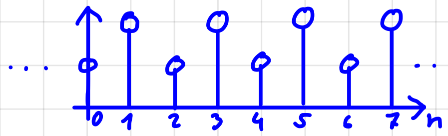
It is always possible to write $x[n] = \text{Even}\{x[n]\} + \text{Odd}\{x[n]\}$,

where $\text{Even}\{x[n]\} = \frac{1}{2}(x[n] + x[-n])$

$\text{Odd}\{x[n]\} = \frac{1}{2}(x[n] - x[-n])$.

Periodic signals: $x[n] = x[n+N] \quad \forall n \; ; \; N \in \mathbb{N}$

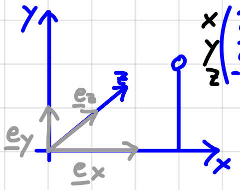
$\uparrow$ period



$N = 2$

What is the implicit basis of a signal?

3D-Euclidean space

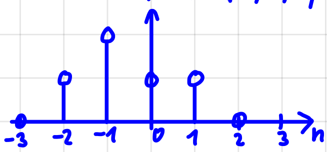


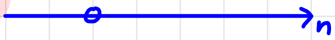
$\begin{matrix} x \\ y \\ z \end{matrix} \begin{pmatrix} 2 \\ 2 \\ -1 \end{pmatrix} = \underline{w}$

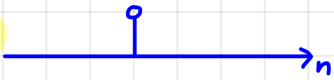
$\underline{w} = 2 \cdot \underline{e}_x + 2 \underline{e}_y - 1 \cdot \underline{e}_z$

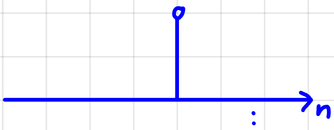
explicit basis

$x[n] = (0, 1, 2, 1, 1, 0)$



$0 \cdot \delta[n+3]$ 

$1 \cdot \delta[n+2]$ 

$2 \cdot \delta[n+1]$ 

$\vdots$

terms are explicit

$x[n] = 0\, \delta[n+3] + 1 \cdot \delta[n+2] + 2 \cdot \delta[n+1] + \ldots$

$$x[n] = \sum_{m=-\infty}^{\infty} x[m] \cdot \delta[m-n]$$

$x[n]$ has the basis of delayed impulses

Lecture 1b: Systems

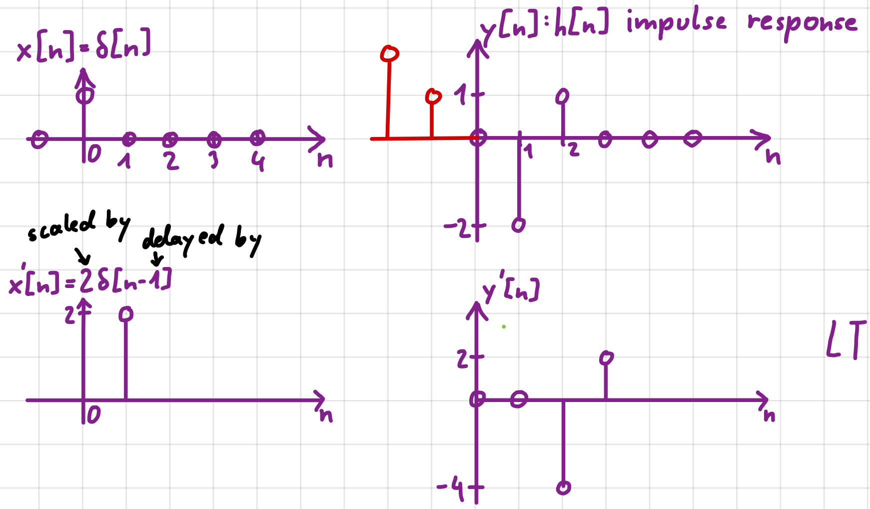
Convolutions of LTI Systems are all aditive because of linearity!

What is a system?

Black box

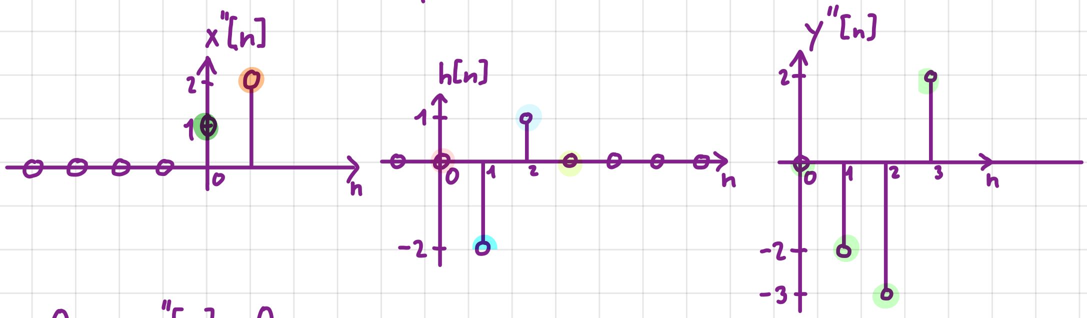
$$x[n] \longrightarrow \boxed{T\{x[n]\}} \longrightarrow y[n]$$

Example:



LTI - Linear time-invariant system

Input: $x''[n] = x[n] + x'[n]$; $h[n]$ as above



$n<0$: $y''[n] = 0$

$n=0$: $y''[0] = x''[0] \cdot h[0] = 1 \cdot 0 = 0$

$n=1$: $y''[1] = x''[0] \cdot h[1] + x''[1] \cdot h[0] = 1 \cdot (-2) + 2 \cdot 0 = -2$

$n=2$: $y''[2] = x''[0] \cdot h[2] + x''[1] \cdot h[1] = -3$

$n=3$: $y''[3] = x''[0] \cdot h[3] + x''[1] \cdot h[2] = 1 \cdot 0 + 2 \cdot 1 = 2$

$n \geq 4$: $y''[n] = 0$

Lineare / diskrete Convolution: $y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k]$

Notes:

- An LTI-system is fully characterized by its impulse response
- The convolution is symmetric, i.e., $x[n] * h[n] = h[n] * x[n]$.
- Systems can also be characterized by difference equations:

$$y[n] = \sum_{l=1}^{L} a_l \cdot y[n-l] + \sum_{k=0}^{M} b_k \cdot x[n-k]$$

$x[n] \rightarrow \otimes \rightarrow \boxed{} \rightarrow y[n]$

often used when impuls responses are infinit

Example: The accumulator

$$y[n] = \sum_{k=-\infty}^{n} x[k]$$

Impulse response: $k[n] = \begin{cases} 1; & n \geq 0 \\ 0; & \text{otherwise} \end{cases}$

Difference equation: $y[n] = y[n-1] + x[n]$

Properties of LTI-systems:

Definition: An LTI-system is stable if $\exists B_x, B_y \in \mathbb{R} < \infty$ s.t. (when Input minimal amount of energy, Output should also be minimal amount of energy)

∞-Norm

$$\|x[n]\|_\infty \leq B_x \Rightarrow \|y[n]\|_\infty \leq B_y.$$

when Input is bounded, Output must also be bounded

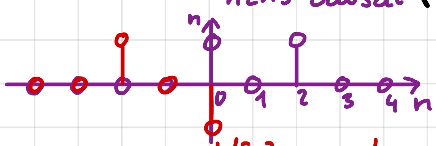
Example: $y[n] = x[n] + \alpha y[n-1]$; $\alpha \in \mathbb{R}$

is $\begin{cases} \text{stable} & \text{for } |\alpha| < 1 \leftarrow \text{(slowly decay to zero)} \\ \text{unstable} & \text{for } |\alpha| \geq 1 \rightarrow \text{(Output of the system will grow without bounds)} \end{cases}$

Remark: LTI systems are stable if the absolute sum of their impulse response is finit.

Definition: We call a system causal iff (only if) $k[n] = 0 \;\; \forall n < 0$.

h[n] causal ← because it only response when it has seen an input

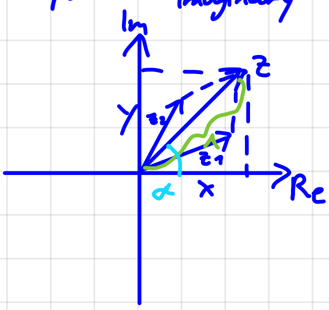


h'[n] acausal ← it response to an input bevor it has seen the input

Offline signal processing after recording the whole signal gives much better performance then online.

Review of complex numbers

$$\underset{\text{complex}}{z} = \overset{\text{real}}{x} + j\underbrace{y}_{\text{imaginary}} \; ; \; j^2 = -1 \;, \; x, y \in \mathbb{R}$$



Addition: $z_1 + z_2 = (x_1 + jy_1) + (x_2 + jy_2) = (x_1 + x_2) + j(y_1 + y_2)$

$$\underline{z}_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} ; \; \underline{z}_2 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \qquad z = \underline{z}_1 + \underline{z}_2 = \begin{bmatrix} x_1 + x_2 \\ y_1 + y_2 \end{bmatrix}$$

Multiplication: $z_1 \cdot z_2 = (x_1 + jy_1) \cdot (x_2 + jy_2) = (x_1 \cdot x_2 - y_1 y_2) + j(x_1 y_2 + x_2 y_1)$

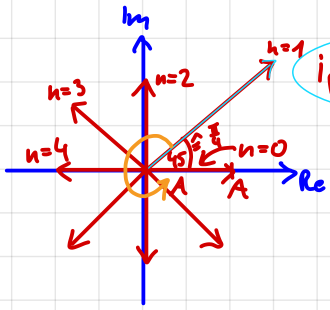
Euler's formula: $z = \underline{A} \cdot e^{j\alpha} = \underbrace{A\cos(\alpha)}_{x} + j \cdot \underbrace{A \cdot \sin(\alpha)}_{y}$

$$z_1 \cdot z_2 = A_1 \cdot e^{j\alpha_1} \cdot A_2 \cdot e^{j\alpha_2} = \underbrace{A_1 \cdot A_2}_{|z'|} \cdot e^{j(\alpha_1 + \alpha_2)}$$

$|z'|$ – Length of complex number

⇒ Multiplying complex numbers multiplies their amplitudes and sums their angles!

Complex oscillations: $x[n] = \underset{\text{Amplitude}}{A} \cdot e^{j\frac{2\pi}{k}n}$ — frequency; angle or phase



if $k = 8$; $\omega = \frac{2\pi}{k}$

How does a LTI-system respond to a complex oscillation?

$$y[n] = h[n] * x[n] = \sum_{m=-\infty}^{\infty} h[m] \cdot x[n-m] \;\Big|\; x[n-m] = e^{j\omega n}$$

$$= \sum_{m=-\infty}^{\infty} h[m] \cdot e^{j\omega(n-m)} = \underbrace{\sum_{m=-\infty}^{\infty} h[m] \cdot e^{-j\omega m}}_{H(\omega)} \cdot \underbrace{e^{j\omega n}}_{x[n]} = \underbrace{H(\omega)}_{\text{constant factor}} \cdot \underbrace{x[n]}_{\text{complex oscillation}}$$

advanced by phase factor

ANALYSIS

Eigenvector of a matrix A

$A\underline{x} = \lambda x$

→ Complex oscillations are eigenfunctions of LTI-systems!

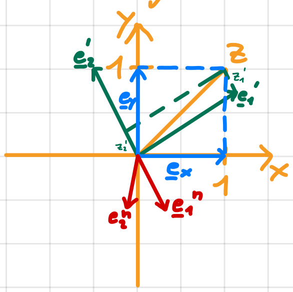
Frequency of the Input does not change even when amplitude or phase change by the Output

Lecture 2: The Discrete (-Time) Fourier Transform

willkürlich

Can we model arbitrary signals as sums of complex exponential oscillations?

Linear algebra analogy



$z=\begin{bmatrix}1\\1\end{bmatrix};\quad z'=\begin{bmatrix} z^T\cdot e_1' \\ z^T\cdot e_2' \end{bmatrix} = \begin{bmatrix} 1\cdot e_1'^{1}+1\cdot e_1'^{2} \\ 1\cdot e_2'^{1}+1\cdot e_2'^{2} \end{bmatrix}$ Basis change (is possible when basis spans our whole space)

Vector with new basis

For orthonormal basis vectors, the coordinates of a vector in the new basis we given by the inner product of the vector in the original basis with the new basis vectors.

Considering that a time-series $x[n]$ of length N has the implicit basis $\delta[0],\dots,\delta[N-1]$, can we construct N orthogonal complex oscillations?

tried oszilations

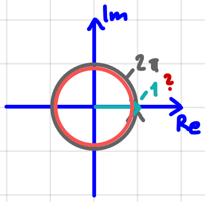
$$e_k[n]=e^{j\frac{2\pi k}{N}\cdot n};\quad k\in\{0,\dots,N-1\}$$

dot product of two frequencies; 1. sequence; 2. sequence; comes from complex contribute

$$\langle e_k[n],e_l[n]\rangle=\sum_{n=0}^{N-1} e^{j\frac{2\pi k}{N}\cdot n}\cdot e^{-j\frac{2\pi l}{N}\cdot n}$$

$$=\sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}(k-l)\cdot n}$$ | geometric series (important when $k\neq l$)

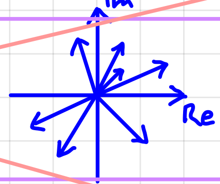
$$=\begin{cases} N & ; k=l \\ \dfrac{1-\overbrace{e^{j2\pi(k-l)}}^{1}}{1-e^{j2\pi(k-l)/N}}=0 & ; k\neq l \end{cases}$$



⇒ The set of complex oscillations with frequencies $\frac{2\pi k}{N}$ forms an orthogonal basis for time-series of length N.

?

$\cancel{\underline{x}^T\cdot\underline{y}=\langle\underline{x},\underline{y}\rangle=\alpha}$ $\cancel{X[k=10]}$ $\cancel{10\cdot e^{j\frac{2\pi\cdot 10}{N}\cdot n}}$



We can hence express signals in the spectral (frequency) domain by protecting the signal on the new basis signals (the complex exponentials):

$\cancel{\langle x[n],e^{j\frac{2\pi k}{N}\cdot n}\rangle=\underline{e}^{*}\cdot\underline{x}}$

- Analysis: $X[k]=\sum_{n=0}^{N-1}x[n]\cdot e^{-j\frac{2\pi k}{N}n}$ time and frequency is discrete (Diskrete Fourier Transformation)
- Synthesis: $x[n]=\frac{1}{N}\sum_{k=0}^{N-1}X[k]\cdot e^{j\frac{2\pi k}{N}n}$ (Inverse Diskrete Fourier Transformation)

only time is discrete, frequency is continous

Letting $N\to\infty$, we obtain the Diskrete-Time Fourier Transformation:

- Analysis: $X(\omega)=\sum_{n=-\infty}^{\infty}x[n]\cdot e^{-j\omega n};\ \omega\in\,]-\pi;\pi]$ continous range of frequencies
- Synthesis: $x[n]=\frac{1}{2\pi}\int_{-\pi}^{\pi}X(\omega)\cdot e^{j\omega n}\,d\omega$

Example: The ideal low-pass filter (let pass all frequencies below a certain frequency with a gain of one

is $\begin{cases} 1 \text{ between } [-\omega_c, \omega_c] \\ 0 \text{ otherwise} \end{cases}$

$$h[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(\omega)\cdot e^{j\omega n}\,d\omega = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\omega n}\,d\omega$$

$$= \frac{1}{2\pi j n} e^{j\omega n}\Big|_{-\omega_c}^{\omega_c} = \frac{1}{2\pi j n}\left(e^{j\omega_c n} - e^{-j\omega_c n}\right)$$ | Euler's formula

where - but flipping the angle means flipping the sign

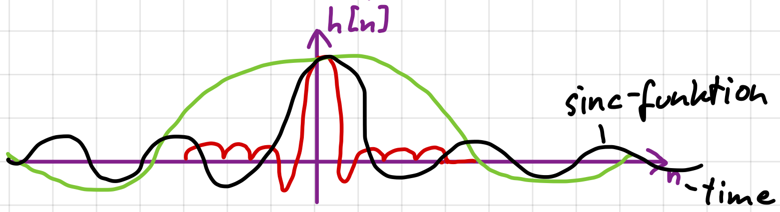
$$= \frac{1}{2\pi \cancel{j} n}\left(\cancel{\cos(\omega_c n)} + \cancel{j}\sin(\omega_c \cdot n) - \cancel{\cos(-\omega_c \cdot n)} + \cancel{j}\cdot\sin(+\omega_c\cdot n)\right) = \frac{1}{\cancel{2}\pi n}\left(\cancel{2}\sin(\omega_c \cdot n)\right)$$

sine-function of a certain frequency

$$= \frac{1}{\cancel{2}\pi n}\cdot\cancel{2}\sin(\omega_c \cdot n) =: \operatorname{sinc}(\omega_c \cdot n)\cdot 2f_c \quad \text{where } \omega_c = 2\pi f_c$$

damped by n

$[\operatorname{sinc}(x) := \frac{\sin(x)}{x}]$



Lecture 3: Properties of the D(T)FT

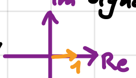
Example: What is the spectrum of the impulse $\delta[n]$?

equation to compute

$$\underbrace{\text{DTFT}\{\delta[n]\}}_{\text{Dirac-Impulse}} = \underbrace{\sum_{n=-\infty}^{\infty} \delta[n] \cdot e^{-j\omega n}}_{\text{because } \delta[n] \text{ is always } 0, \text{ except when } n=0} = 1 \cdot e^{-j\omega n}\Big|_{n=0} = 1$$

angle is 0

← for every frequency (every angle is 0)



x-Achse = Zeit T

Next, consider the power of a signal in the time-domain:

$$\underbrace{\sum_{n=-\infty}^{\infty} |x[n]|^2}_{\text{Energy in time domain}} = \sum_{n=-\infty}^{\infty} x[n] \cdot x^*[n] = \underbrace{\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty}}_{\text{(trick)}} x[n] \cdot x^*[m] \cdot \delta[n-m]$$

sneek in the impulse

$$= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x[n] \cdot x^*[m] \cdot \underbrace{\frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-j\omega(n-m)} d\omega}_{\text{Frequency expression of}}$$

iDTFT{$\delta[n]$} – inverse DTFT

InverseDTFT{1} = $\delta(n)$

flipps sign of imaginary part

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} \underbrace{\sum_{n=-\infty}^{\infty} x[n] \cdot e^{-j\omega n}}_{X(\omega)} \cdot \underbrace{\sum_{m=-\infty}^{\infty} x^*[m] \cdot e^{j\omega m}}_{X^*(\omega)} d\omega$$

$X(\omega)$ frequency domain

$X^*(\omega)$ Spectrum of our signal in the frequency domain; complex conjungaet of the spectrum of X

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) \cdot X^*(\omega)\, d\omega = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(\omega)|^2 d\omega$$

⇒ power is preserved (variants in the frequency domain are equal the variants in the time domain)

known as

Parseval's theorem: $\displaystyle\sum_{n=-\infty}^{\infty} |x[n]|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(\omega)|^2 d\omega$

Omega = $2\pi f$ | f [Hz]

$$H(\omega) \cdot X(\omega) = \sum_{n=-\infty}^{\infty} h[n] \cdot e^{-j\omega n} \cdot \sum_{m=-\infty}^{\infty} x[m] \cdot e^{-j\omega m}$$

n & m ... because the two FT should not interact

sind fourier transformatierte von h[n] (Reihe)

$$= \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} h[n] \cdot x[m] \cdot e^{-j\omega(n+m)} \quad \Big|\, k := n+m$$

$$= \underbrace{\sum_{k=-\infty}^{\infty} \underbrace{\sum_{n=-\infty}^{\infty} h[n] \cdot x[k-n]}_{h[n] * x[n]} \cdot e^{-j\omega k}}_{\text{DTFT}\{\ \}}$$

$h[n] * x[n]$ ← convolution

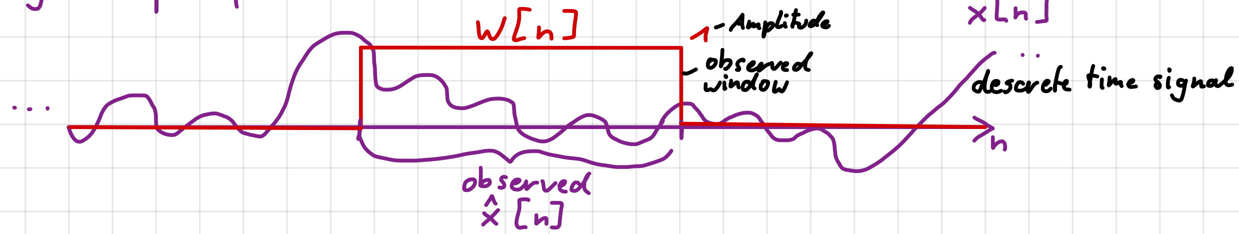
DTFT{ } ← DTFT of convolution

$$= \text{DTFT}\{h[n] * x[n]\}$$

Reihen in der Zeit (diskrete Darstellung)

Convolution theorem: $\underbrace{\text{DTFT}\{h[n] * x[n]\}}_{\text{DTFT of convolution in time domain}} = \underbrace{H(\omega) \cdot X(\omega)}_{\text{multipling two spectra in the frequency domain}}$

Motivating example for the modulation theorem:



What is the spectrum of $\hat{x}[n]$?

$$\hat{x}[n] = w[n] \cdot x[n]$$

$w[n]$: window function

$x[n]$: infinitly long

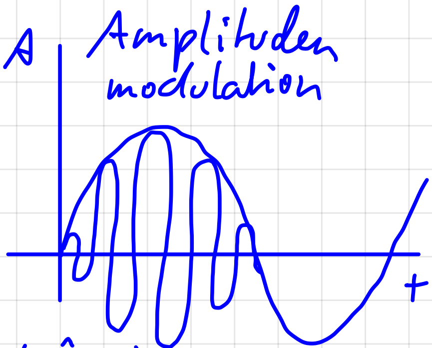
$$\hat{X}(\omega)=\sum_{n=-\infty}^{\infty} w[n]\cdot x[n]\cdot e^{-j\omega n}=\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} w[n]\cdot x[m]\cdot \delta[n-m]\cdot e^{-j\omega n}$$

$$=\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} w[n]\cdot x[m]\cdot \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{j\theta(n-m)}\, d\theta \cdot e^{-j\omega n}$$

theta (frequency index)

$$=\frac{1}{2\pi}\int_{-\pi}^{\pi} \underbrace{\sum_{n=-\infty}^{\infty} w[n]\cdot e^{-j(\omega-\theta)\cdot n}}_{W(\omega-\theta)\ \text{FT}} \cdot \underbrace{\sum_{m=-\infty}^{\infty} x[m]\cdot e^{-j\theta m}}_{X(\theta)\text{ - Fourier transform of } X}\, d\theta$$

$$=\frac{1}{2\pi}\int_{-\pi}^{\pi} \underbrace{W(\omega-\theta)\cdot X(\theta)\, d\theta}_{\text{continous analog of the convolutio}} = \overbrace{W(\omega) * X(\omega)}^{\text{complex analog of the convolution}}$$

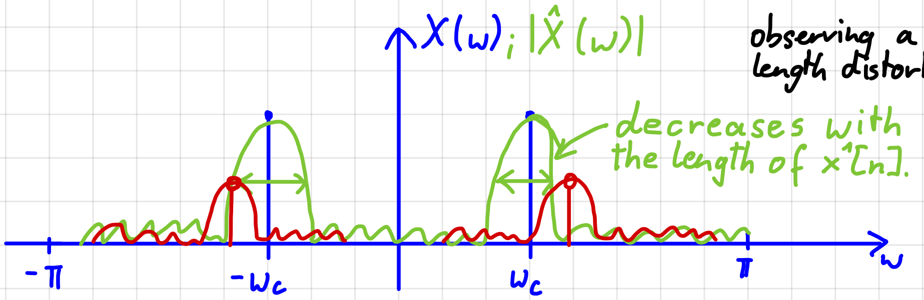


Modulation theorem: $\mathrm{DTFT}\{w[n]\cdot x[m]\} = W(\omega) * X(\omega)\ (=\hat{X}(\omega))$

spectrum of rectangular window; convolving their spectra

$$w[n]=\begin{cases}1; & -n_c < n \le n_c \\ 0; & \text{otherwise}\end{cases} \quad \overset{}{\underset{\text{DTFT}}{\Longleftrightarrow}} \quad W(\omega)^{1} = \text{sinc-function}$$

?



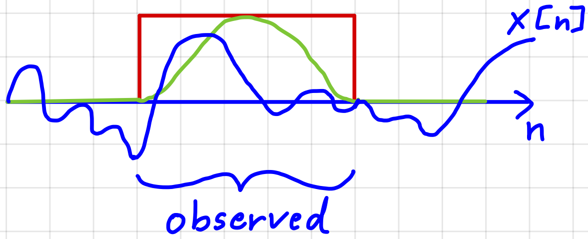
observing a signal for only a finite length distorts your spectrum and may make it difficult to propperly resolve all the spectrum components of your signal

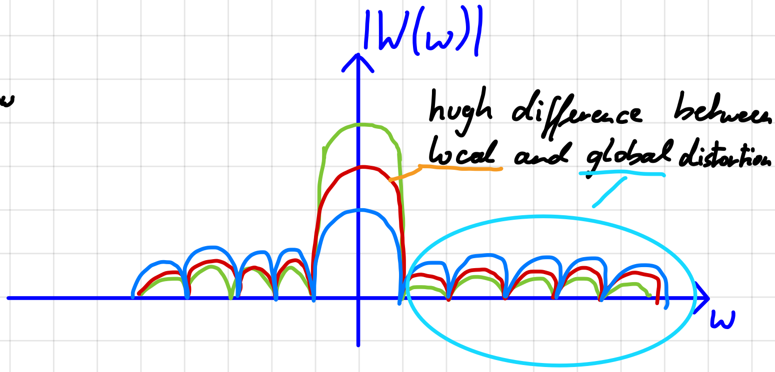
shorter signal → more sevier is the distortion

Window design:

can't avoid some distortion of finit length data, but have some flexibility by choosing the window

$$\hat{x}[n] = w[n]\cdot x[n]$$



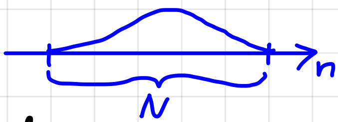


hugh difference between local and global distortion

One of the most popular windows is the Hann-window:

$$w[n]=\frac{1}{2}\left(1-\cos\left(\frac{2\pi n}{N-1}\right)\right)$$

N: length of observed signal



Symmetries in the DTFT:

Consider the $\mathrm{DTFT}\{x^*[n]\}$:

$$\mathrm{DTFT}\{x^*[n]\} = \sum_{n=-\infty}^{\infty} x^*[n]\cdot e^{-j\omega n} = \left(\sum_{n=-\infty}^{\infty} x[n]\cdot e^{+j\omega n}\right)^* = \underbrace{X(-\omega)^*}_{}$$

makes - to +

original spectrum that is flipped alonge the frequencies and have to take the complex conjugate

Next, assume $x[n]=x^*[n]$ ($x[n]$ is real-valued). Then

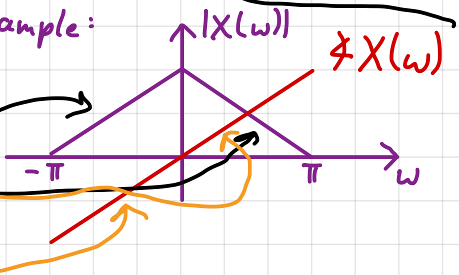
$$X(\omega) = X(-\omega)^*$$

spectrum of X

Amplitude $\Rightarrow$ $|X(\omega)| = |X(-\omega)|$

phase $\measuredangle X(\omega) = -\measuredangle X(-\omega)$

Example:



check: spectrum should be simetric otherwise something is wrong

Linear vs. circular Convolution:

DTFT: $X(\omega) = \sum_{n=-\infty}^{\infty} x[n] \cdot e^{-j\omega n} \longleftrightarrow x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)\cdot e^{j\omega n}\, d\omega$

infinite length signal

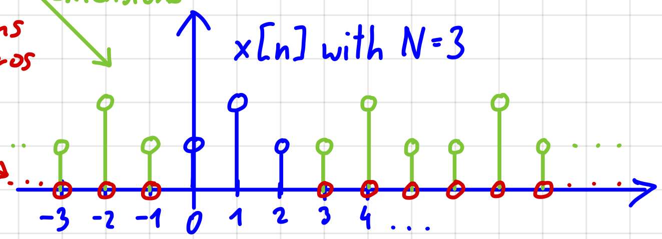
DFT: $X(k) = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}\cdot k \cdot n} \longleftrightarrow x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\cdot e^{j\frac{2\pi}{N}\cdot k\cdot n}$

finit length signal

finit number of frequencies

implicitly assume periodic extensions

implicitly assume extensions with zeros



$iDTFT\{X(\omega)\cdot H(\omega)\} = \underbrace{x[n] * h[n]}_{\text{linear convolution}}$ with "$*$" the linear convolution.

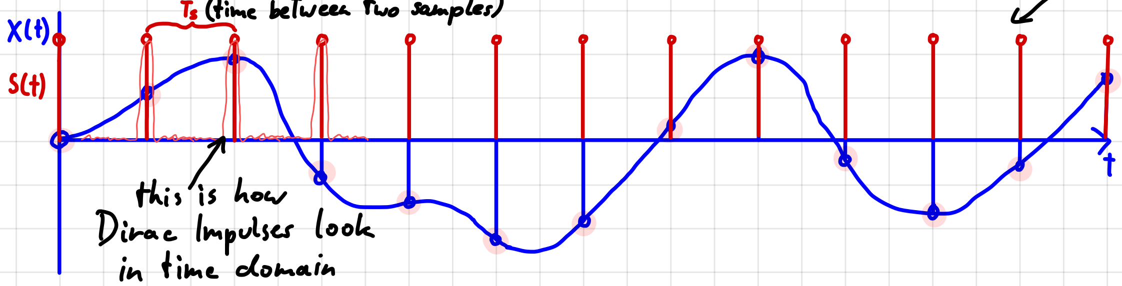
$iDTFT\{X(k)\cdot H(k)\} = \underbrace{x[n] \circledast h[n]}_{\text{circular convolution}}$ with "$\circledast$" the circular convolution.

Linear, circular affect filter design?

Lecture 4: Sampling

groß oder klein?

$$X(t) \xrightarrow{\text{sampling}} x[n] \xrightarrow{\text{reconstruction}} X(t)$$



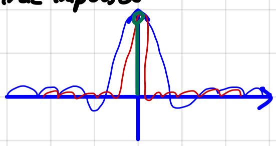
$$X_s(t) = X(t)\cdot S(t) \qquad \left| S(t) = \sum_{n=-\infty}^{\infty} \delta(t-nT_s);\ f_s = 1/T_s \right.$$

$$= \sum_{n=-\infty}^{\infty} x(nT_s)\cdot\delta(t-nT_s)$$

$$= \sum_{n=-\infty}^{\infty} \underbrace{x[n]}_{\text{this are the samples at the n of Dirac Impulses}}\cdot\delta(t-nT_s)$$

because continous time signal

$$S(\Omega) = \int_{-\infty}^{\infty} S(t)\cdot e^{-j\Omega t}dt = \int_{-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\delta(t-nT_s)\cdot e^{-j\Omega t}dt$$



Spectrum of our sampling function

change ok, because both are linear operators

$$= \sum_{n=-\infty}^{\infty}\int_{-\infty}^{\infty}\delta(t-nT_s)\cdot e^{-j\Omega t}dt = \sum_{n=-\infty}^{\infty} e^{-j\Omega n T_s}$$

Dirichlet-Kernel
https://de.wikipedia.org/wiki/Dirichlet-Kern

$$= \frac{2\pi}{T_s}\sum_{k=-\infty}^{\infty}\delta(\Omega-k\Omega_s);\ \Omega_s = \frac{2\pi}{T_s} = 2\pi f_s$$

ist $\Omega_s = T_s$???

is given by this infinite sum of Dirac Impulses, this impulses are spaced apart by Ω_s

From the modulation theorem:

$$X_s(\Omega) = \underbrace{X(\Omega)}_{\text{Spectrum original sample}} \underset{\text{convolve}}{*} \underbrace{S(\Omega)}_{\text{Spectrum of sampling function}}$$

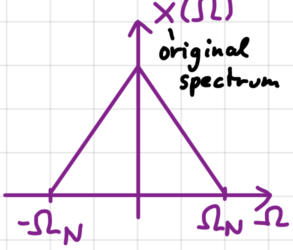
$$= \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\Omega-\Omega')\cdot S(\Omega')\,d\Omega'$$

$$= \frac{1}{T_s}\int_{-\infty}^{\infty} X(\Omega-\Omega')\cdot\sum_{k=-\infty}^{\infty}\delta(\Omega'-k\Omega_s)\,d\Omega'$$

$$= \frac{1}{T_s}\sum_{k=-\infty}^{\infty}\int_{-\infty}^{\infty} X(\Omega-\Omega')\cdot\delta(\Omega'-k\Omega_s)\,d\Omega'$$

$$X_s(\Omega) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X(\Omega-k\Omega_s)\ ;\ \Omega_s = 2\pi f_s$$

Example:



Sketch $X_s(\Omega)$ for

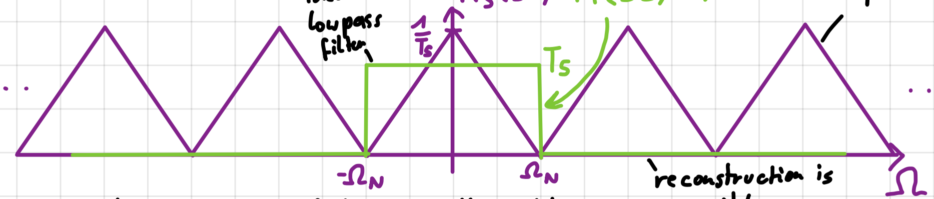
- $\Omega_s = 2\Omega_N$
- $\Omega_s < 2\Omega_N$
- $\Omega_s > 2\Omega_N$

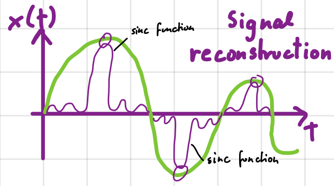
$\Omega_s = 2\Omega_N$: …

ideal lowpass filter

$X_s(\Omega) \cdot H(\Omega) = X(\Omega)$

repetitions do not overlap



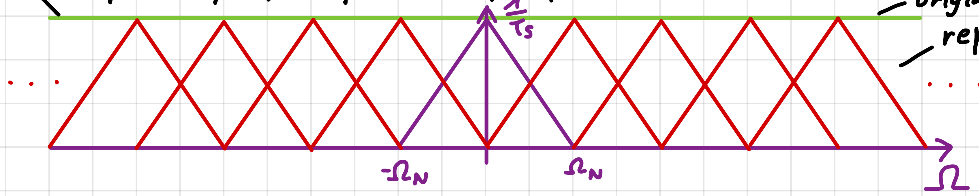


reconstruction is possible

if add all repetitions up we get this flat curve at $1/T_s$ amplitude

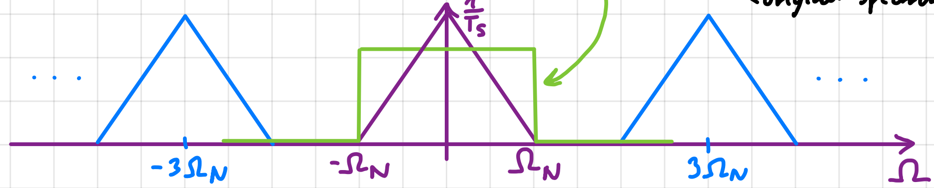
original spectrum is lost (no reconstruction possible)

repetitions do overlap



$X_s(\Omega) \cdot H(\Omega) = X(\Omega)$

original spectrum (reconstruction is possible)



Nyquist-Shannon Sampling Theorem:

tells when the sampling process does not loose any informations

Let $x(t)$ be a bandlimited signal with $X(\Omega) = 0$ for $|\Omega| \geq \Omega_N$.

Then $x(t)$ is uniquely determinal by its samples $x[n] = x(nT_s)$ if $\Omega_s \geq 2\Omega_N$

On the relation of Ω and w:

frequency in physical units — depends on what we measure: • time series → represent in Hz

frequency for discrete time series

$$X_s(\Omega) = \int_{-\infty}^{\infty} x(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \cdot e^{-j\Omega t}\, dt$$

$$= \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} x(t) \cdot \delta(t - nT_s) \cdot e^{-j\Omega t}\, dt$$

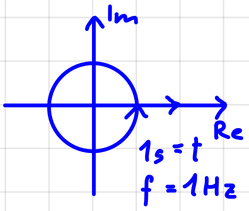
continous time FT — discrete time Fourier Transform

$$= \sum_{n=-\infty}^{\infty} \underbrace{x(nT_s)}_{x[n]} \cdot e^{-j\Omega T_s n} \overset{?}{=} \sum_{n=-\infty}^{\infty} x[n] \cdot e^{-jwn} = X(w)$$

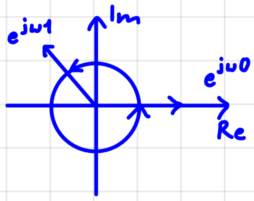
$e^{j\Omega t}$: $t \in \mathbb{R}$; $\Omega = 2\pi f$

$\Omega T_s = w$

e^{jwn}: $n \in \mathbb{N}, w \in [0, 2\pi]$

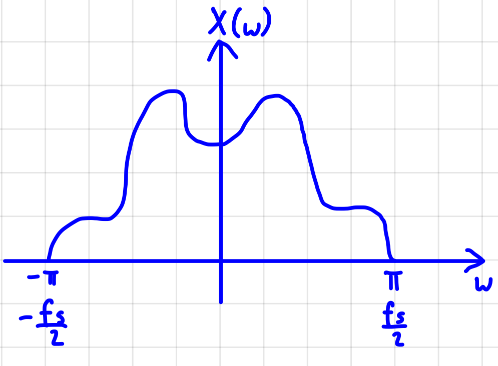


$\rightarrow$ $2\pi f \cdot T_s = w$ $\leftarrow$



$\Leftrightarrow f = \frac{w}{2\pi T_s}$

$\Leftrightarrow \boxed{f = \frac{w \cdot f_s}{2\pi}}$



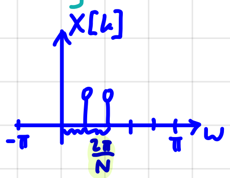
$f|_{w=\pi} = \frac{\pi \cdot f_s}{2\pi} = \frac{f_s}{2}$

$f|_{w=-\pi} = -\frac{f_s}{2}$

Relation of the sampling frequency, window length and the frequency resolution:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j \underbrace{\frac{2\pi}{N}}_{\omega_k} \cdot k \cdot n}$$

space of frequencies



freq. resolution (physical) $\hat{=} \frac{f_s}{N} = \frac{f_s}{T \cdot f_s} = \frac{1}{T}$

T: Length of recording window in physical units

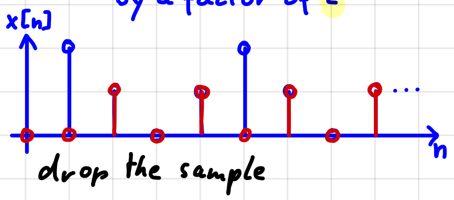
Resampling: Change f_s

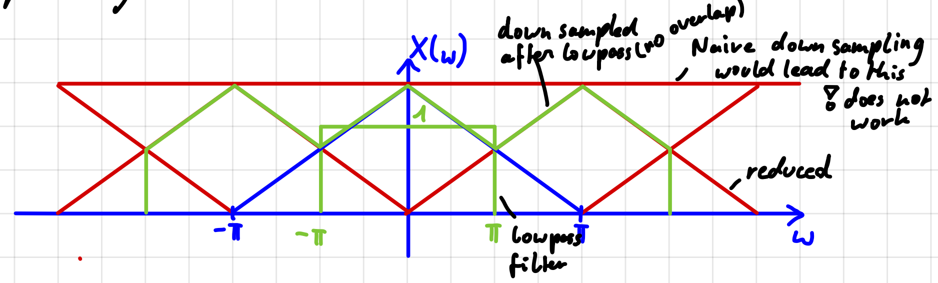
for save up storage space, speed up computations, if signal was sampled with really high sampling frequency, which we don't need and we want to downsample the signal

if recording window is to short then your frequency resolution may be to cause grained to see what is going on in the signal

Downsampling:

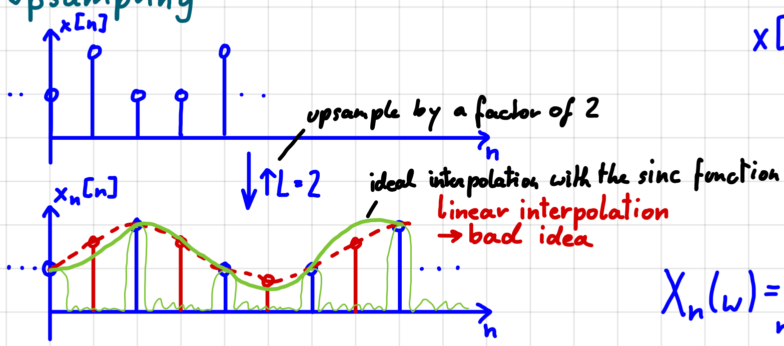
Native down-sampling by a factor of 2





$\Rightarrow$ To downsample by an integer factor M, we first have to apply a lowpass-filter at cut-off frequencies $\pm \frac{\pi}{M}$, and then compute $x_d[n] = x_{lp}[M \cdot n]$.

Upsampling:



$$x[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot \delta[n - k \cdot L]$$

$n = 0$: $x[0]$

$n = 1$: 0

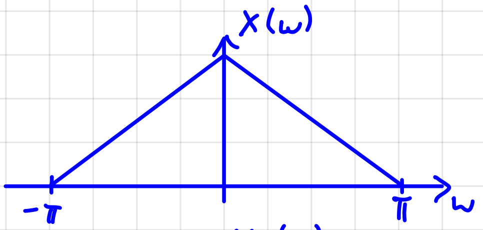
$n = 2$: $x[1]$

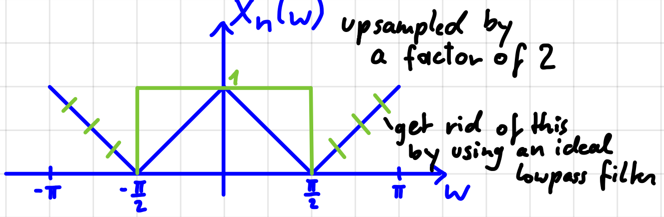
$$X_u(\omega) = \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x[k] \cdot \delta[\underbrace{n - k \cdot L}_{l = n - k \cdot L}] \cdot e^{-j\omega n} \quad \Leftrightarrow \quad n = l + k \cdot L$$

$$= \sum_{k=-\infty}^{\infty} x[k] \sum_{l=-\infty}^{\infty} \delta[l] \cdot e^{-j\omega(l + kL)}$$

$$= \sum_{k=-\infty}^{\infty} x[k] \underbrace{\sum_{l=-\infty}^{\infty} \delta[l] \cdot e^{-j\omega l}}_{1} \cdot e^{-j\omega L \cdot k}$$

$$= \sum_{k=-\infty}^{\infty} x[k] \cdot e^{-j\omega L \cdot k} = X(\omega L)$$





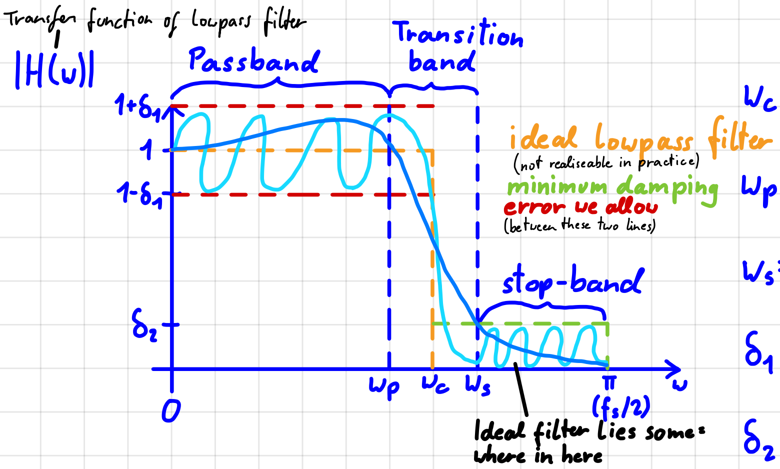
Create $x_u[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot \delta[n - k \cdot L]$ and then apply a low-pass filter with cut-off frequencies $\pm \frac{\pi}{L}$.

Resample bei $\frac{L}{M}$: First upsample by L, then downsample by M.

so we can resample at any arbitrary rational number, with this we can aproximate any irrational number

Lecture 5: Digital Filter Design

Filter specifications: lowpass filter



w_c: cut-off frequency

w_p: end of the pass-band

w_s: start of the stop-band

δ_1: Peak pass-band ripple $\quad -20 \cdot \log_{10}(1-\delta_1)\ [dB]$

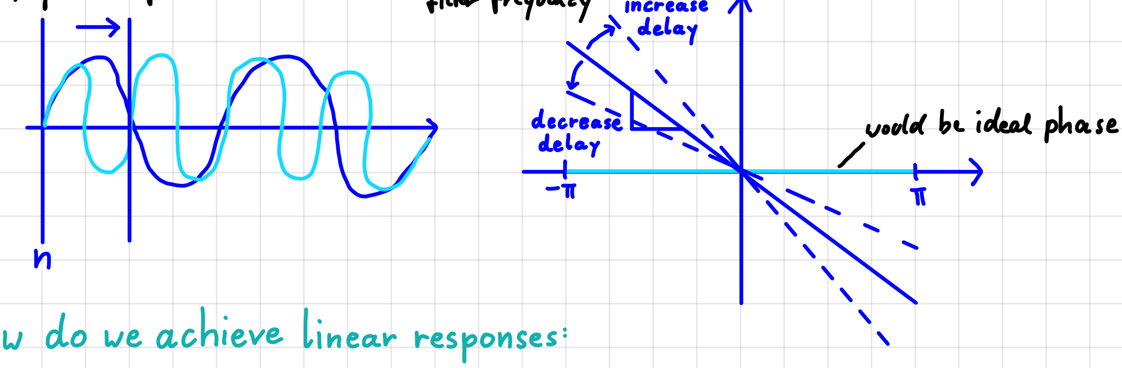
δ_2: Peak stop-band ripple $\quad -20 \cdot \log_{10}(\delta_2)\ [dB]$

Ideal filters have zero or linear phase responses! Why?

$$h_{id}[n] = \delta[n-n_d] \Leftrightarrow H(w) = \sum_{n=-\infty}^{\infty} \delta[n-n_d] \cdot e^{-jwn} = e^{-jwn_d}$$

impulse response

filter frequency response



How do we achieve linear responses:

1.) Use a finite impulse response filter (FIR-filter), because they have a linear phase response.
better than filter properties in therms of the amplitude response

2.) If you need to use an infinite impulse response filter (IIR-filter), use forward-backward filtering, e.g., us implemented in filtfilt.m in Matlab.

filter signal twice with the same filter, once with a time forward direction and then flipp the signal and filter than with a time backward direction
⇒ is flipping the phase (any distortion that we get after forward are eliminated by backward ⇒ zero phase filter, only works with offline data

How to design FIR-filters?

1.) Filter design by windowing:

- Analytically specify the desired frequency response, e.g,

$$H_{lp}(w) = \begin{cases} 1; & |w| \leq w_c \\ 0; & \text{otherwise} \end{cases}$$

ideal filter

- Apply the inverse DTFT to obtain the impulse response, e.g,

$$h_{lp}[n] = \text{sinc}[w_c n]$$

Problem: filter is infinitly long and acausal!

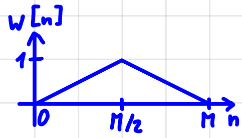
- Use a window function to obtain a finite-length filter, e.g.,

$$\underbrace{h'_{lp}[n]}_{\text{finite filter}} = \underbrace{h_{lp}[n]}_{\text{infinite filter}} \cdot \underbrace{w[n]}_{\text{finite window}}$$

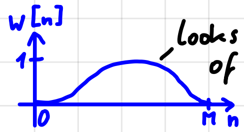
Typical windows:

Rectangular: $w[n] = \begin{cases} 1; & 0 \leq n < M \leftarrow \text{length of your filter} \\ 0; & \text{otherwise} \end{cases}$

Bartlett: $w[n] = \begin{cases} 1 - \left| \frac{2n}{M} - 1 \right|; & 0 \leq n < M \\ 0 & ; \text{otherwise} \end{cases}$



Hann: $w[n] = \begin{cases} 0{,}5 - 0{,}5 \cdot \cos(2\pi n / M); & 0 \leq n < M \\ 0 & ; \text{otherwise} \end{cases}$



looks like one half of a cos

Hamming: $w[n] = \begin{cases} 0{,}54 - 0{,}46 \cdot \cos(2\pi n / M); & 0 \leq n < M \\ 0 & ; \text{otherwise} \end{cases}$

- Delay $h'_{(lp)}[n]$ to obtain a casual filter.

2.) FIR filter design by least-squares optimization (firls.m): Mathlab file

- Choose the filter length $N+1$ with N even. ?

- Specify the desired filter frequency response $H(\omega_i)$

with $\omega_i = \frac{\pi}{k} \cdot i$ with $i = 0, \dots, k-1$; $k \gg N$

$$H(\omega_i) = \sum_{n=-N/2}^{N/2} h[n] \cdot e^{-j\omega_i \cdot n}$$

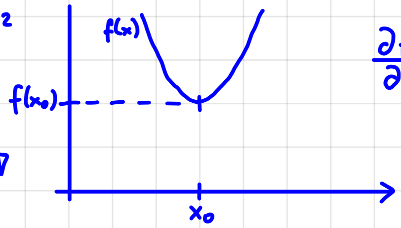
To obtain a zero-phase filter, i.e., with no delay, we want <u>the imaginary parts of $H(\omega)$ to be zero.</u>

Euler's formula $\Rightarrow H(\omega_i) = \sum_{n=-N/2}^{N/2} h[n] \cdot (\cos(\omega_i \cdot n) \cancel{- j \cdot \sin(\omega_i \cdot n)})$

$$= h_0 + \sum_{n=1}^{N/2} h[n] \cdot 2 \cdot \cos(\omega_i \cdot n)$$

$$\underbrace{\begin{bmatrix} H(\omega_0) \\ H(\omega_1) \\ \vdots \\ H(\omega_k) \end{bmatrix}}_{=: \underline{d}} = \underbrace{\begin{bmatrix} 1 & 2 \cdot \cos(\omega_0 \cdot 1) \dots 2\cos(\omega_0 \cdot N/2) \\ 1 & 2 \cdot \cos(\omega_1 \cdot 1) \dots 2\cos(\omega_1 \cdot N/2) \\ \vdots & \vdots \quad\quad \vdots \\ 1 & 2 \cdot \cos(\omega_k \cdot 1) \dots 2\cos(\omega_k \cdot N/2) \end{bmatrix}}_{=: A} \cdot \underbrace{\begin{bmatrix} h[0] \\ \vdots \\ h[N/2] \end{bmatrix}}_{\underline{h}}$$

$$\underline{h}^* = \underset{\underline{h}}{\operatorname{argmin}} (\underline{d} - A \cdot \underline{h})^2$$



$$\frac{\partial f(x)}{\partial x} = 0$$

$$\frac{\partial}{\partial \underline{h}} (\underline{d} - A \cdot \underline{h})^2 \overset{!}{=} 0 \rightarrow \text{Solve for } \underline{h}$$

$$\frac{\partial}{\partial \underline{h}} (\underline{d} - A \cdot \underline{h})^T (\underline{d} - A\underline{h}) = \frac{\partial}{\partial \underline{h}} (\underline{d}^T \underline{d} - \underline{d}^T A \underline{h} - \underline{h}^T A^T \underline{d} + \underline{h}^T A^T A \underline{h})$$

$$= 0 - d^T A - d^T A + 2 \underline{h}^T A^T A \overset{!}{=} 0$$

$$\Leftrightarrow \cancel{2} \underline{h}^T A^T A = \cancel{2} \underline{d}^T A \quad | (A^T A)^{-1}$$

$$\Leftrightarrow \underline{h}^T = \underline{d}^T A (A^T A)^{-1}$$

$$\Leftrightarrow \boxed{\underline{h}^* = \underbrace{(A^T A)^{-1} \cdot A^T}_{\text{Moore-Penrose Pseudo-Inverse}} \cdot \underline{d}}$$

no imaginary part

acausal filter, to turn into causal filter, need to delay it by $N/2$

- Mirror $h[n]$ to $h[-n]$ to obtain the desired impulse response.

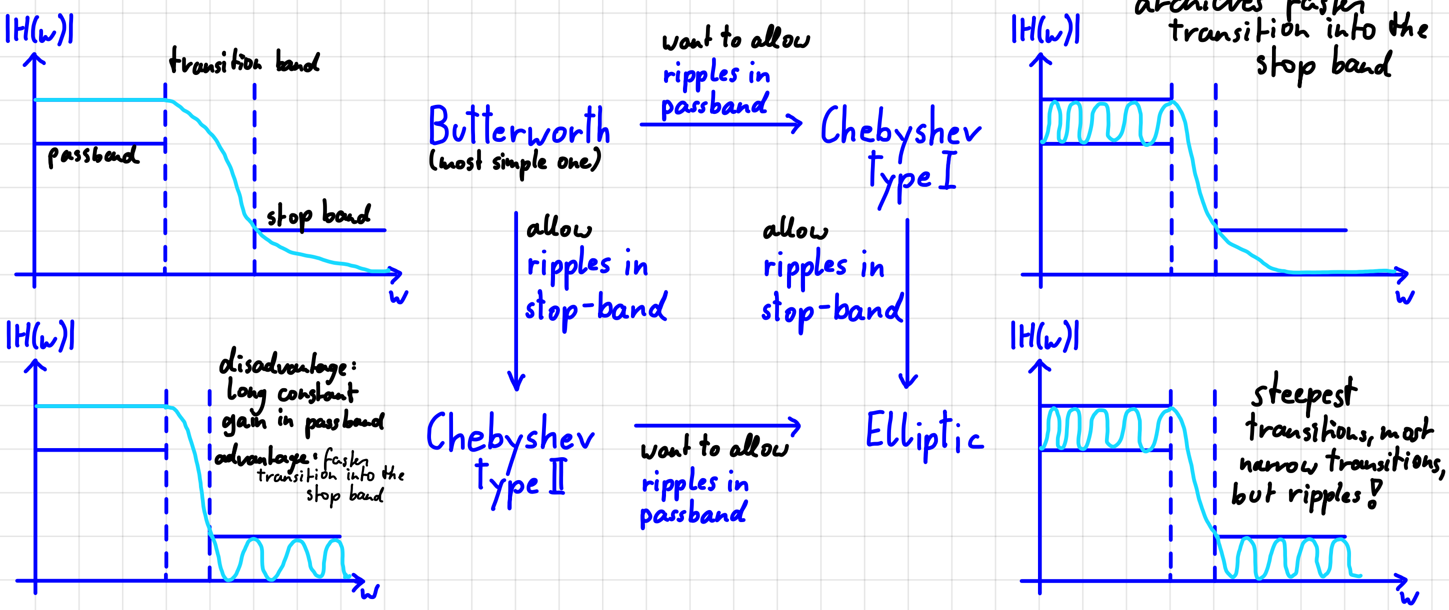
Infinite impulse response (IIR filters):

$$a[0] \cdot y[0] = b[0] \cdot x[n] + b[1] \cdot x[n-1] + \ldots + b[p] \cdot x[n-p] - a[1] \cdot y[n-1] - \ldots - a[q] \cdot y[n-q]$$

Notes:

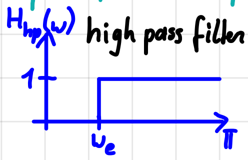
- IIR filters may be unstable
- IIR filters do not, in general, have a linear phase — they may delay or typical do delay different frequencies by different samples, can lead to very strange results
  (→ for offline data use forward-backward filtering, if possible)
- IIR filters typically have steeper transition bands than FIR filters

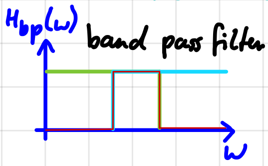
There are 4 prototypical IIR filters:

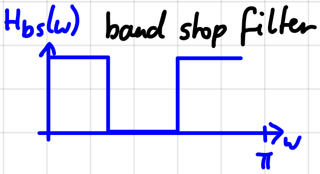


From low-pass to high-pass, band-pass and band-stop filters:

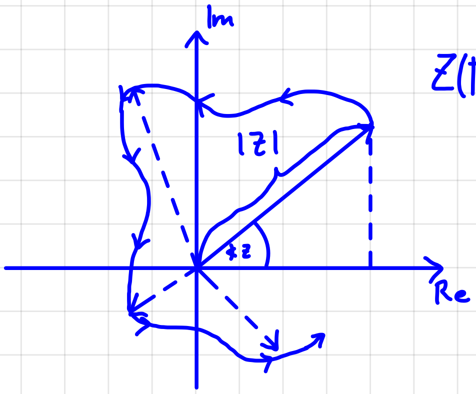
- $H_{hp}(\omega) = 1 - H_{lp}(\omega)$
- $H_{bp}(\omega) = H_{lp}(\omega) \cdot H_{hp}(\omega)$
- $H_{bs}(\omega) = 1 - H_{bp}(\omega)$







Lecture 6: The hilbert Transform



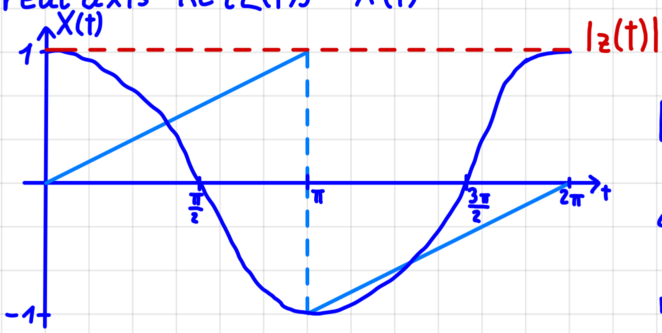
$$Z(t) = |Z(t)| \cdot e^{j\measuredangle Z(t)}$$

Instantaneus amplitude : $|Z(t)|$

Instantaneus phase : $\measuredangle Z(t)$

Instantaneus frequency : $\frac{d\measuredangle Z(t)}{dt}$

Assume that $|Z(t)|$ and $\frac{d\measuredangle Z(t)}{dt}$ are constant and consider the projection of $Z(t)$ onto the real axis: $\mathrm{Re}\{Z(t)\} =: X(t)$



By the hilbert transform, we aim to reconstruct a complex-valued signal from its projection to the real axis?

We want to reconstruct the analytic signal,

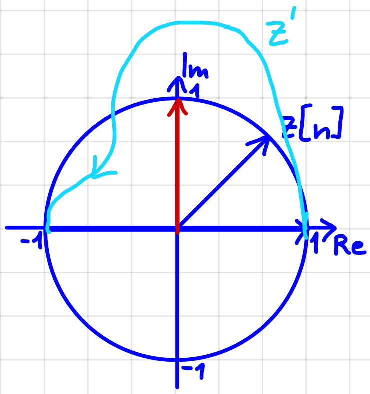
$$z[n] = \underbrace{z_r[n]}_{=x[n]} + j \cdot z_i[n]$$

We want to „reconstruct" $z_i[n]$ from $z_r[n] = x[n]$.

How do we choose $z_i[n]$?

Consider $z_r[n] = x[n] = A \cdot \cos(w \cdot n) = \mathrm{Re}\{|z| \cdot e^{jwn}\}$

$$z_i[n] = A \cdot \sin(w \cdot n) = A \cdot \cos(w \cdot n - \tfrac{\pi}{2})$$



By delaying $x[n]$ by $\frac{\pi}{2}$, we can „reconstruct" an analytic signal that, for $x[n]$ a pure oscillation, has constant amplitude and linear phase ($\hat{=}$ constant frequency).

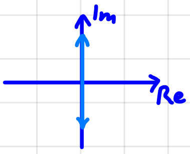
Extend this idea to signals composed of multible oscillations:

$$z[n] = x[n] + j z_i[n]$$

$$\overset{DTFT}{\Longleftrightarrow} \quad Z(w) = X(w) + j Z_i(w)$$

$$= X(w) + j \overbrace{H(w) \cdot X(w)}$$

$H(w)$?



$$H(w) = \begin{cases} e^{-j\frac{\pi}{2}} = -j \; ; & 0 \leq w \leq \pi \\ e^{j\frac{\pi}{2}} = j \; ; & -\pi < w < 0 \end{cases}$$

Practical implementation:

Note that

$$Z(w) = \begin{cases} 2X(w) \; ; & 0 \leq w \leq \pi \\ 0 \; ; & -\pi < w < 0 \end{cases}$$

1.) Band-pass filter $x[n]$ in the frequency range of interest.

2.) Compute $X(w)$ via the D(T)FT.

3.) Set all $X(w) = 0$ for all $w < 0$, and rescale $X(w)$ by a factor of two for all $w \geq 0$.

4.) Compute the inverse D(T)FT to obtain the analytic signal $z[n]$.

Instantaneus amplitude : $|z[n]|$

Instantaneus phase : $\measuredangle z[n]$

Instantaneus frequency : $\frac{d \measuredangle z[n]}{dn}$

Lecture 7: The Discrete Cosine Transform (DCT)

— one of the major parts of jpeg image format to compress images

frequencies · Normalisation factor · projection of original signal onto our basis

Analysis: $X[k] = 2\sum_{n=0}^{N-1} x[n]\cdot\underbrace{\cos\left(\frac{\pi}{2N}\cdot k\cdot(2n+1)\right)}_{\text{basis functions indexed by } k};\ 0 \le k \le N-1$

(time to frequency domain)

Synthesis: $x[n] = \frac{1}{N}\sum_{n=0}^{N-1} X'[k]\cdot\cos\left(\frac{\pi}{2N}\cdot k\cdot(2n+1)\right);\ 0 \le n \le N-1$

(frequency to time domain)

where $X'[k] = \begin{cases} X[0]/2 & ; k=0 \\ X[n] & ; \text{otherwise} \end{cases}$

Similar Transform to FT, it also computes the spectrum

It uses a difference set of $\underbrace{\text{basis functions}}_{\text{cosins}}$ compared to FT,

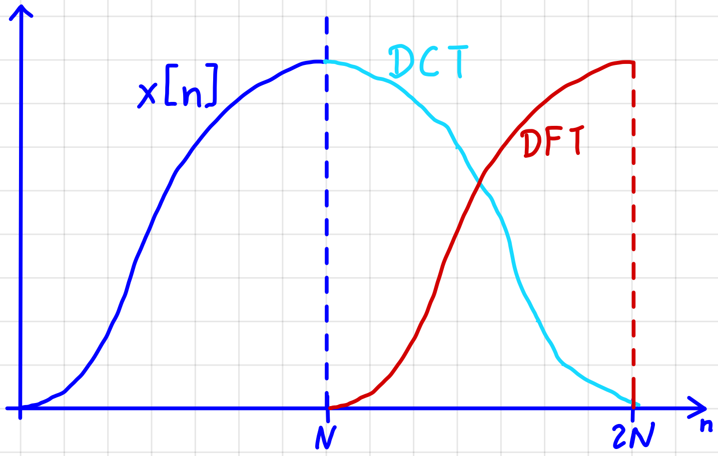
cosins → advantage will assume a different type of periodicity

other assumtion: it is very benefital for representing images (in terms of spectra components) and compressing images

FT: uses complex exponentiells → assums that signals are periodic

Remarks:

- The DCT is an invertible transformation.
  To show that: prove the basises are orthonormal
- There are variants of the DCT: I-IV (the one above is the DCT-II).



→ The DCT extends/repeats signals symmetrically, while the DFT extends signals periodically.

The former is particular for representing images, which are typically not periodic.

```
In [1]: # import packages
        import numpy as np
        import matplotlib.pyplot as plt
        import imageio
        import scipy.fftpack

In [2]: # define DFT and DCT

        def dft(N):

            k = np.arange(0, N)
            n = np.arange(0, N)

            A = np.exp(-1j*2*np.pi/N * np.outer( k, n))

            return A

        def dct(N):

            k = np.arange(0, N)
            n = np.arange(0, N)

            A = np.cos( np.pi / 2 / N * np.outer( k, 2*n+1))

            return A

In [12]: # inspect basis functions

         N = 100

         DCT = dct(N)
         DFT = dft(N)

         plt.figure( figsize = (20,10))
         plt.subplot(2,1,1)
         plt.plot( np.real(DFT[0:5,:]).T)
         plt.grid()
         plt.subplot(2,1,2)
         plt.plot( DCT[0:5,:].T)
         plt.grid()
```

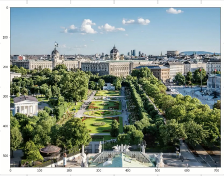
basis functions of DFT (first five ones)

all end with 1
n=100 == n=0
any signal that is transformed or modelled will be periodic, because the basis functions are periodic => sum is also periodic

basis functions of DCT (first five ones)

both start at [0]=1

```
In [16]: # read image
         image = imageio.imread('vienna.jpg')
         plt.figure( figsize = (20,10))
         plt.imshow(image)

Out[16]: <matplotlib.image.AxesImage at 0x7ffaf3700cc0>
```



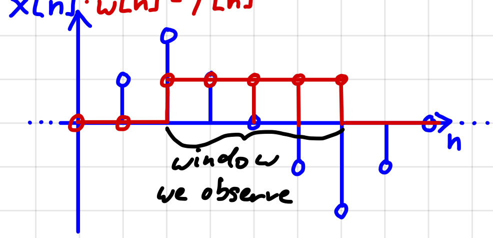
```
In [21]: # read image
         image = imageio.imread('vienna.jpg')
         R = image[:,:,0]
         G = image[:,:,1]
         B = image[:,:,2]
         z = 0.299*R + 0.587*G + 0.114*B # convert to grayscale
         plt.figure( figsize = (20,10))
         plt.imshow( z, cmap = 'gray')

Out[21]: <matplotlib.image.AxesImage at 0x7ffaf3848dd8>
```

```
In [25]: # apply 2d dct for compression

         Z = scipy.fftpack.dct( z, axis = 0)
         Z = scipy.fftpack.dct( Z, axis = 1)

         # create and apply a compression mask
         M = 300  ← changing this for sharpness
         mask = np.zeros( Z.shape)
         mask[0:M,0:M] = 1
         Z = Z * mask

         # inverse 2d dct
         iz = scipy.fftpack.idct( Z, axis = 0)
         iz = scipy.fftpack.idct( iz, axis = 1)

         print('Compression ratio:' + str( M**2 / (Z.shape[0]*Z.shape[1]) ))

         plt.figure( figsize = (20,10))
         plt.imshow( iz, cmap = 'gray')

Compression ratio:0.2331002331002331

Out[25]: <matplotlib.image.AxesImage at 0x7ffaf390ee10>
```

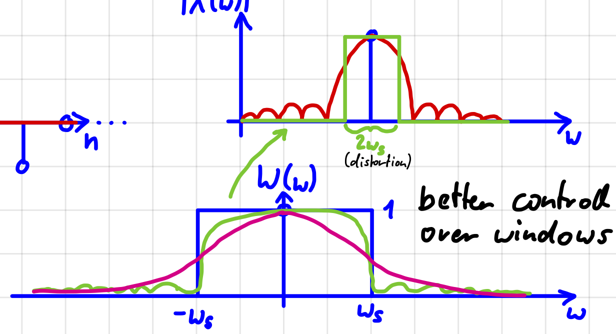
Lecture 8: Multitapers

special class of windows

$x[n] \cdot w[n] = Y[n]$



$X(\omega) * W(\omega) = Y(\omega)$



Idea is: to control the particular shape of the window that is used for power spectral estimation

We want to design a set of optimal windows, in the sense that we do a smoothing/averaging of the original spectrum over a bandwith of $2 \cdot \omega_s$. As $w[n]$ has to be finite length $(N+1)$, the above frequency is possible to realize. To find the best approximation, we consider the following optimization problem:

$$\boxed{\underset{w[n]}{\operatorname{argmax}} \int_{-\omega_s}^{\omega_s} \overbrace{|W(\omega)|^2}^{\text{spectrum of window}} d\omega} \quad \text{s.t.} \quad \underbrace{\|\underline{w}\|_2 = 1}_{*}$$

impulse response (time domain)

constant

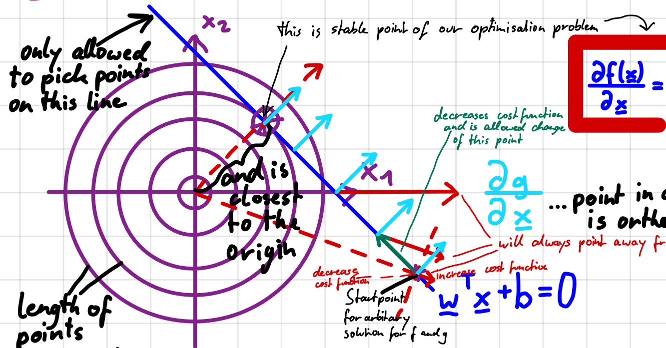
that means, that our window function W should concentrate its energy in the band from $-\omega_s$ to ω_s and enhance to minimise the energy outside of it.
=> hope to get a good aproximation to regtangular window (has all energy in the given range)
constrained that the energy is finite and concentrated in this band,
shape of the window does not depend on the constraint

Review: Constrained Optimization

$$* \, g(\underline{w}) = \|\underline{w}\|_2 - 1 \overset{!}{=} 0 = \\ = \sqrt{\underline{w}^T \underline{w}} - 1 \overset{!}{=} 0$$

$$\underset{\underline{x}}{\operatorname{argmin}} \, f(\underline{x}) \quad \text{s.t.} \quad \underbrace{g(\underline{x}) = 0}_{\text{constrained}}$$

Example: $\underset{\underline{x}}{\operatorname{argmin}} \|\underline{x}\|_2$ s.t. $\underline{w}^T \underline{x} + b = 0$ ... linear constrained



$$\boxed{\frac{\partial f(\underline{x})}{\partial \underline{x}} = \lambda \frac{\partial g(\underline{x})}{\partial \underline{x}}}$$

$\lambda \in \mathbb{R}$ says that we do not care about the length of the gradients (just want them to point in same direction

$\frac{\partial g}{\partial \underline{x}}$ ... point in direction were change is the biggest is orthogonal to the constraint

will always point away from origin (so it turns by moving over the constraint axes)

$$L(\underline{x}, \lambda) = f(\underline{x}) + \lambda g(\underline{x})$$

cost function; parameters we want to optimize over; function we want to minimize; ↑ lagrange multiplier

$$\frac{\partial}{\partial \underline{x}} L(\underline{x}, \lambda) \overset{!}{=} 0 \iff \frac{\partial f(\underline{x})}{\partial \underline{x}} = -\lambda \frac{\partial g(\underline{x})}{\partial \underline{x}} = \lambda' \cdot \frac{\partial g(\underline{x})}{\partial \underline{x}}$$

$$\frac{\partial}{\lambda} L(\underline{x}, \lambda) \overset{!}{=} 0 \iff g(\underline{x}) = 0$$

$\square$

$$\int_{-\omega_s}^{\omega_s} |W(\omega)|^2 d\omega \overset{FT}{=} \int_{-\omega_s}^{\omega_s} \Big(\sum_{n=-\frac{N}{2}}^{N/2} w[n] \cdot e^{-j\omega n} \Big)^{*} \cdot \Big(\sum_{m=-\frac{N}{2}}^{N/2} w[m] \cdot e^{-j\omega m} \Big) d\omega$$

complex conjugate

$$= \sum_{n=-\frac{N}{2}}^{N/2} \sum_{m=-\frac{N}{2}}^{N/2} w[n] \cdot w[m] \cdot \int_{-\omega_s}^{\omega_s} e^{-j\omega(m-n)} d\omega$$

$$= \sum_{n=-\frac{N}{2}}^{N/2} \sum_{m=-\frac{N}{2}}^{N/2} w[n] \cdot w[m] \cdot \left[\frac{1}{j(m-n)} \cdot e^{-j\omega(m-n)} \right] \Big|_{-\omega_s}^{\omega_s}$$

$$= \sum_{n=-\frac{N}{2}}^{N/2} \sum_{m=-\frac{N}{2}}^{N/2} w[n]\cdot w[m] \cdot \frac{e^{j\omega_s(m-n)} - e^{-j\omega_s(m-n)}}{j(m-n)}$$

$$= \sum_{n=-\frac{N}{2}}^{N/2} \sum_{m=-\frac{N}{2}}^{N/2} w[n]\cdot w[m] \cdot \frac{2\cdot \sin(\omega_s(m-n))}{m-n}$$

Let $\underline{w} := [W[-\frac{N}{2}], \dots, W[\frac{N}{2}]]$ and $A_{m,n} = \frac{2\cdot\sin(\omega_s(m-n))}{m-n}$: (row, colour)

Then, $\underbrace{\underset{w[n]}{\operatorname{argmax}} \int_{-\omega_s}^{\omega_s} W(\omega)^2 d\omega}_{\text{initial cost function}} = \underbrace{\underset{\underline{w}}{\operatorname{argmax}} \{\underline{w}^T \cdot A \cdot \underline{w}\}}_{\text{simplified cost function}}$ s.t. $\underbrace{\|W\|_2}_{\text{Norm of impulse response}} = 1$

$g(\underline{w}) = \|\underline{w}\|_2 - 1 \overset{!}{=} 0 =$
$= \sqrt{\underline{w}^T\underline{w}} - 1 \overset{!}{=} 0$

$\Rightarrow \quad L(\underline{w}, \lambda) = \underline{w}^T A \underline{w} + \lambda(\sqrt{\underline{w}^T\underline{w}} - 1) \quad \Big| \ \sqrt{\underline{w}^T\underline{w}} = (\underline{w}^T\underline{w})^{\frac{1}{2}}$

$\frac{\partial}{\partial \underline{w}} L(\underline{w}, \lambda) = 2A\underline{w} + \lambda\underline{w} \overset{!}{=} 0 \Leftrightarrow A\underline{w} = \underbrace{-\lambda}_{-\lambda =: \lambda'}\underline{w}$

Eigenvectors of a symmetric matrix are orthogonal.

Matrix — Eigenvalues (corresponding)

$\Leftrightarrow \boxed{A\underline{w} = \lambda'\underline{w}}$ all w that fullfill this equation are solutions to our optimizations problem

rescalling length of w (no change of orientation

Eigenvectors of A (set of windows, that are solutions to the problem

$\underline{w}^T A \underline{w} = \lambda' \underline{w}^T \underline{w} = \lambda'$

Eigenvalues give value of the cost function

Eigenvalue of an eigenvector gives amount of energy that it puts into that
$\Rightarrow$ can be sorted, to get Eigenvector with the greatest amount of energy put in

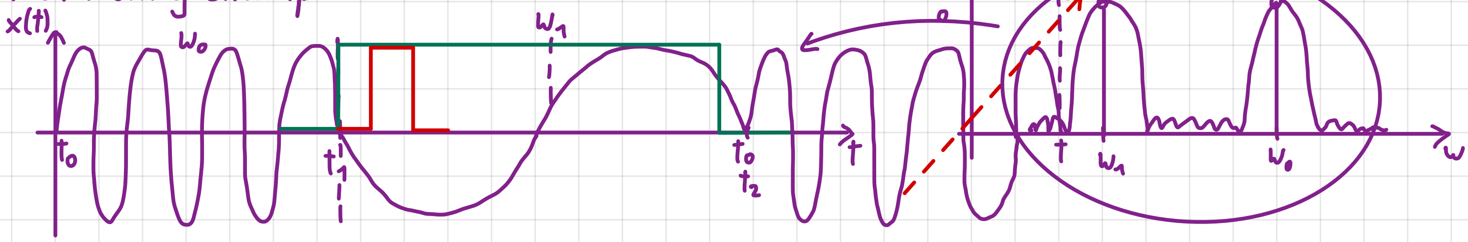
The solutions to the eigenvalue problem are called the discrete prolate spheroidal (DPS) sequences or Slepians.

The multitaper spectral estimate is given by:

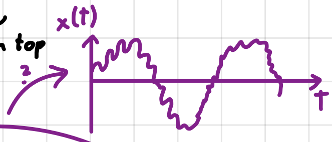
$$|X(\omega)| = \frac{1}{K}\sum_{k=1}^{K} \left| \sum_{n=-\frac{N}{2}}^{N/2} w_k[n]\cdot x[n]\cdot e^{-j\omega n} \right|$$

Lecture 9: Wavelets

Motivating example:



Looks like fast and slow oscillation on top of each other ?



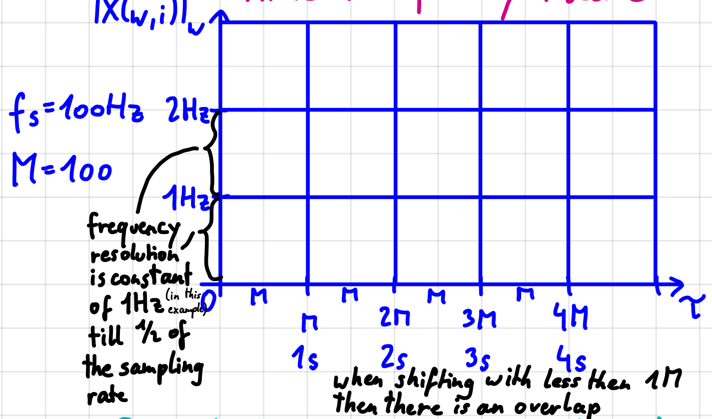
Idea: Perform the DTFT in multiple (moving and potentially overlapping) windows

→ Short-time Fourier Transform (STFT)

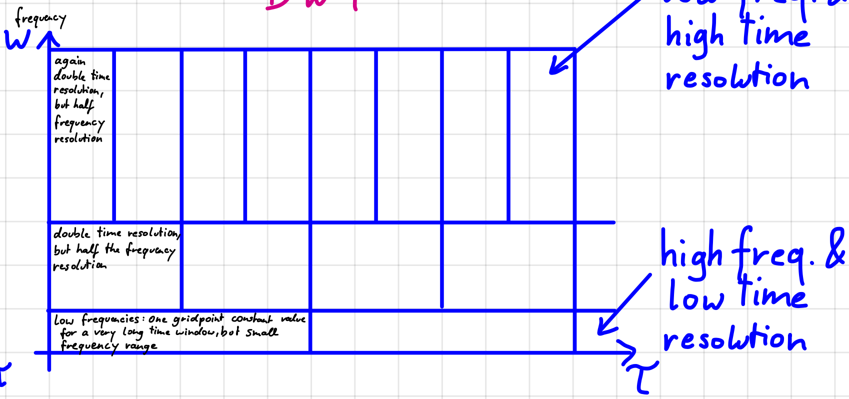
$$X(w,\tau) = \sum_{n=-\infty}^{\infty} \underbrace{w[n-\tau]}_{\text{window}} \cdot x[n] \cdot e^{-jwn}$$

for some window $w[n]$ of length M.

Time-Frequency Plane



DWT



very slow oscillations also change amplitude slowly
fast oscillations change quickly

The Discrete Wavelet Transform (DWT):

Given a mother wavelet $h[n]$ of length M, we define

$$h_{m,k}[n] = \underbrace{2^{-\frac{m}{2}}}_{\text{Normalisation constant}} \cdot h[\overbrace{2^{-m}}^{\text{scaling the mother wavelet (streches out the wavelet)}} \cdot n - \underbrace{k \cdot 2^{-m} \cdot M}_{\text{shift of the wavelet}}]; \quad m,k \in \mathbb{N}$$

The DWT is given by

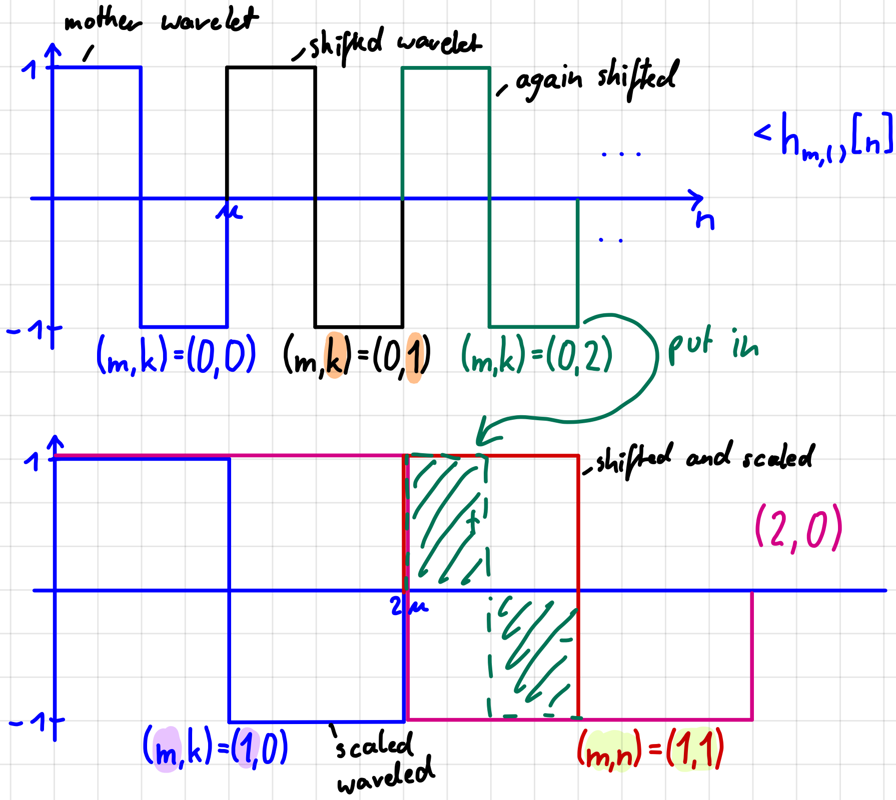
$$\underbrace{X[\overset{\text{frequency}}{m},\overset{\text{position}}{k}]}_{\text{wavelet domain}} = 2^{-\frac{m}{2}} \cdot \sum_{n=-\infty}^{\infty} x[n] \cdot h[2^{-m} \cdot n - k \cdot 2^{-m} \cdot M]$$

How do we design a mother wavelet s.t. $\underbrace{\langle h_{m,k}[n], h_{m',k'}[n] \rangle}_{\text{inner product}} = \delta_{m \cdot m'} \cdot \delta_{k \cdot k'}$?

inner product of two mother wavelets (two basis functions) is **zero** (inner product is orthogonal) exept we project the mother wavelet onto its own duplicate, than we want it to be **one** ⇒ scaling and position has the same value

The Haar wavelet:

$$h[n] = \begin{cases} 1; & 0 \le n \le \frac{M}{2} \\ -1; & \frac{M}{2} < n < M \\ 0; & \text{otherwise} \end{cases}$$



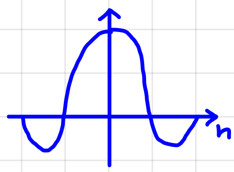
$$\langle h_{m,l}[n], h_{m,l}[n] \rangle = 0$$

$$\langle h_{(0,0)}[n], h_{(1,0)}[n] \rangle$$

$\Rightarrow$ Any set of N pairs of (m,k) gives an orthonormal basis $h_{m,k}[n]$!

Mexican hat / Richert wavelet:

$$h[n] = \frac{2}{\sqrt{3\delta}\cdot\pi^{1/4}}\left(1-\left(\frac{n}{\delta}\right)^2\right)\cdot e^{-\frac{n^2}{2\delta^2}}$$



The Mexican hat wavelet gives rise to an orthonormal basis but has infinite support.

Daubechies wavelet:

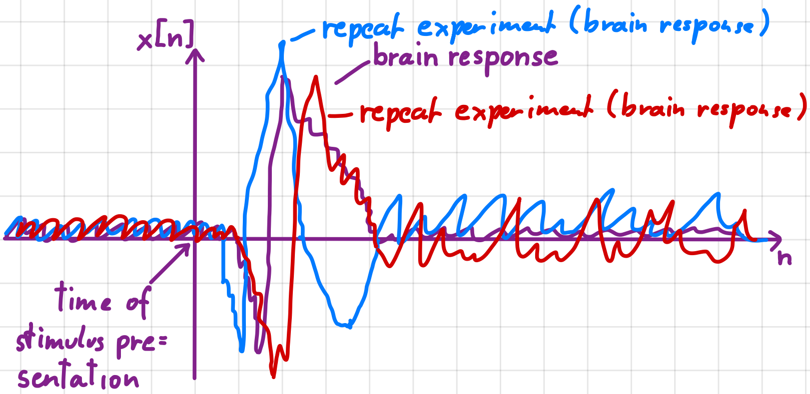
$\underbrace{\text{Compact support}}_{\text{so finite length}}$, gives rise to an orthonormal basis, but has no analytic expression.

Lecture 10: Stochastic processes

In this lecture, we consider the time-series $x[n]$, $n\in\{1,\dots,N\}$, to be one realization of a discrete-time random process $\{x_n, n\in\{1,\dots,N\}\}$ of N random variables (r.v.s.) with a probability density function (pdf)

$$p_{\{x_n\}}(x_1,\dots,x_N).$$

Example: Consider k repetitions of a brain imaging experiment, where a subject's response to a visual stimulus is measured by electroencephalogram (EEG).



Our goal is to estimate the spectrum of the random process from its realizations. Naively, we could so by first estimating the spectrum of each realization, and then averaging across all estimated spectra. In the following, we will study if (or under which conditions) this is a sensible idea.

Because estimating high-dimensional pdfs is a difficult task, we typically introduce some simplifying assumptions:

Gaussianity: We often assume that $\{x_n\}$ is multivariate Gaussian, i.e.,

$$p(\underset{\text{random variables}}{\underline{x}})=\underbrace{(2\pi)^{-\frac{N}{2}}}_{\text{for normalizing}}\cdot|\Sigma|^{-\frac{1}{2}}\cdot\exp\{-\tfrac{1}{2}(\underline{x}-\underset{\text{mean vector}}{\underline{\mu}})^T\Sigma^{-1}(\underline{x}-\underline{\mu})\},\ \text{where}$$

$\underline{x}:=[x_1,\dots,x_N]^T$, $\underline{\mu}:=[\underset{\text{mean of }x_1}{\mu_1},\dots,\underset{\text{mean of }x_n}{\mu_N}]$ the mean of each r.v., and $\Sigma\in\mathbb{R}^{N\times N}$ the covariance matrix of $\underline{x}$.

In practice, we are often unable to observe multiple realizations of $\{x_n\}$. (run experiments multiple times) To facilitate estimation of $p\{x_n\}$ from one realization $(X[1],\dots,X[N])$, we often assume the process to be stationary:

$$\forall\underset{\text{time shifts}}{\tau}\in\underset{\text{integers}}{\mathbb{Z}}:\ P_{\{x_{n+\tau}\}}=P_{\{x_n\}}.$$

Note that for a stationary multivariate Gaussian random process we have

$$E\{x_n\}=\mu_1=\dots=\mu_N=:\mu\ \text{(mean)}$$

expected value is identical to all the other ones (across time)

$$E\{(x_\mu-\mu)^2\}=\sigma_1^2=\dots=\sigma_N^2=:\sigma^2\ \text{(variance)}$$

and

$$r_\tau = E\{(x_{n+\tau}-\mu)(x_n-\mu)\} \quad (= E\{x_{n+\tau}\cdot x_n\}-\mu^2)$$

the covariance sequence of the process.

We can then estimate all parameters of the multivariate Gauss. r.p. as follows:

estimated value

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} x[n] \quad ; \quad \hat{\sigma}^2 = \frac{1}{N-1}\sum_{n=1}^{N}(x[n]-\hat{\mu})^2$$

After subtracting the mean from each observation, i.e., $\tilde{x}[n] = x[n]-\hat{\mu}$, we estimate the covariance sequence as:

$$\hat{r}_\tau = \frac{1}{N-\tau}\sum_{n=1}^{N-\tau}\tilde{x}[n+\tau]\cdot\tilde{x}[n].$$

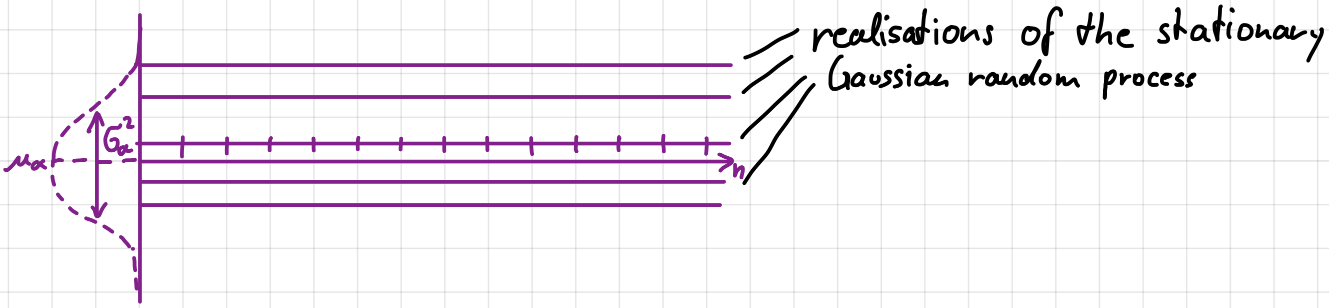
because of

time shift

We call a random process whose parameters can be estimated from one realization ergodic.

Example of a non-ergodic stationary Gaussian random process:

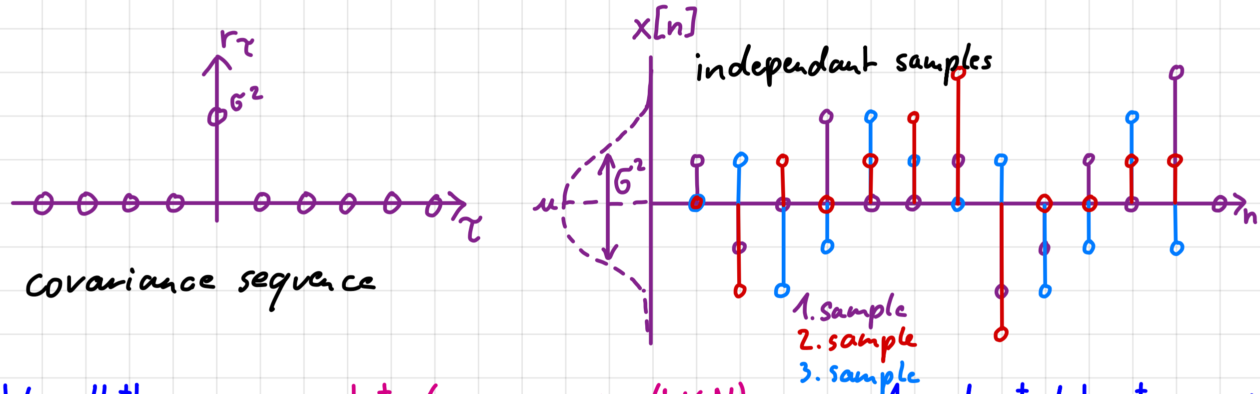
Let $\alpha \sim N(\mu_\alpha, \sigma_\alpha^2)$ and $x_n = \alpha$ for all n.



Example of an ergodic stationary Gaussian random process:

statistical independent

Let $x_n \sim N(\mu, \sigma^2)$ with $x_i \perp\!\!\!\perp x_j \;\; \forall i,j \in \{1,\dots,N\}$ if $i \neq j$. Then $r_\tau = \delta[\tau]\cdot\sigma^2$



We call this process a white Gaussian noise (WGN) process. As indicated by its covariance sequence, the WGN process has no temporal structure. In the next lecture, we explore relationship between the covariance sequence and the spectrum of a random process.

Lecture 10b: Spectral estimation of stochastic processes

Consider the squared amplitude spectrum of one realization of the random process:

$$\frac{1}{N}|X(\omega)|^2 = \frac{1}{N} X^*(\omega)\cdot X(\omega)$$

$$= \frac{1}{N} \overset{\text{realisation}}{\sum_{n=1}^{N}} x[n]\cdot e^{+j\omega n} \cdot \sum_{m=1}^{N} x[m]\cdot e^{-j\omega m}$$

$$= \frac{1}{N}\sum_{n=1}^{N}\sum_{m=1}^{N} x[n]\cdot x[m]\cdot e^{-j\omega(m-n)} \qquad | \, l := m-n$$

$$= \sum_{l=-(N-1)}^{N-1} \underbrace{\frac{1}{N}\sum_{n=1}^{N-|l|} x[n]\cdot x[n+|l|]}_{\hat{r}_l' \text{ - bias estimate of covariance sequence}}\cdot e^{-j\omega l} \qquad | \text{reordering}$$

$$= \sum_{l=-(N-1)}^{N-1} \hat{r}_l' \cdot e^{-j\omega l} =: S_N(\omega) \qquad \text{(periodogram)}$$

Take DTF of the covariance sequence, gives us the same estimates

We call $\lim_{N\to\infty} S_N(\omega) = \sum_{l=-\infty}^{\infty} r_l \cdot e^{-j\omega l} =: S(\omega)$ the power spectral density (PSD).

Due to Parseval's theorem, $\sum_{n=-\infty}^{\infty} |x[n]|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(\omega)|^2 d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} S(\omega)\, d\omega$

→ The PSD can be interpreted as the power per frequency generated by the random process.

Remark: A zero-mean stationary Gaussian process is fully characterized by its PSD.

We can estimate the PSD (via the periodogram) via the squared DTFT spectrum of one realization of the r.p. (random prozess), or by first estimating the covariance sequence of the r.p. and then computing its DTFT.

The periodogram is a poor estimate of the PSD:

$$\lim_{N\to\infty} \operatorname{var}\{S_N(\omega)\} = \begin{cases} 2S^2(\omega) & \text{if } \omega \in \{0, \pi\} \\ S^2(\omega) & \text{otherwise} \end{cases}$$

Variance of the periodogram estimator, kind of the noise that you have in your estimate of the PSD is the same at every frequency ω and it does not decrease as increase the length of your estimations

Bartlett's method:

Segment $x[n]$ into N_1 consecutive segments of length N_2 with $N = N_1 \cdot N_2$ and then average the periodogram across all N_1 segments:

$$\hat{S}_B(\omega) := \frac{1}{N_1}\sum_{m=1}^{N_1} \frac{1}{N_2}\left(\sum_{n=1}^{N_1} x[n+m\cdot N_2]\cdot e^{-j\omega n}\right)^2$$

The averaging reduces the variance of the estimator by a factor $\frac{1}{N_1}$ (roughly), but it also reduces the frequency resolution.

An extension, called Welch's method, segments $x[n]$ into overlapping segments and windows each segment, e.g., Hann window.

Windowed periodogram:

Instead of averaging across segments, average across frequencies:

$$\hat{S}_w(w) = \sum_{k=-K}^{K} w_k[k] \cdot \hat{r}_k \cdot e^{-jwk}$$

with $K<N$ and $w_k[k]$ a window function, e.g., Hann window. The degree of smoothing in the frequency domain is controlled by K, with smaller K leading to more smoothing.

Lecture 11: Information Theory and compression

Consider a random variable $x \in \mathcal{X}$ with $|\mathcal{X}| = h$ and $P(x=i) = P_i \ (i \in \mathcal{X})$. We want to design a function $H(x)$, which we will call the entropy of x, that express the reduction in uncertainty (Ungewissheit) due to observing x.

What properties should $H(x)$ have?

1.) $H(x)$ should be maximal for $P_i = 1/k$. uniformly distributed (each event has an equal propability)

2.) Permutation invariance: Clanging the lables of events does not affect $H(x)$.

3.) Adding events with zero probability does not affect $H(x)$.

4.) Let $x \overset{\text{independant}}{\perp\!\!\!\perp} y$: $H(x,y) = H(x) + H(y)$ $\quad H(x) = -P(x) \cdot \log_2(P(x)) = -P(x) \cdot \log_{10}(P(x)) / \log_{10}(2)$

Let $x \overset{\text{dependant}}{\not\perp\!\!\!\perp} y$: $H(x,y) = H(x) + \sum_{i=1}^{k} P_i \, H(y|x_i)$

Theorem (Khinchin, 1957):

$H(x)$ satisfies properties 1.-4. iff (if and only if)

$$H(x) = -\sum_{i=1}^{k} P_i \cdot \log_2 P_i \quad [\text{bits for } \log_2]$$

Example:

Let $x \in \mathcal{X}$ with $|\mathcal{X}| = 2^3 = 8$ and $(P_1, \dots, P_8) = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64})$.

What is the entropy of x?

$$H(x) = -\tfrac{1}{2} \cdot \log_2(\tfrac{1}{2}) - \tfrac{1}{4} \cdot \log_2(\tfrac{1}{4}) - \tfrac{1}{8} \cdot \log_2(\tfrac{1}{8}) - \tfrac{1}{16} \cdot \log_2(\tfrac{1}{16}) - 4 \cdot \tfrac{1}{64} \cdot \log_2(\tfrac{1}{64}) = 2 \text{ bits.}$$

Can we compress x with 2 bits per symbol on average?

x	m(X)	m'(x)	|m'(x)|
1	000	0	1
2	001	10	2
3	010	110	3
4	011	1110	4
5	100	111100	6
6	101	111101	6
7	110	111110	6
8	111	111111	6

Average code length:

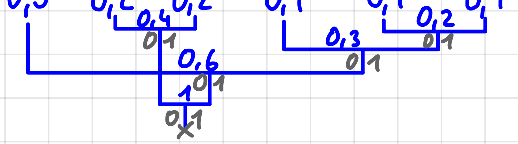
$$\tfrac{1}{2} \cdot 1 + \tfrac{1}{4} \cdot 2 + \tfrac{1}{8} \cdot 3 + \tfrac{1}{16} \cdot 4 + 4 \cdot \tfrac{1}{64} \cdot 6 = 2$$

$\underline{0}\,\underline{110}\,\underline{10} \ \hat{=}\ 132$

$1, 3, 1, 7, 5, 6 \dots \ \hat{=}_{m'} \ 0 \cdot 110 \ 0 \ 11111 \ 0$

Huffman coding: Construct a hirachical code by fusing low-probability symbols.

Code	00	10	11	010	0110	0111
Symbols	x_1	x_2	x_3	x_4	x_5	x_6
P(x)	0,3	0,2	0,2	0,1	0,1	0,1



String: $x_1\, x_5\, x_2\, x_3 \triangleq \underbrace{00}, \underbrace{0110}, \underbrace{10}, \underbrace{11} \triangleq x_1\, x_5\, x_2\, x_3$

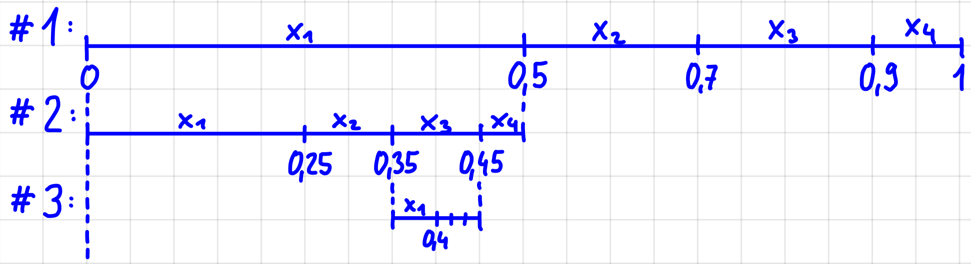
Remarks:

- We need to know P(x) for each symbol inadvance.
- If symbols we generated i.id. according P(x) and all probabilities are dyadic ($P(x) = 2^{-N}$ for some integer N for all x) then Huffman coding is optimal.

Arithmetic coding:

Represent a string of symbols by one single (real-valued) number between 0 and 1.

Symbol	P(x)
x_1	0,5
x_2	0,2
x_3	0,2
x_4	0,1



String: $\boxed{\begin{matrix} x_1 & x_3 & x_1 \\ 0 & 1 & 1 \end{matrix}}$ ≙ 0,375 → [0,35 0,4]

Remarks:

- AC requires specification of the string length.
- AC requires probabilities of each symbol.
- AC is optimal for iid symbol strings.

Lempel-Ziv-Welch (LZW)-Coding

Build a codebook on the fly by creating new words for novel sub-strings.

Symbol	Code word
x_1	0000
x_2	0001
x_3	0010
x_4	0011
x_1x_1	0100
x_1x_2	0101
x_2x_1	0110
$x_1x_2x_3$	0111
x_3x_1	1000

String: $x_1x_1x_2x_1x_2x_3x_1x_2x_3$ | 18 bits

Coded String: $\underset{x_1}{00}\ \underset{x_1}{000}\ \underset{x_2}{001}\ \underset{x_1x_2}{101}\ \underset{x_3}{010}\ \underset{x_1x_2x_3}{0111}$ | 18 bits

Remarks:

- LZW coding is asymptotically optimal.
- Decoding reconstructs the code book on the fly.
- LZW is the basis for ZIP, GIF, TIFF and is used in JPG.