Digital Signal Processing

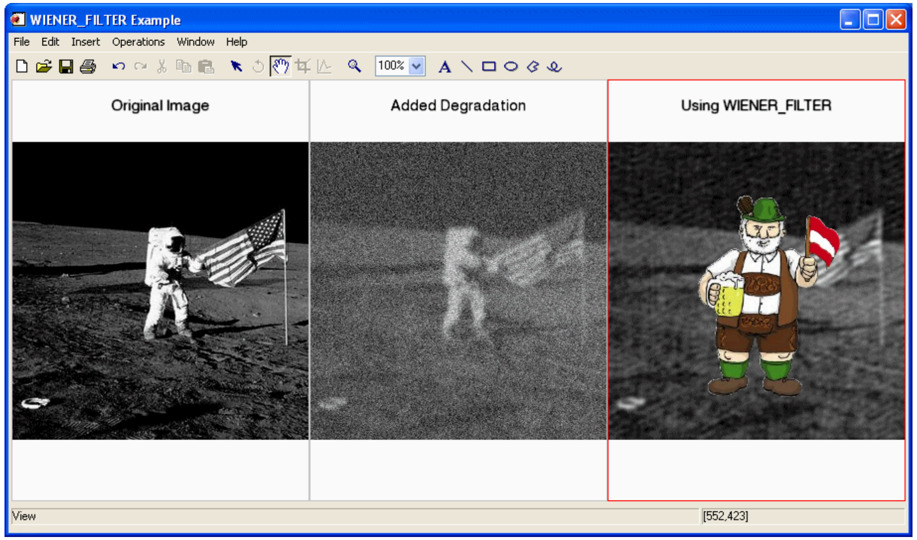
Zusammenfassung

Joel von Rotz / Quelldateien

Table of Contents

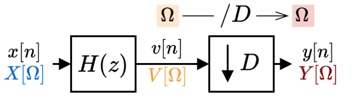
Achtung, Achtung!

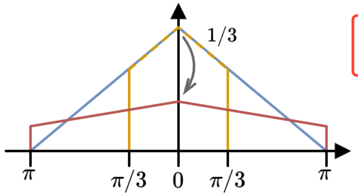
Anstatt über die Fehler in der Zusammenfassung zu meckern, wäre ein *Pull Request* sehr töfte! Github Repo Link



Sampling Rate Conversion

Decimation
Reducing sampling rate by an **Integer Factor** D





Sampling Theorem $f_{\max} < \frac{f_s}{2}$
Don't forget!

$$H(z) = \begin{cases} 1 \text{ if } \Omega \in [-\pi/D, \pi/D] \\ 0 \text{ otherwise} \end{cases}$$

For $\Omega \in [-\pi, \pi]$

Decimated Frequency: $\boxed{F_Y = F_X/D \leftrightarrow \Omega_Y = \Omega_{X,V}/D}$

Ideally filtered

$$Y(z) = \frac{1}{D}\sum_{d=0}^{D-1} V\left(\frac{\Omega}{D} - 2\pi\cdot\frac{d}{D}\right)$$

General Formula

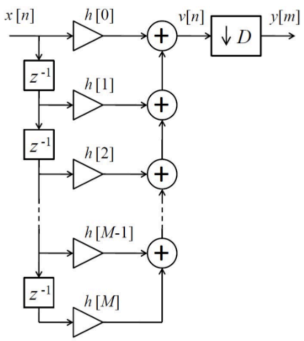
$$Y(z) = \frac{1}{D}\sum_{d=0}^{D-1} V\left(\frac{\Omega}{D} - 2\pi\cdot\frac{d}{D}\right)$$

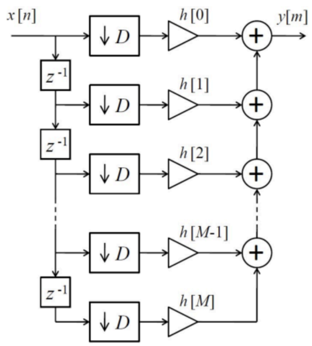
Direct Implementation

FIR Filter of order M produces full signal $v[n]$ + downsampler discards $D-1$ samples afterwards → **inefficient!**

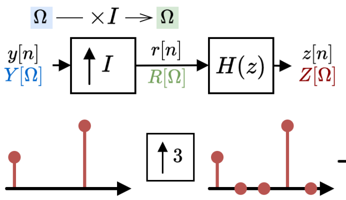
Efficient Implementation

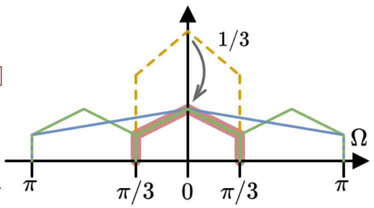
Downsampling beforehand allows the multiplier to operate at the reduced sampling rate → **much better!**





Interpolation
Increase sampling rate by an **Integer Factor** I





$$r[n] = \begin{cases} y[n/I] \text{ if } n \in [0, \pm I, \pm 2I, \dots] \\ 0 \text{ otherwise} \end{cases}$$

Interpolated Frequency: $\boxed{F_{Z,R} = I\cdot F_Y \leftrightarrow \Omega_{Z,R} = I\cdot\Omega_Y}$

Interpolation Formula

$$R(\Omega) = \sum_{n=-\infty}^{\infty} r[n]e^{-j\Omega\cdot n} = \sum_{n=-\infty}^{\infty} y[m]e^{-j\Omega\cdot I\cdot m} = Y(I\Omega)$$

Low Pass Filter

For $\Omega \in [-\pi, \pi]$

$$H(z) = \begin{cases} I \text{ if } \Omega \in [-\pi/I, \pi/I] \\ 0 \text{ otherwise} \end{cases}$$

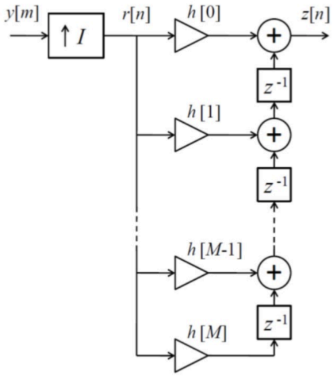
Lowpass-filter uses Ω and **NOT** Ω !

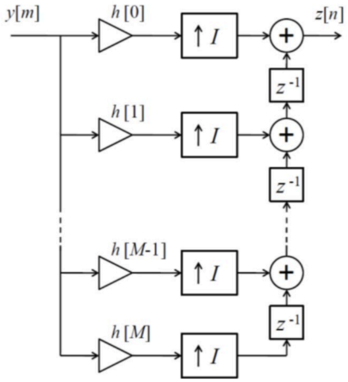
Direct Implementation

FIR or IIR Filter ; $I-1$ out of I $r[n]$ samples are zero → **inefficient!**

Efficient Implementation

Upsampling after filtering → multiplier operates at **reduced** sampling rate (F_Y) → **much better!**



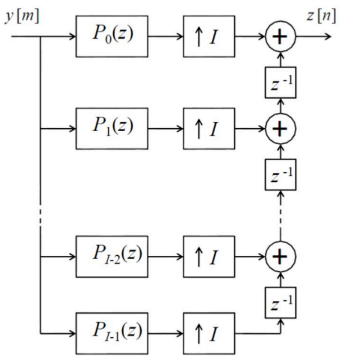


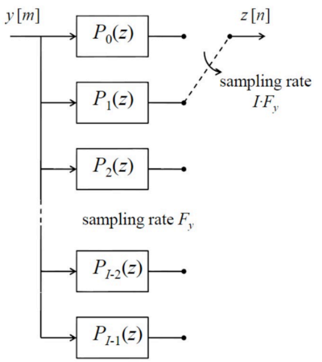
Polyphase Filter Structure
Efficient filter implementation

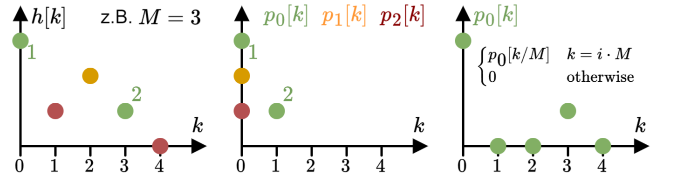
Split filter into M **downsampled** variants of the impulse resonse $h[k]$. Every variant $p_{i[k]}$ holds only every M-th coefficient („sum" of variants = $h[k]$)

$$p_i[k] = h[kM+i], \quad i = 0,1,\dots,M-1 \quad | \quad H(z) = \sum_{i=0}^{M-1} z^{-1}P_i(z^M)$$

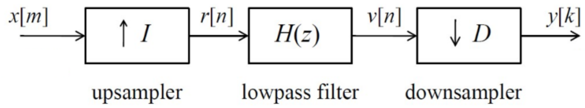
$$P_{i(z)} = \sum_{k=-\infty}^{\infty} p_i[k]z^{-k} = \sum_{k=-\infty}^{\infty} h[kM+i]z^{-k}, \quad i = 0,1,\dots,M-1$$







Sampling Rate Conversion



$$F_X \Rightarrow F_R = I\cdot F_X \Rightarrow F_V = F_R \Rightarrow F_Y = I/D\cdot F_X = 1/D\cdot F_V$$

Decimation ⇒ loss of information

Interpolation ⇒ higher intermediate sampling rate

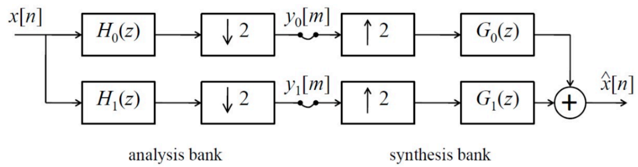
$$F_H(\Omega) = \min\left(\frac{\pi}{I}, \frac{\pi}{D}\right)$$

$$\frac{I}{D} = \underbrace{I\cdot\left(\frac{1}{D}\right)}_{H_I\to I\to D\to H_D} = \underbrace{\left(\frac{1}{D}\right)\cdot I}_{I\to H\to D}$$

Filter Banks

Quadrature Mirror Filters

QMF: compensate loss of information caused by decimation



$$Y_i(\Omega) = \frac{1}{2}\left(H_i\left(\frac{\Omega}{2}\right)\cdot X\left(\frac{\Omega}{2}\right) + H_i\left(\frac{\Omega}{2}-\pi\right)\cdot X\left(\frac{\Omega}{2}-\pi\right)\right)$$

$H_0(z), G_0(z)$: lowpass filter ; $H_1(z), G_1(z)$: highpass filter

$$\hat{X}(\Omega) = \frac{1}{2}(H_0(\Omega)\cdot G_0(\Omega) + H_1(\Omega)\cdot G_1(\Omega))\cdot X(\Omega)$$

$$+\underbrace{\frac{1}{2}(H_0(\Omega-\pi)\cdot G_0(\Omega) + H_1(\Omega-\pi)\cdot G_1(\Omega))\cdot X(\Omega-\pi)}_{\text{alias term}}$$

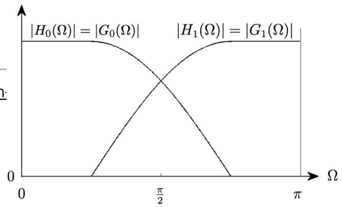
Condition for $\hat{x}[n]$ free of aliasing:

$$H_0(\Omega-\pi)\cdot G_0(\Omega) + H_1(\Omega-\pi)\cdot G_1(\Omega) = 0$$

For getting rid of aliasing and manage perfect reconstruction:

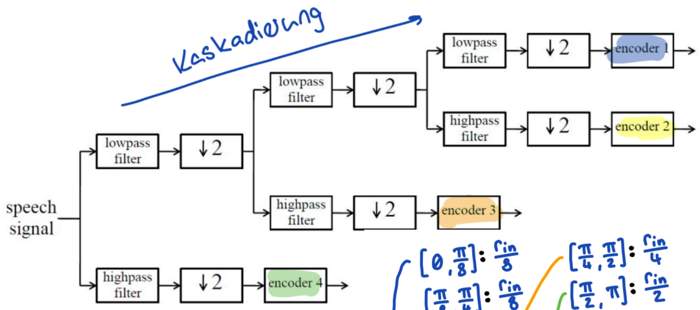
$$H_0(\Omega) = H(\Omega) \quad H_1(\Omega) = H(\Omega-\pi)$$

$$G_0(\Omega) = H(\Omega) \quad G_1(\Omega) = -H(\Omega-\pi)$$

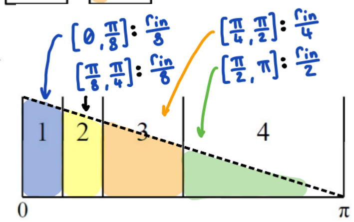


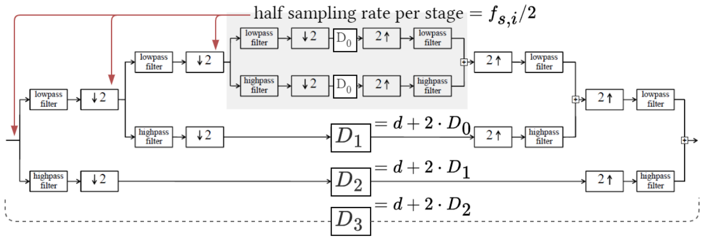
ⓘ Perfect Reconstruction

Synthesized signal $\hat{x}[n]$ is identical to input signal $x[n]$ except for **arbitrary delay** and **scaling** by a constant factor.



R_{in}: Abtastrate





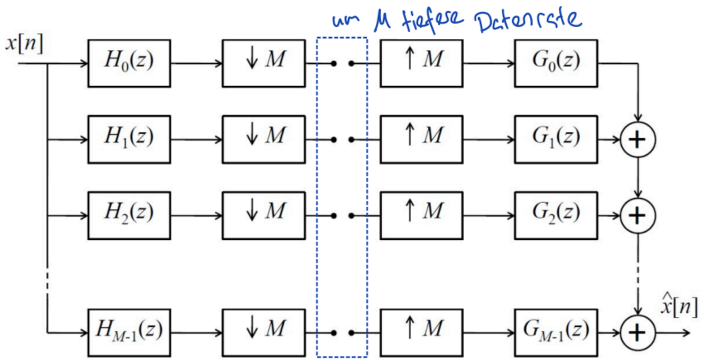
Example with $d = 5$ sample delays

$$D_1 = 5 \quad ; \quad D_2 = d + 2\cdot D_1 = 15 \quad ; \quad D_3 = d + 2 \cdot D_2 = 35$$

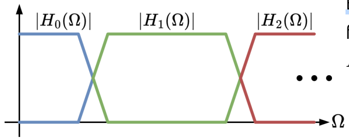
- D_3 is viewed from outside, therefore the sample rate is halved leading to 35 spl delay

DFT Filter Banks

- QMFB (Quadrature-M-Filter-Banks) → downsamples the sampling frequency by M
  - *critical sampling* → channels = downsampling factor D (over: $C > D$; under: $C < D$)
  - slightly imperfect reconstruction with critical sampling → oversampling (longer)

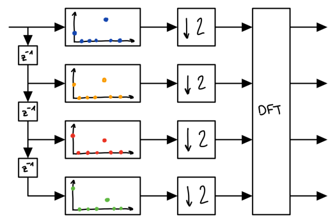


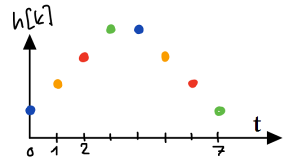
$$H_\ell(z) = H\left(z \cdot e^{-j2\pi\ell/M}\right) \; ; \; H_\ell(\Omega) = H(\Omega - 2\pi\ell/M) \; ; \; l = 0, 1, ..., M-1$$



Example

following describes a filter bank with $M = 4$ chls and oversmpl'ing factor $D = 2$





⇐ input: $h[0], h[1], ..., h[M-1]$

ⓘ Some Basics

deterministic: value can be determined at any time **transient**: limited time duration

Finite Impulse Response (always stable)
- Impulse response is infinite
- No feedback ($y[n-\cancel{m}]$)

$$y[n] = \sum_{a=0}^{N-1} b_a \cdot x[n-a]$$

Infinite Impulse Response (un-/stable)
- Impulse response infinite due to feedback
- Feedback of signals ($y[n-m]$)

$$y[n] = \sum_{n=0}^{N-1} b_n \cdot x[k-n] - \sum_{m=1}^{M} a_m \cdot y[n-m]$$

Levinson-Durbin-Rekursion
- efficient algo for solving linear equation systems with the *Toeplitz*-Matrix
- instead of inverting matrices, the TM is used to iteratively determine the solution.

(linear) convolution

$$z[n] = \sum_{i=-\infty}^{\infty} x[i]y[n-i] = x[n] * y[n]$$

Linear Correlation

$$r_{xy}[n] = \frac{1}{N}\sum_{i=0}^{N-1} x[i]y[i+n]$$

$$N_{xy} = N_x + N_y - 1 \quad r_{xy}[n] \neq r_{yx}[n]$$

Random Signals

- Characterization
  - first-order statistic: **Mean value**
  - second-order statistic: **Autocorrelation**
- Stationary Signals
  - no change in mean value ($\dot{m}_x = \dot{\gamma}_{xx} = 0$)
  - infinite energy → no fourier transform $\underbrace{(X_{[n1,n2]}(\Omega))}_{\text{ranged DTFT}}$

(γ_{xx} higher → more determinable/similar ; more fluctuating → narrower the γ_{xx})

Autocorrelation γ and Spectrum

Mean value

$$m_x = E\{x[n]\}$$

$$\dot{m}_x = 0 \text{ for stationary signals}$$

Autocorrelation

$$\gamma_{xx}[m] = E\{x^*[n] \cdot x[n+m]\}$$

$$\gamma_{xx}[m] = \gamma^*_{xx}[-m] \text{ for real random}$$

$$\dot{\gamma}_{xx} = 0 \text{ for stationary signals}$$

m: distance between two samples

Autocovariance (for signals with $m_x \neq 0$)

$$c_{xx}[m] = E\{(x[n] - m_x)^* \cdot (x[n+m] - m_x)\}$$

$c_{xx}[0]$: Variance of signal ($E\{|x[n] - m + x|^2\}$)

Power: $P = \gamma_{xx}[0]$

ⓘ stationary time ≠ stationary frequency

A stationary random signal has fluctuating f

Power Density Spectrum Power cannot be negative! Squared Amplitude Spectrum

- Reveals Spectral composition of a stationary process („where energy is")
- Mirrored on Y-axis (range $\Omega \in [0, \pi]$ suffices)
- $\Gamma_{xx} = \mathcal{F}(\gamma_{xx})$

$$\Gamma_{xx}(\Omega) = \sum_{m=-\infty}^{\infty} \gamma_{xx}[m] \cdot e^{-j\Omega m} \quad ; \quad P_x = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Gamma_{xx}(\Omega) d\Omega$$

White Noise

White noise has a **constant** spectrum, as they represent all noises at the same time!

$$\gamma_{ww}[m] = \begin{cases} \sigma^2 \text{ if } m = 0 \\ 0 \text{ if } m \neq 0 \end{cases} ; \; \Gamma_{ww}(\Omega) = \sigma^2_w$$

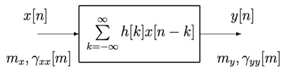
Wiener-Khinchin-Theorem

Power Spectrum Density corresponds to the DTFT of the autocorrelation sequence (of stat. rand. sig)

$$\Gamma_{xx}(\Omega) = \sum_{m=-\infty}^{\infty} \gamma_{xx}[m] \cdot e^{-j\Omega m}$$

Spectral Shaping

Stationary random in, stationary random out



$$y[n] = \sum_{k=-\infty}^{\infty} h[k] \cdot x[n-k] \quad H(\Omega) = \sum_{k=-\infty}^{\infty} h[k] \cdot e^{-j\Omega k}$$

$$m_y = H(0) \cdot m_x \quad | \quad \gamma_{yy} = h^*[-i] * \gamma_{xx}[m] * h[k]$$

for auto correlation one $h[n]$ is mirrored and complex conjugated! $i = -\infty, ..., -1, 0, 1, ..., \infty$

$$\Gamma_{yy}(\Omega) = H^*(\Omega) \cdot \Gamma_{xx}(\Omega) \cdot H(\Omega) = |H(\Omega)|^2 \cdot \Gamma_{xx}(\Omega) = \underbrace{|H(\Omega)|^2 \cdot \sigma^2_w}_{\text{white noise}}$$

Linear Models for Stochastic Processes

Include noise in simulations!

$$\Gamma_{ww}(z) = \sum_{m=-\infty}^{\infty} \gamma_{ww}[m] \cdot z^{-m} = \sigma^2_w$$

$$\Gamma_{yy} = H(z^{-1}) \cdot \Gamma_{xx}(z) \cdot H(z) = \sigma^2_w(z) \cdot H(z) \cdot H(z^{-1})$$

↓ Stable when all poles and zeroes inside unit circle ↓

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum b_k z^{-k}}{1 + \sum a_k z^{-k}} \Longrightarrow \Gamma_{yy}(z) = \sigma^2_w \cdot \frac{B(z) \cdot B(z^{-1})}{A(z) \cdot A(z^{-1})}$$

white no. provides complete random portion

Noise Whitening: reverse operation $\boxed{H_w(\Omega) = 1/(H(\Omega))}$ reverses the generated random noise back to white noise. $w[n]$: „innovations process" of $y[n]$.

$$w[n] = \frac{1}{b_0} \cdot \left(y[n] + \sum_{k=1}^{N} a_k y[n-k] - \sum_{k=1}^{M} b_k w[n-k]\right)$$

example: Pre-Filter for making calculation of wiener-filters easier.

Moving Average (MA) model Wideband applications

- White noise + FIR-Filter $H(z)$ with Mth order (M delays + no poles)

$$y[n] = \sum_{k=0}^{M} b_k w[n-k]$$

$H(z)$: FIR filter

$1/H(z)$: IIR Filter

- $b_0 = b_1 = ... = (M+1)^{-1}$: every output sample = avg, over sliding window $M+1$
- different coefficients result in selective combinations of white noise input
- *World Representation*: every stat. stoc. process ⇒ infinite moving average process

$$\gamma_{yy}[m] = \begin{cases} \sigma^2_w \sum_{k=m}^{M} b_k \cdot b^*_{k-m} & \text{if } 0 \leq m \leq M \\ 0 & \text{if } m > M \\ \gamma^*_{yy}[-m] & \text{if } m < 0 \end{cases}$$

Autoregressive (AR) model Narrowband applications (Human Vocal Tract: all pole filter)

- $B(...) = 1$: no zeroes, only poles

$$y[n] = w[n] - \sum_{k=1}^{N} a_k \cdot y[n-k]$$

$$\gamma_{yy}[m] = \begin{cases} -\sum_{k=1}^{N} a_k \cdot \gamma_{yy}[m-k] & \text{if } m > 0 \\ \sigma^2_m - \sum_{k=1}^{N} a_k \cdot \gamma_{yy}[-k] & \text{if } m = 0 \\ \gamma^*_{yy}[-m] & \text{if } m < 0 \end{cases}$$

- weighted sum older values + noise ⇒ staionary not guaranteed

Yule-Walker equations:

$$\begin{pmatrix} \gamma_{yy}[0] & \gamma_{yy}[-1] & \cdots & \gamma_{yy}[-N] \\ \gamma_{yy}[1] & \gamma_{yy}[0] & \cdots & \gamma_{yy}[-N+1] \\ \vdots & \vdots & & \vdots \\ \gamma_{yy}[N] & \gamma_{yy}[N-1] & \cdots & \gamma_{yy}[0] \end{pmatrix} \cdot \begin{pmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{pmatrix} = \begin{pmatrix} \sigma^2_w \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

ARMA model

- Generalization of MA & AR
- $B(...) : M \geq 1$ and $A(...) : N \geq 1$
- More suitable for random processes due to having fewer coefficients for „same" accuracy.

Spectral Density Estimation

Nonparametric Method

Big no-no: periodogram

DTFT² = *periodogram*

$$\hat{\Gamma}_{xx}(\Omega) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x[n] \cdot e^{-j\Omega n}\right|^2$$

⊗: any $\Omega_0 \to \hat{\Gamma}(\Omega_0)$ has large variance

⊗: variance does not decrease with increasing sample basis

Biased Autocorrelation Estimator / **Unbiased Autocorrelation Estimator**

for m between 0 and $N-1$ / $= 0, 1, ..., N-1$

$$\hat{\gamma}_{xx}[m] = \frac{1}{N}\sum_{n=0}^{N-m-1} x^*[n] \cdot x[n+m] \qquad \hat{\gamma}_{xx}[m] = \frac{1}{N-m}\sum_{n=0}^{N-m-1} x^*[n] \cdot x[n+m]$$

Biased:
+ Easy to calculate
— Distorts result, as higher m leads to less data in the sum
— same problem as periodogram

Unbiased:
+ No distortion → corrected by $1/(Nm)$
— a bit complexer to calculate
— High variance at $|m| \approx N \to$ few summands
— does not guarantee $\hat{\Gamma}_{xx}(\Omega) \geq 0$ (big nono)

Possible estimator corrections

- Bartlett's method: segmentation and averaging
- Windowing: smooth spectral density → deceases variance at cost of resolution

Parametric Method

- Structure of a noise source is known (to some degree) → build on specific model tuned by a fixed number of parameters
  - ARMA models are populars, due few parameters with tight fitting to real sources

+ High accuracy
— Complexer to calculate
— improper parameters leads to instability

Yule-Walker (using AR model)

Substituting $\gamma_{xx}[m]$ in the equations with $\hat{\gamma}_{xx}[m]$
— unbiased variant can lead to unstable AR

Burg's Method

Uses forward and backwards prediction similar to Levinson-Durbin recursion.
+ Better results, even with fewer samples
+ More efficient, as forward and backwards error is minimized

Least Square model fitting (with AR)

Fitting via square error. $\hat{y}$ = interpreted as forward prediction from the preceding N samples

$$\hat{y} = -\sum_{k=1}^{N} a_k \cdot y[n-k]$$

prediction error: $w[n] = y[n] - \hat{y}[n]$

- might lead to same values as Yule-Walker

Use Case of AR model: nuclear reactors. AR model fitted to reactor noise during normal operation. Deviating noise leads to sudden increase in prediction error $|y[n] - \hat{y}[n]|$.

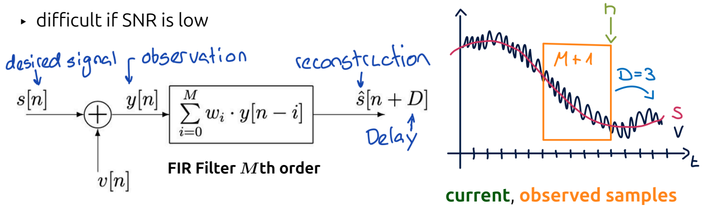
Optimal Linear Filters

Perfect reconstruction might not be possible in presence of noise → approximate original signal as accurately as possible. *Estimation* is interpreted in various different ways, but *mean squared error* is generally used (also used for power/energy calc.).

+ Simple structure
+ Linear filters are preferred: (almost) as powerful as complex nonlinear estimators.
— MSE: strong deviations are heavily weighted by squaring

Wiener Filters

- For discrete & continuous stationary signals/noise
- Efficient implementation using Levinson-Durbin recursion
- $s[n]$ & $v[n]$ are independent, zero-mean stationary random signals
- Calculate filter coefficients $w_0, w_1, ..., w_M$ to keep MSE low
  - difficult if SNR is low



Estimated signal $\hat{s}$ with D delays:

$$\hat{s}[n+D] = \sum_{i=0}^{M} \omega_i \cdot y[n-i]$$

Smoothing $D < 0$: eliminate noise
filtering $D = 0$: (almost) recover signal in real time
prediction $D > 0$: forecast the future course

$$\varepsilon_{\text{MSE}} = E\{|\underbrace{\hat{s}[n+D]}_{\text{prediction}} - \underbrace{s[n+D]}_{\text{original}}|^2\}$$

mean ↑

optimal coefficients:

$$\underline{\tilde{w}} = \arg\min_{\underline{w}}(\varepsilon_{\text{MSE}}(\underline{w})) \Rightarrow \text{derivative} = 0$$

Wiener Hopf Equation

$$R_{yy} = \begin{pmatrix} \gamma_{yy}[0] & \gamma_{yy}[-1] & \cdots & \gamma_{yy}[-M] \\ \gamma_{yy}[1] & \gamma_{yy}[0] & \cdots & \gamma_{yy}[1-M] \\ \vdots & \vdots & & \vdots \\ \gamma_{yy}[M] & \gamma_{yy}[M-1] & \cdots & \gamma_{yy}[0] \end{pmatrix} \quad r_{sy} = \begin{pmatrix} \gamma_{sy}[D] \\ \gamma_{sy}[D+1] \\ \vdots \\ \gamma_{sy}[D+M] \end{pmatrix} \quad \tilde{w} = \begin{pmatrix} \tilde{w}_0 \\ \tilde{w}_1 \\ \vdots \\ \tilde{w}_M \end{pmatrix}$$

Autocorrelation of filter input: $\gamma_{yy}[m] = E\{y[n] \cdot y^*[n-m]\}$

Autocorrelation of white noise: $\gamma_{vv}[m] = \delta[m] = \begin{cases} 1 \text{ if } m = 0 \\ 0 \text{ if } m \neq 0 \end{cases}$

Crosscorrelation of filter input and desired response: $\gamma_{sy}[m] = E\{s[n] \cdot y^*[n-m]\}$

$$R_{yy} \cdot \tilde{w} = r_{sy} \quad \Longrightarrow \quad \tilde{w} = R_{yy}^{-1} \cdot r_{sy}$$

Special Case undistorted signal of interest appears superimposed by additive noise:

$$\gamma_{yy}[m] = \gamma_{ss}[m] + \gamma_{vv}[m] \qquad \gamma_{sy}[m] = \gamma_{ss}[m]$$

Unconstrained Wiener Filters

Neglecting the constraints imposing causality and finite length. Obtain optimal, non-causal IIR filter with the impulse response ..., $\tilde{w}_{-1}, \tilde{w}_0, \tilde{w}_1$, ... setting $D = 0$.

$$\sum_{i=0}^{M} \tilde{w}_i \cdot \gamma_{yy}[m-i] = \gamma_{sy}[m+D]$$

$$\widetilde{W}(z) \cdot \Gamma_{yy}(z) = \Gamma_{sy}(z)$$

with $\Gamma_{yy}(z) = \Gamma_{ss}(z) + \Gamma_{vv}(z)$ and $\Gamma_{sy}(z) = \Gamma_{ss}(z)$:

$$\widetilde{W}(z) = \frac{\Gamma_{ss}(z)}{\Gamma_{ss}(z) + \Gamma_{vv}(z)}$$

$$\widetilde{W}(\Omega) = \frac{\Gamma_{ss}(\Omega)}{\Gamma_{ss}(\Omega) + \Gamma_{vv}(\Omega)}$$

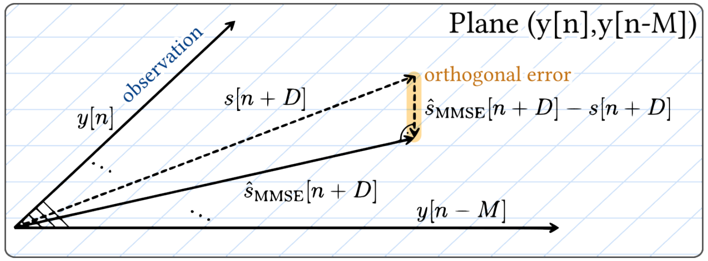
- At frequencies Ω with very low noise power density $\widetilde{W}(\Omega) \approx 1$
  ↳ ... with very high noise power density $\widetilde{W}(\Omega) \approx 0$
- The higher the noise power density, the greater the attenuation imposed on the spectrum input

⚠ Filter need to be causal

↳ Method of omitting anti-causal values is **suboptimal**
  ↳ **Special case**: white noise → apply *noise whitening* before filter

The Principle of Orthogonality

- necessary condition for optimality
- estimation error is orthogonal (thus uncorrelated) to the filter input
  - error does not contribute to the estimation
  - not meeting the condition leads to suboptimal filter



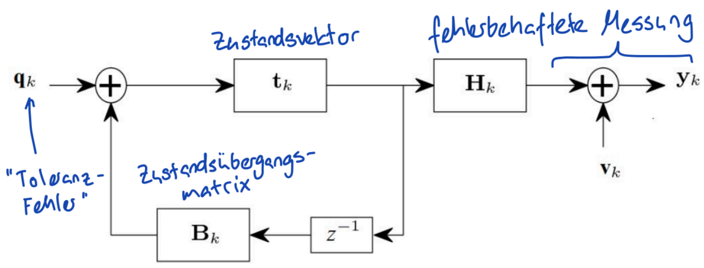
$E\{\hat{s}_{\text{MMSE}}[n+D] - s[n+D]\} = 0$ to achieve minimum mean square error!

Kalman Filter

for discrete systems ; for continuous it's called *Kalman Bucy*-filter

- „Improvement" over Wiener Filter
- Can be employed in dynamic systems → not stationary signals needed
  - due to generic character/nature can be used for various applications (such as coordinate tracking of aircraft or spacecraft)
- Wiener filter has finite order ⇒ Kalman algorithm has a recursive nature and thus represents an **infinite-length filter**
- Iterative error reduction (error starts high, reduced over time)

Prediction ⇆ Correction



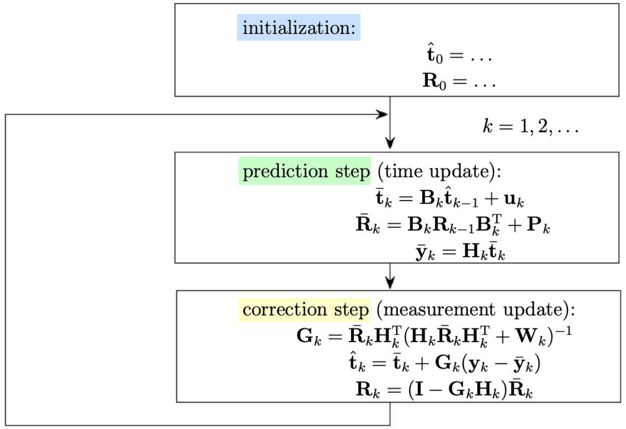
Measurement equation / Process equation

t_k contains all information of the system state at time k

$$y_k = H_k \cdot t_k + v_k \qquad t_k = B_k \cdot t_{k-1} + q_k$$

entity	symbol	deterministic/random
state vector	$\mathbf{t}_k$	random
covariance matrix of $\mathbf{t}_k$	$\mathbf{R}_k$	deterministic
observation	$\mathbf{y}_k$	random
measurement matrix	$\mathbf{H}_k$	deterministic
measurement error	$\mathbf{v}_k$	random with mean $\mathbf{0}$, covariance $\mathbf{W}_k$
state transition matrix	$\mathbf{B}_k$	deterministic
input	$\mathbf{q}_k$	random with mean $\mathbf{u}_k$, covariance $\mathbf{P}_k$

+ No limit on filter order (it's recursive, therefore infinite)
+ Dynamic adaptation in time-variant systems
+ Supports systems with stochastic and determinable components



$\hat{x}$: real estimation ; $\bar{x}$: predictive estimation

Adaptive Filters

- Wiener & Kalman filters are optimal filters under the assumption, the statistics of the processes are known → Rarely met in practice :(
- Filters which adapt to unknown and possibly varying environmental conditions are known as **adaptive filters**
  - *Example application*: wireless radio receivers to compensate for signal distortions in the channel
- **Optimal Linear Filters**: estimate the necessary autocorrelation and crosscorrelation sequences to derive filter coefficients
- **Filters with minimization of a cost function**: typically LMS deviation of the filter output from the desired response

Linear Predictive Coding

For speech coding

- estimate the necessary autocorrelation and crosscorrelation sequences to derive filter coefficients
- **Vocoders** → built on certain speech synthesis model and extract the model parameters
  - transmitting only necessary parameters requires less bandwidth than transmitting the sampled waveform directly!
  - *Code Excited Linear Prediction* (CELP) compresses speech into a 13kbit/s signal.
  - *LPC-10e* compresses with 2400bit/s at the cost of reduced audio quality
- **Waveform coders** → aim to preserve the signal waveform
  - usually requires high bandwidth

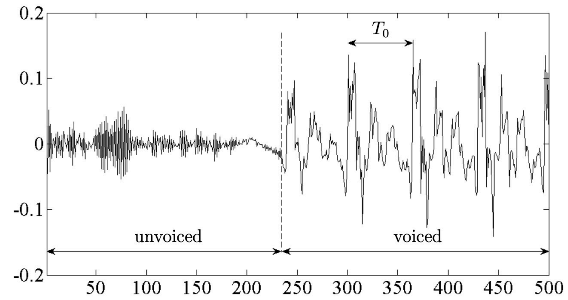
Example: Human Vocal Tract

Approximation of the human vocal tract as an all-pole filter:

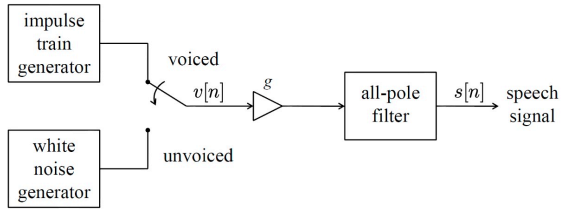
$$H(z) = \frac{g}{1 - \sum_{k=1}^{P} a_k \cdot z^{-k}}$$

P: filter order g: gain
a_k coefficients

The vocoder determines following parameters: filter coefficients $a_1, \dots, a_{10}$, gain g, voiced or unvoiced $v[n]$ and the voiced period T_0.



- **unvoiced** (noise-like): consonants such as „f"
- **voiced** (show periodical signals): vocals and some consonants
- lower k coefficients $(a_1, \dots, a_4)$ carry more information, than „loud" coefficients (a_9, a_{10})



LMS Algorithm

Least mean square – tries to minimize its cost function (sample-by-sample)

+ Less complex (steps until sufficent optimum) than Recursive Least Square (RLS)
— Slow convergence than RLS
— More issues with non-stationary signals than RLS

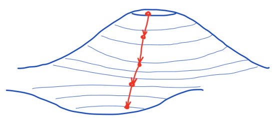
1. Calculate gradient (iterative process) usingmean-squared error ε_{MSE}
   - Large step size μ overshoots the target and oscillates around it
   - Small μ leads to slow **convergence speed** ($\propto$ $^{-1}$ iterations)

$$\varepsilon_{\text{MSE}}(\underline{w}) = E\Big\{\big(\underline{w}^T \cdot \underline{y}_n - s[n]\big)^2\Big\} \Longrightarrow \nabla\varepsilon_{\text{MSE}}(\underline{w}) = E\Big\{2 \cdot \big(\underline{w}^T \cdot \underline{y}_n - s[n]\big) \cdot \underline{y}_n\Big\}$$

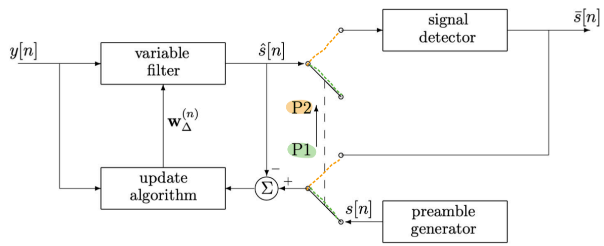
2. Update values for next steps

(n) indicates the iteration count + $\underline{w}^{(0)} = 0_{M+1}$

$$\underline{w}^{(n+1)} = \underline{w}^{(n)} + \mu \cdot \underbrace{\Big(s[n] - \big(\underline{w}^{(n)}\big)^T \cdot \underline{y}_n\Big) \cdot \underline{y}_n}_{=\underline{w}_\Delta^{(n)}}$$

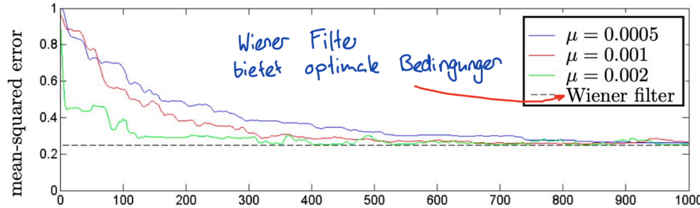


(-1) due to going against the gradient (go to **local** minima)



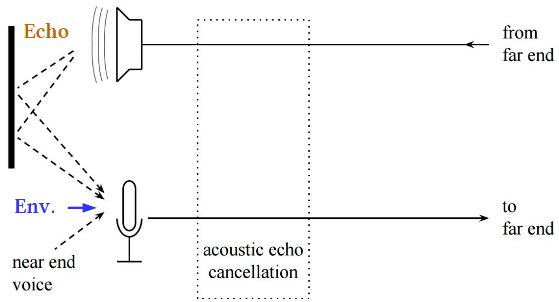
P1 Training: data is known → for calibration and distortion

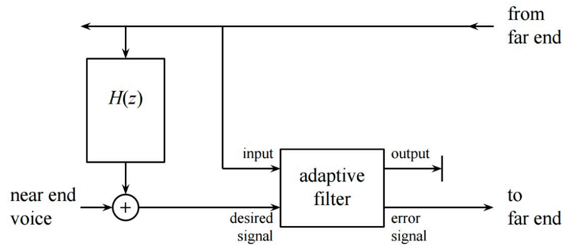
P2 Decision Directed Operation: Filter is checked and adjusted



Acoustic Echo Cancellation (AEC)

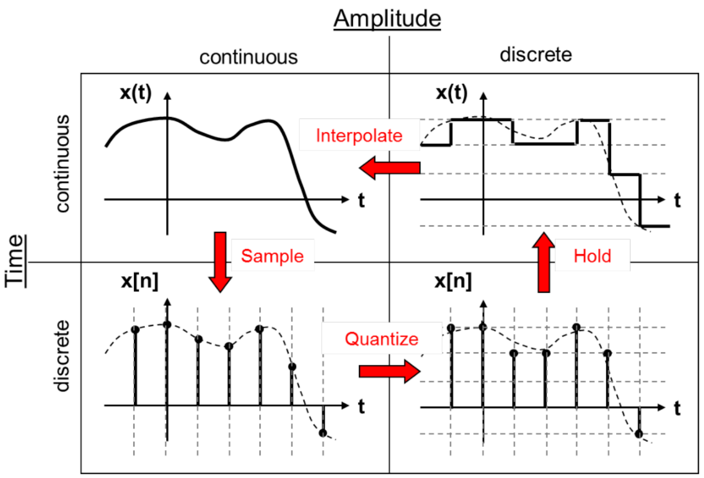
- AEC's task is to remove far end components from signal delivered by microphone.
- Filter adjusts itself to transmitted time variant sample
- reproduce echo & subtract from microphone-sample
- mode detection: speaker A, speaker B, both speakers, none
  - None: switch with ambient noise to protect the adaptive filter from its own degradation (when exposed to ambient noise)





Digital Signal Processing (DSP)

DSP concerned in the application of the following methods: ① **Signal Generation**, ② **Signal Analysis**, ③ **Signal Composition**, ④ **Signal Selection**



Pros (3 P's): **P**rogrammability, **P**arametrizability, Re-**P**eatability

Cons: additional effort for ADC & DAC, No processing of broadband HF, electro-magnetic disturbance

Signal Analysis

Sampling an Analog Signal

$$f_S = \frac{1}{T_S} \quad x(n \cdot T_S) = x[n]$$

Other Functions

causal: $x[n] = 0$ for $n < 0$

T_S: Always known!

unit impulse

$$\delta[n] = \begin{cases} 0 : n \neq 0 \\ 1 : n = 0 \end{cases}$$

step impulse

$$u[n] = \begin{cases} 0 : n < 0 \\ 1 : n \geq 0 \end{cases}$$

periodic symbols ($k = T_0/T_S$)

$$x[n] = x[n + T_0/T_S]$$

$$= \hat{X} \cdot e^{j2\pi \cdot f_0 \cdot n \cdot T_S}$$

$$= \hat{X}\left(C(\_) + j \cdot S(\_)\right)$$

expected/mean value

$$\mu_x = \frac{1}{N}\sum_{i=0}^{N-1} x[i]$$

(mean value)^2 / avg DC power

$$\rho^2 = \frac{1}{N}\sum_{i=0}^{N-1} x[i]^2$$

variance/avg AC power

$$\sigma_x^2 = \frac{1}{N}\sum_{i=0}^{N-1} (x[i] - \mu_x)^2$$

Acoustic signals: corresponds to audible power content

power ratio

$$A_{dB} = 10 \cdot \log_{10}\left(\frac{P_1}{P_2}\right)$$

Signal-to-Noise-ratio

$$\text{SNR} = 10 \cdot \log_{10}\left(\frac{P_S}{P_N}\right)$$

$$= 20 \cdot \log_{10}\left(\frac{U_S}{U_N}\right)$$

power ratio

$$A_{dB} = 10 \cdot \log_{10}\left(\frac{P_1}{P_2}\right)$$

static correlation ($x = y \Rightarrow \uparrow R$)

$\Rightarrow$ yields new signal, quantifying the similarty of x and y

$$R = \frac{1}{N}\sum_{i=0}^{N-1} x[i]y[i]$$

linear correlation

$$r_{xy}[n] = \frac{1}{N}\sum_{i=0}^{N-1} x[i]y[i+n]$$

$$N_{xy} = N_x + N_y - 1$$

$$r_{xy}[n] \neq r_{yx}[n]$$

(linear) convolution

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

$$= x[n] * y[n]$$

circular convolution

$$N_X = N_Y$$

$$z[n] = x[n] \circledast_N y[n]$$

Example:

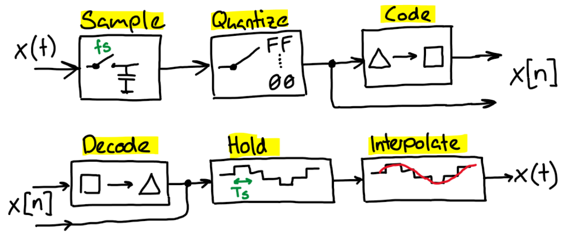
$$x = \underbrace{\{0.5, 1\}}_{=a+0.5} \quad \& \quad y = \underbrace{\{1, -1\}}_{=-b+1}$$

$$\begin{pmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0.5 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0.5 \\ -1 \end{pmatrix}$$

$$z[n] = \{0.5, 0.5, -1\}$$

$$x \cdot y\,|_{b=a} = -a^2 + 0.5a + 0.5$$

A/D & D/A Conversion



Code/Decode z.B. DFT & IDFT ; *Interpolate* z.B. Tiefpass-Filter

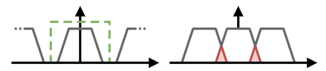
Sampling & Aliasing

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

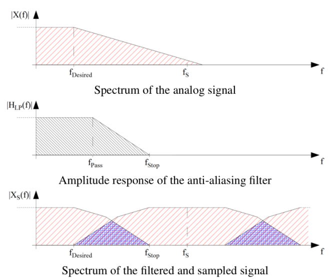
Sampling Theorem! $\left[f_{\max} < \frac{f_S}{2}\right]$

⚠ Aliasing

Bei $f_{\max} > f_S/2$ ensteht Aliasing.

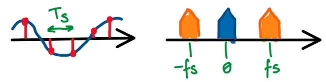


Wenn Theorem nicht möglich ist.



$$f_{\text{Pass}} \geq f_{\text{Desired}} \quad f_{\text{Stop}} \geq f_S - f_{\text{Desired}}$$

⚠ Sampling



period. Spektrum mit f_s-vielfachen *Spiegelbilder*. Mit spektraler Verschiebung

$$x(t)e^{j2\pi f_0 t} \circ\!\!-\!\!\bullet\, X(f - f_0)$$

ergibt

$$X_s(f) = \frac{1}{T_S}\sum_{k=-\infty}^{\infty} X(f - k \cdot f_S)$$

Sampling of Band-Pass Signals

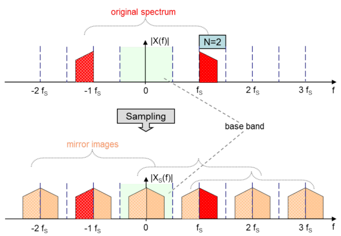
generalized sampling theorem:

$$-\frac{N+1}{2} f_S \leq f \leq -\frac{N}{2} f_S \quad \text{and}$$

$$\frac{N}{2} f_S \leq f \leq \frac{N+1}{2} f_S$$

Wird generalisierten Theorem eingehalten, kann Signal rekonstruiert werden. Zum prüfen, ob eine Sampling Frequenz für ein Band-Pass Signal gültig ist:

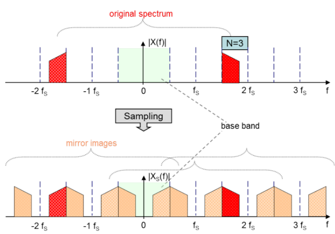
$$2 \cdot \frac{f_{\min}}{N} \geq f_S \geq 2 \cdot \frac{f_{\max}}{N+1}$$



Odd N: Verschiebung mit Kosinus $f_S/2$

$$\tilde{x}[n] = (-1)^n \cdot x[n]$$

$$(-1)^n = \cos(\pi \cdot n) = \cos(2\pi f_S/2 \cdot n \cdot T_S)$$



Digital Signals in Frequency Domain

Fourier Transformation to DFT

Discrete-Time Fourier Transform (DTFT)

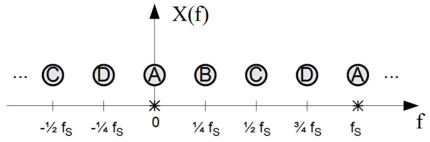
Transition to Discrete Time

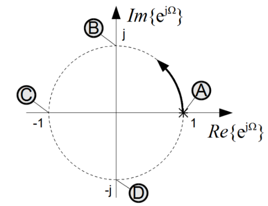
$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi f \cdot t}dt \longrightarrow X_S(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi f \cdot n \cdot T_S}dt$$

$$\Omega = 2\pi f T_s = 2\pi \frac{f}{f_S} \Longrightarrow \boxed{X(\Omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n}}$$

$\Longrightarrow$ $X(\Omega)$: **Discrete-Time Fourier Transform (DTFT)**

Ω: normalized angular frequency





Transition to Finite Measurement Level

Fourier has ∞ long measurement time $\to$ Confine to N sample points, which leads to a discrete frequency range.

Discrete frequency range:

$$0, \frac{f_S}{N}, 2\frac{f_S}{N}, ..., (N-1)\frac{f_S}{N}$$

Measurement Interval: $T = N \cdot T_S$

ⓘ Lowest capturable frequency

(With exception of any DC component)

$$f_{\min} = f_1 = \frac{1}{T} = \frac{1}{N \cdot T_S} = \frac{f_S}{N}$$

Discrete Fourier Transform (DFT)

$$\boxed{X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi n \frac{k}{N}}}$$

with $k = 0, 1, 2, ..., N-1$

Inv. Discrete Fourier Transform (IDFT)

synthesis equation: $x[n]$ is periodic at $T_S \cdot N$

$$\boxed{x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] \cdot e^{j2\pi n \frac{k}{N}}}$$

with $n = 0, 1, 2, ..., N-1$

Either DFT or IDFT require the normalization factor $1/N$ to re-obtain the original signal. $\Rightarrow$ IDFT has the normalization factor above.

Signal

Time	periodic	aperiodic
continuous	Fourier Series Discrete Line Spectrum c_n	Fourier Transform Continuous Frequency Spectrum X(f)
discrete	Discrete Fourier Transform (DFT) Discrete periodic Line spectrum X[k]	Discrete-Time Fourier Transform (DTFT) Continuous periodic Frequency Spectrum X[Ω]

- periodicity in time $\to$ discrete line spectra in frequency (Fourier & DFT)
- sampling in time $\to$ periodic in frequency (DFT, DTFT)

DFT Intuitive

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi n \frac{k}{N}}$$
$$= \sum_{n=0}^{N-1} x[n] \cos\left(-2\pi n \frac{k}{N}\right) + j \cdot \sum_{n=0}^{N-1} x[n] \sin\left(-2\pi n \frac{k}{N}\right)$$
$$= \underbrace{\sum_{n=0}^{N-1} x[n] \cos\left(2\pi n \frac{k}{N}\right)}_{\Re(X[k])} + j \cdot \underbrace{\sum_{n=0}^{N-1} x[n](-1) \sin\left(2\pi n \frac{k}{N}\right)}_{\Im(X[k])}$$

Static Correlation

Every DFT coefficient $X[k]$ is equal to the *static* correlation between the signal $x[n]$ and discrete sine and cosine functions of frequency $k f_S/N$.

Meaning: the DFT indicates how similar the signal is to harmonic oscillations with frequency k

Properties of the DFT

Important Properties

Periodicity DFT works with discrete time signal samples, the spectrum is f_S periodic.

$$\text{DFT}: X[k] = X[k+N] \qquad \text{IDFT}: x[n] = x[n+N] \text{ with } T = NT_S$$

Symmetry DFT of a real-valued signal is symmetric around the point $k = N/2$

$$X\left[\frac{N}{2} + m\right] = X^*\left[\frac{N}{2} - m\right]$$

Time/Frequency Shifting Shifting a periodic time sequence corresponds to a linear phase offset to all spectral values

$$x[n + n_0] \quad \circ\!\!-\!\!\bullet \quad e^{j2\pi \cdot n_0 \frac{k}{N}} \cdot X[k]$$

The inverse is also true → mult. complex exp. in time leads to frequency shift

$$e^{j2\pi \cdot n_0 \frac{k}{N}} \cdot x[n] \quad \circ\!\!-\!\!\bullet \quad X[k - k_0]$$

Modulation Direct consequence of frequency shift → modulation property

$$\cos\left(2\pi k_0 \frac{n}{N}\right) \cdot x[n] \quad \bullet\!\!-\!\!\circ \quad \frac{1}{2}(X[k+k_0] + X[k-k_0])$$

Parseval Theorem left side equals to energy of signal → right side has use for SNR (separate noise frequency from signal frequency)

$$\frac{1}{N}\sum_{n=0}^{N-1} x[n]^2 = \sum_{n=0}^{N-1} \left|\frac{X[k]}{N}\right|^2$$

Correspondence of Conv. and Multi. fast conv. → $\text{IDFT}(\text{DFT}(x[n]) \cdot \text{DFT}(y[n])))$

$$x[n] \circledast_N y[n] \quad \circ\!\!-\!\!\bullet \quad X[k] \cdot Y[k] \quad (k = 0, 1, \ldots N-1)$$

Range of Validity of the DFT

aperiodic $x[n]$ all signal values $x[n]$ are zero outside the range $0 \le n \le N$. DFT samples the DTFT at discrete points of normalized angular frequency:

$$X[k] = X(\Omega)|_{\Omega = 2\pi \frac{k}{N}}$$

IF NOT (range outside $\neq 0$) → DFT = approximation of DTFT → solution: *windowing*

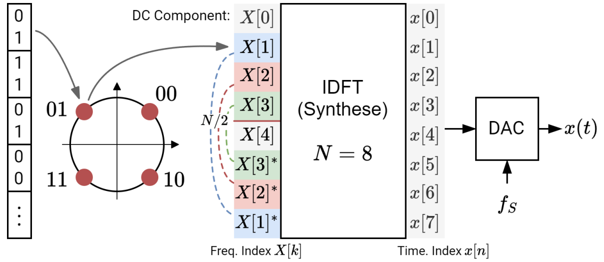
periodic $x[n]$ measurement interval $N \cdot T_S$ is an integer multiple of the period duration of $x[n]$

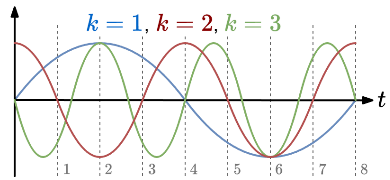
OFDM Principle

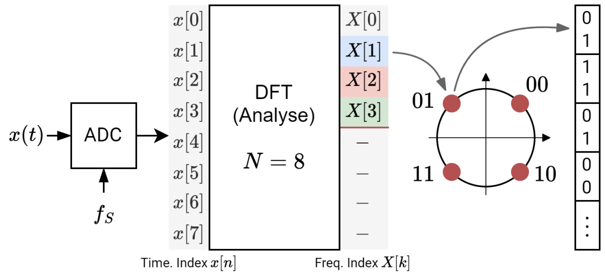
Orthogonal Frequency Division Multiplexing

Bits are spread across different frequencies.

1. bits are converted to phase (QPSK)
2. the result → IDFT 3. $x[n] \to x(t)$ via DAC



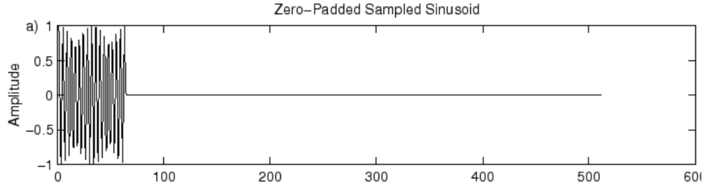


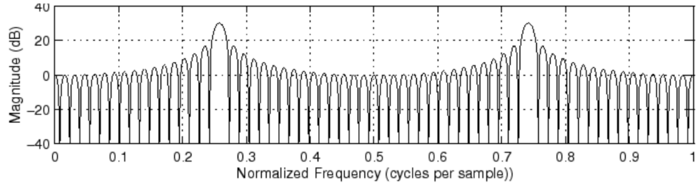


Practical Application Aspects of the DFT

DFT and Zero-padding

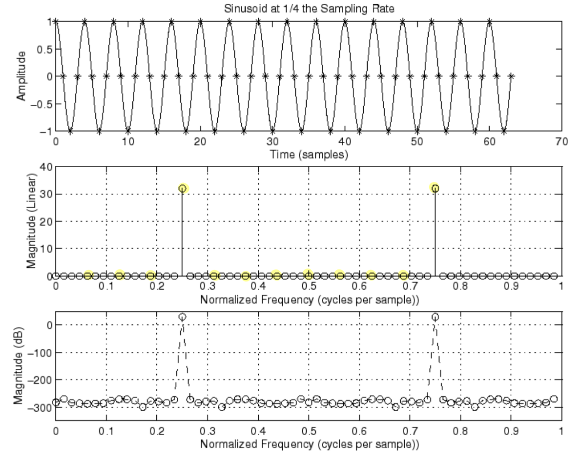
- Extending signal(t) with zeros → better interpolation (thinner frequency *bins*)
- does not modify DTFT $X[\Omega]$, but provides additional sample points along Ω
- Rectangular window of length N → convolution of $X[k]$ with $\sin(x)/x$ (lobes)
- Important lobe-structure characteristics
  - **Width of the main lobe** (example: ≈ 0.03 cycles per sample)
  - **Attenuation of the first side lobe** relative to main lobe (≈ 12dB)



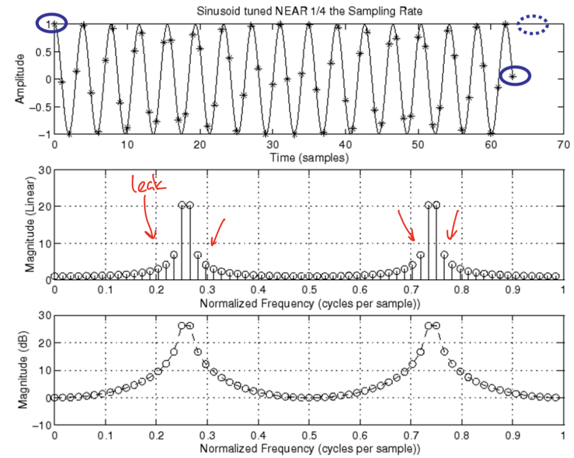


Choice of Measurement Interval & Leakage Effect

Example: $N = 64, f_0 = f_S/4, T = N \cdot T_S = 16 \cdot T_0$ (peak at $k = 16$ & $k = 48$)



Example: $f_0 = f_S/4 + f_S/128$ → measurement interval no integer multiple of the period duration: **Leakage effect**:



DFT and Windowing

- DFT applies rectangular window N samples
- Applying the *Blackman Window* and afterwards appending zeros
  - Reduces virtual periodic continuation of the signal „outside" of signal, thus reducing the leakage effect.

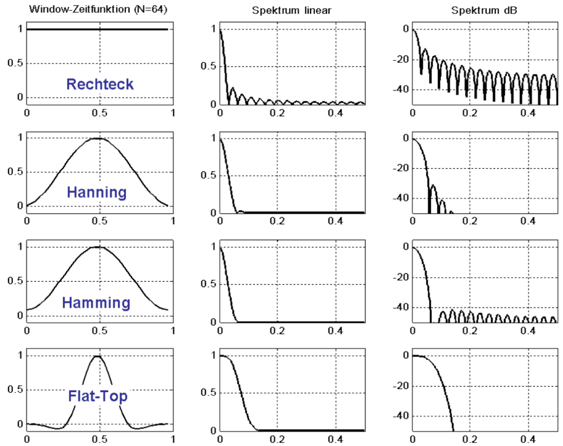
Choice of Windowing Function

Choice Compromise

Choice of Window function leads to a compromise between the attenuation of leakage and spectral resolution in the spectrum $X[k]$

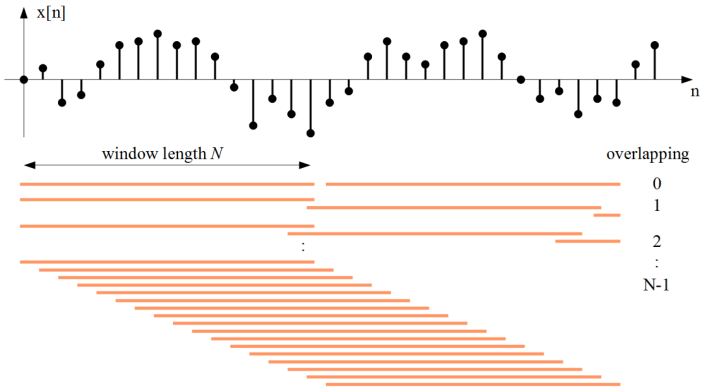
- **Narrow main lobe**: higher the spectral resolution for $X[k]$
- **Higher the side lobe attenuation**: better suppression of leakage in $X[k]$

$\Longrightarrow$ **Ideal**: DC-function $\rightarrow$ indefinitely small main lobe, no side lobes ($N \rightarrow \infty$)



Short-Time DFT

- continuous evaluation of the frequency spectrum of short signal sections
  - Allows the observation of frequency spectrum over time
  - **BUT** more computation required $\rightarrow$ solution: FFT



Fast Fourier Transformation (FFT)

Fast DFT

Complexity of the FFT

Divide-and-Conquer principle

- **Divided** get either N sample values (**decimation-in-time**) or N spectral values (**decimation-in-frequency**)
  - Split values recursively into r sub-sequences (r: radix) $\rightarrow$ radix-2 algo often used

$$N = 2^L \text{ where } L \text{ is some integer}$$

- N almost always a power of two

$$\text{DFT}: \quad [N^2]_{\text{cpl.Mul.}} \qquad + [N^2 - N]_{\text{cpl.Add.}}$$

$$\text{FFT}: \quad \left[\frac{N}{2} \cdot \log_2(N)\right]_{\text{cpl.Mul.}} \qquad + [N \cdot \log_2(N)]_{\text{cpl.Add.}}$$

assuming $T_{\text{compute,Add}} = T_{\text{compute,Mul}}$: speedup factor$_{\text{FFT}} = \dfrac{8N-2}{5 \cdot \log_2(N)} \approx 1.5 \dfrac{N}{\log_2(N)}$

Properties of the Twiddle Factors

In order to reduce the computational effort we introduce the **twiddle factor** $W_N = e^{-j2\frac{\pi}{N}}$ and can write the DFT new:

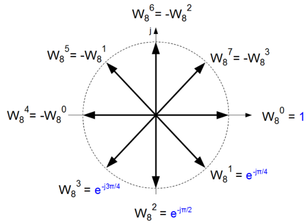
$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{n \cdot k}, \quad k = \{0, 1, 2, ..., N-1\}$$

Periodicity W_N^k can evaluate to N different numbers only

$$W_N^{k+N} = W_N^k$$

Symmetry Apart from sign, every W_N^k takes on only $N/2$ different values within each period.

$$W_N^{k+N/2} = -W_N^k$$



MCU only requires $\frac{N}{2} \cdot 2$ $(\Re \& \Im)$ space.

Radix-2 decimation-in-time FFT

Splitting the twiddle-factor DFT up into odd and even yields two new sequences of length $N/2$:

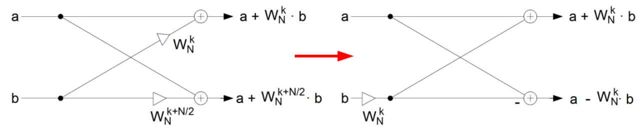
$$X[k] = \underbrace{\sum_{n=0}^{N/2-1} x_1[n] W_N^{2nk}}_{x_1 \rightarrow n \text{ even}} + \underbrace{\sum_{n=0}^{N/2-1} x_2[n] W_N^{(2n+1)k}}_{x_2 \rightarrow n \text{ odd}}$$

$$\text{introducing } W_N^2 = W_{N/2}: \quad = \underbrace{\sum_{n=0}^{N/2-1} x_1[n] W_{N/2}^{nk}}_{X_1[\tilde{k}]} + W_N^k \cdot \underbrace{\sum_{n=0}^{N/2-1} x_2[n] W_{N/2}^{nk}}_{X_2[\tilde{k}]}$$

$\Rightarrow X_1[\tilde{k}], X_2[\tilde{k}]$: $N/2$-point DFT $\longrightarrow \tilde{k} = k \bmod N/2$ (limit k-range to meaningful $N/2$)

Recursively applying the splitting procedure leads to $\frac{N}{2}$ 2-point DFTs:

$$X[k] = \sum_{n=0}^{0} x_1[n] W_2^{nk} + W_2^k \cdot \sum_{n=0}^{0} x_2[n] W_2^{nk} = \underline{x_1[0] + W_2^k \cdot x_2[0]}, \qquad k = \{0, 1\}$$

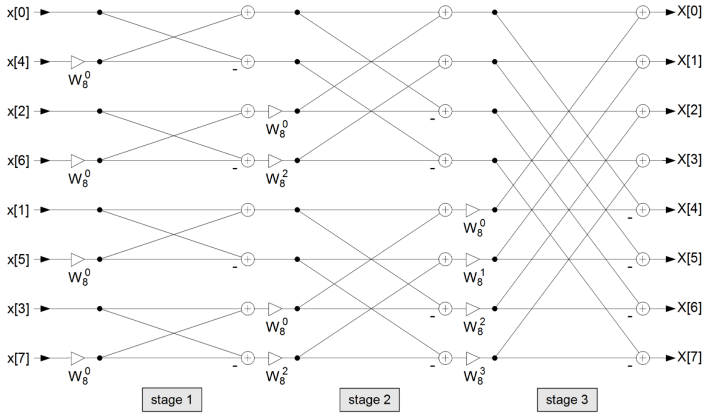


- Butterfly structure requires $\log_2(N)$ processing stages ($N = 8 \rightarrow 3$ stages)

Efficient FFT Implementation

1. As soon as the butterfly operation has been performed, input pair can be re-used to store the calculated output-pair, thus performing the entire FFT **in-place**.
2. Order of input values is **bit-reversed**: 0 (`000`), 4 (`001` $\rightarrow$ `100`), 2 (`010`), 6 (`110`), 1, 5, 3, 7.

Matlab command `bitreorder` for bit-reversed order



The Goertzel Algorithm

Goertzel is used, if only an individual $X[k]$ of all N spectral components is required:

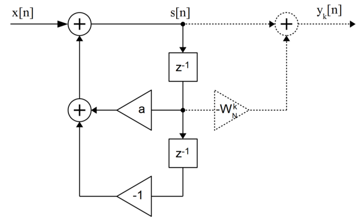
$$s[n] = x[n] + a \cdot s[n-1] - s[n-2]$$

$$y_{k[n]} = s[n] - W_N^k \cdot s[n-1]$$

$$P_k = 2\left|\frac{X[k]}{N}\right|^2 = \frac{2}{N^2}(\Re^2 + \Im^2)$$

$$f_k = k\frac{f_S}{N}$$

$$a = 2 \cdot \cos\left(2\pi\frac{k}{N}\right), \quad W_N^k = e^{-j \cdot 2\pi k/N}$$



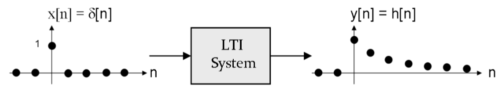
Digital LTI Systems

- **Definition** of LTI Systems
  - **Linearity**: $y[n] = k_1 \cdot S\{x_1\} + k_2 \cdot S\{x_2\} = S\{k_1 \cdot x_1 + k_2 \cdot x_2\}$
  - **Time-Invariance**: $x[n] \rightarrow y[n] \quad \Longrightarrow \quad x[n-d] \rightarrow y[n-d]$
- **Allowed Operations**
  - Multiplication of a signal with a constant: $x[n] \cdot a$
  - Addition of two signals: $x[n] + y[n]$
  - Time delay of a signal by $k \cdot T_s$: $x[n - k \cdot T_S]$

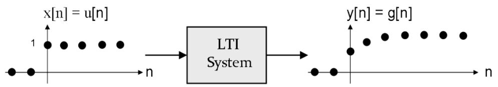
System Descriptions in the Time Domain

Impulse Response

System Identification, Measurement



$$y[n] = \sum_{i=-\infty}^{\infty} x[i] \cdot h[n-i] = x[n] * h[n] \rightarrow \text{Linear Convolution}$$



$$h[n] = g[n] - g[n-1] \quad \Longleftrightarrow \quad g[n] = \sum_{k=0}^{n} h[k]$$

Difference Equation — System Implementation (algorithm)

$$y[n] = \sum_{k=0}^{N} b_k \cdot x[n-k] - \sum_{k=1}^{M} a_k \cdot y[n-k]$$

System Order

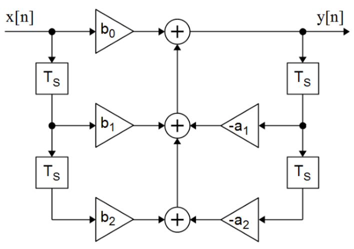
System order is defined by $\max(N, M)$

Recursive

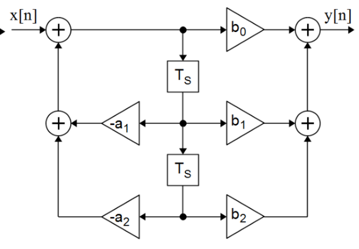
A system is recursive, when $M \geq 1$.

Signal-Flow Diagram — System Implementation (architecture)

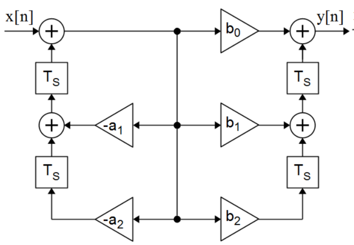
Direct Form 1



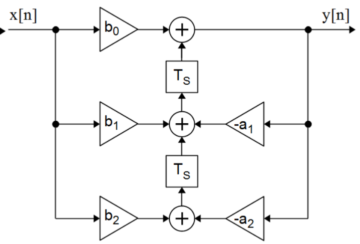
Direct Form 2



Tranposed Direct Form 1



Transposed Direct Form 2



System Descriptions in the Frequency Domain

Transfer Function — Coupling Analysis and Implementation

$$y[n] = \sum_{k=0}^{N} b_k \cdot x[n-k] - \sum_{k=1}^{M} a_k \cdot y[n-k]$$

$$Y(z) = \sum_{k=0}^{N} b_k \ z^{-k} \cdot X(z) - \sum_{k=1}^{M} a_k \ z^{-k} \cdot Y(z)$$

z-Transfer-Function

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{n=0}^{N} b_n z^{-n}}{\sum_{m=0}^{M} a_m z^{-m}}$$

Pol/Zero-Plot — Intuitive Analysis and Design

$$H(z) = K_0 \cdot \frac{(z-z_1)(z-z_2)...(z-z_N)}{(z-p_1)(z-p_2)...(z-p_M)} \cdot z^{M-N}$$

z_i zeros ; p_i poles

K_0 gain

$\boxed{(M > N) \wedge b_0 \neq 0}$ → $M - N$ additional zeros at $z = 0$ & only this case holds $K_0 = b_0$

$\boxed{N > M}$ → $N - M$ additional poles at $z = 0$

Causal LTI System is stable if $|p_i| < 1, i = 1, ..., M$ (all poles within the unit circle of z-plane)

Frequency Response — System Identification, Analysis and Design

$$h[n] \circ\!\!-\!\!\bullet \ H[\Omega] \quad ; \quad H(\Omega) = |H(\Omega)| \cdot e^{j \cdot \varphi(H(\Omega))} \quad ; \quad |H(\Omega)|_{\text{dB}} = 20 \cdot \log_{10}(|H(\Omega)|)$$

$|H(\Omega)|$: amplitude response ; $\varphi(H(\Omega))$ phase response ;

- Frequency components in input are delayed differently, the output suffers from distortions → Therefore, linear phase response $\varphi(H(\Omega)) = -K \cdot \Omega$ is desirable, since only then all frequency components are delayed: **group delay**

$$\tau_g = -\frac{d\varphi(H(\Omega))}{d\Omega}$$

Any LTI system reacts to a sinusoidal input signal with a sinusoidal output signal of the same frequency:

$$x[n] = \cos(2\pi f_0 \cdot n \cdot T_S) \Longrightarrow y[n] = |H(\Omega_0)| \cdot \cos(2\pi f_0 \cdot n \cdot T_S + \varphi(H(\Omega_0)))$$

The phase and amplitude can be extracted through:

$$|Y(\Omega)| = |X(\Omega)| \cdot |H(\Omega)| \quad ; \quad \varphi(Y(\Omega)) = \varphi(X(\Omega)) + \varphi(H(\Omega))$$

Relation between frequency response and transfer function

- DTFT and z-transform relation $\Rightarrow z = r \cdot e^{j\Omega}$
- Frequency response = DTFT of impulse response $\Rightarrow H(\Omega) = H(z)|_{z=e^{j\Omega}}$
- To obtain frequency response by evaluating:

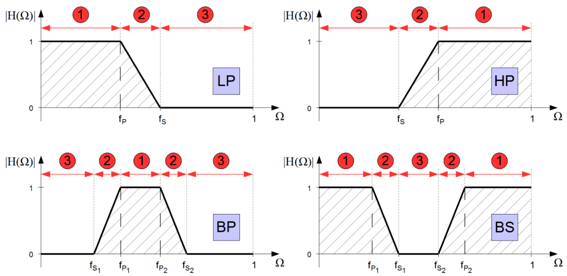
Amplitude response

$$|H(z)| = |K| \frac{\prod_{n=1}^{N} |z - z_n|}{\prod_{m=1}^{M} |z - p_M|} |z|^{M-N}$$

Phase response

$$\varphi(H(\Omega)) = \sum_{k=1}^{N} \varphi(z - z_k) - \sum_{k=1}^{M} \varphi(z - p_k) + \sum_{k=N+1}^{M} \varphi(z)$$

Design of Digital Filters



$a_k, b_k \longrightarrow$ filter coefficients $\qquad N, M \longrightarrow$ filter order

FIR Filter

Definition and Properties

FIR filter of order N has the transfer function & impulse response:

$$H(z) = b_0 + b_1 z^{-1} + ... + b_N z^{-N}$$

$$h[n] = \{b_0, b_1, ..., b_N, 0, 0, ...\} \qquad \text{size } N+1$$

with $N + 1$ coefficients.

- **Stability** per definition, all poles are at $z = 0$
- **Linear Phase**: easier to realize a linear-phase transfer characteristics (group delay)
- **Implementation**: easy implementation on HW and SW

Disadvantages: higher order requires more computational effort.

Other names: all-zero filter, transversal filter, moving-average filter

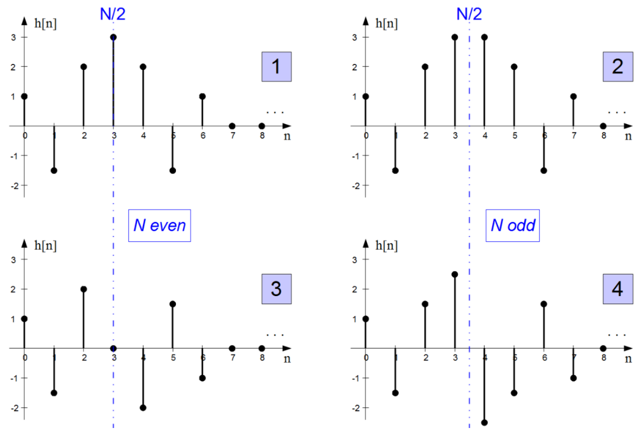
Symmetric FIR Filters

A FIR filter is symmetric when $\boxed{b_i = \pm b_{N-1}, \quad i = 0, 0, ..., N}$

- in case of $+$: (mirror-)symmetric
- in other cases: anti-symmetric

Lineare Phase Response for all symmetric FIR filters

1 All symmetric FIR filters feature a linear phase response within their **pass** band (group delay): $\tau_g = \frac{N}{2} \cdot T_S$



Type	Symmetry	Order N	$\lvert H(f=0)\rvert$	$\lvert H(f=f_S/2)\rvert$	$H(\Omega)$[1]
1	$h[n] = h[N-n]$	even	any	any	$e^{-j\Omega\frac{N}{2}} \cdot H_{\text{zp}(\Omega)}$
2	(symmetric)	odd	any	0	
3	$h[n] = -h[N-n]$	even	0	0	$e^{-j(\Omega\frac{N}{2}-\frac{\pi}{2})} \cdot H_{\text{zp}(\Omega)}$
4	(anti-symmetric)	odd	0	any	

[1]: transfer function of symm. FIR are the product of a linear-phase term and some real-valued transfer function $H_{\text{zp}(\Omega)}$ (**zp**: zero-phase filter)

$\Longrightarrow$ anti-symmetric: constant 90° phase offset

Stop Band 180° Jump

In the stop band of a symm. FIR filter there can be 180°-phase-jumps. Such discontinuities in phase response occur at a pair of complex-conj. zeros at the unit circle. This are often tolerated in favor of sufficient attenuation in the stop band

Type	low-pass (LP)	high-pass (HP)	band-pass(BP)	band-stop (BS)
1	yes	yes	yes	yes
2	yes	–	yes	–
3	–	–	yes	–
4	–	yes	yes	–

Window Design Method

Matlab: `fir1`

The *Window Design Method* always yields low pass filters → other filters are done via sum and difference of low-pass filters at different cut-off frequencies.
Start of with a desired frequency response $H_{d(\Omega)}$ of an ideal TP-filter with cutoff at f_C:

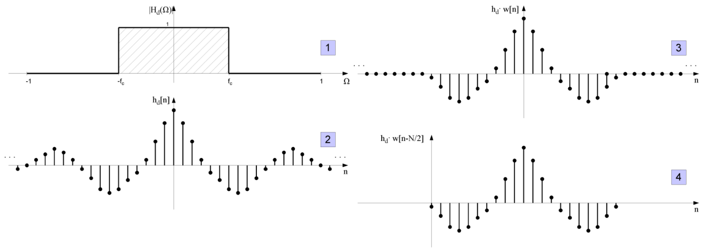
$$h_{d\mathrm{TP}}[n] = \frac{\sin(\Omega_C \cdot n)}{\pi \cdot n} \bullet\!\!-\!\!\circ \text{ rectangular signal in freq. domain}$$

- **Restrictions**: finite length ideal impulse response (corresponds to multiplication with □-window **/** convolution with sinc-function) → overshoot at edges of pass and stop bands
  - persists for $N \to \infty$ (only helps to reduce the width of the transition band)
  - Solution: use different windowing functions to smooth the overshoot at a cost of wider transition bands

Example High-Pass Filter

TP with cutoff at $f_S/2$ minus TP with cutoff at f_C:

$$\begin{aligned} h_{d\mathrm{HP}}[n] &= \frac{\sin(\pi \cdot n)}{\pi \cdot n} - \frac{\sin(\Omega_C \cdot n)}{\pi \cdot n} \\ &= \{..., -h_{d\mathrm{TP}}[-2], -h_{d\mathrm{TP}}[-1], 1 - h_{d\mathrm{TP}}[0], -h_{d\mathrm{TP}}[1], -h_{d\mathrm{TP}}[2], ...\} \end{aligned}$$



Steps of the window design method for FIR filters: Ideal low-pass frequency response (1), ideal low-pass impulse response (2), windowed (3) and shifted (4) practical low-pass impulse response.