Digitale Signalverarbeitung

Zusammenfassung

Andreas Ming / Quelldateien

Table of contents

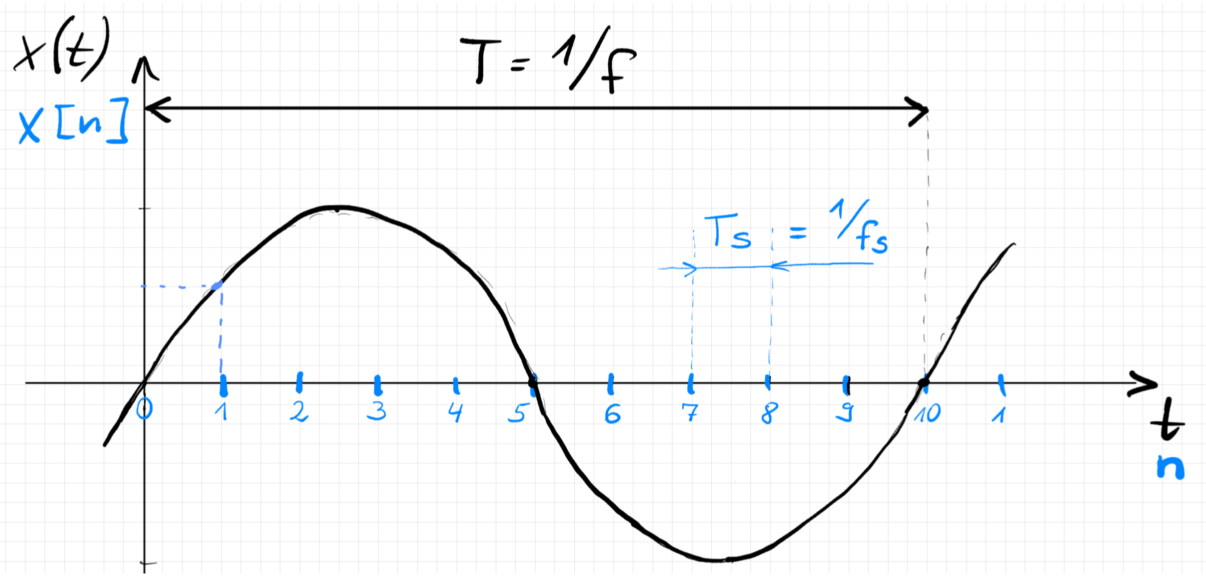
Digital Signals in the Time Domain

Signal Analysis

Sampling of Analog Signals

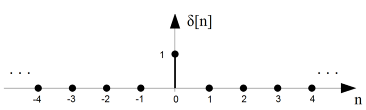
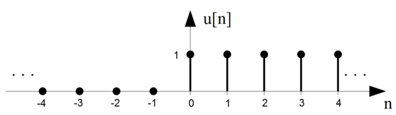
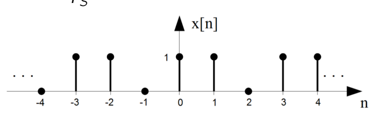
By sampling $x(t)$ in the interval of T_S we get the sequence of signal values $x[n]$ with $-\infty \leq n \leq +\infty$

$$x(n \cdot T_S) = x[n]$$



Signal	Property
causal	$x[n] = 0$ for $n < 0$
real	$x[n]$ Real
complex	Re & Im or Amplitude & Phase

Basic Digital Signals

unit impulse	unit step	periodical signal
$\delta[n] = \begin{cases} 0 : n \neq 0 \\ 1 : n = 0 \end{cases}$	$u[n] = \begin{cases} 0 : n < 0 \\ 1 : n \geq 0 \end{cases}$	$x[n] = x\left[n + \frac{T_0}{T_S}\right]$ with $\frac{T_0}{T_S} = k$
		

There is also a **complex hamonic** sequence with the period duration of $T_0 = \frac{1}{f_0}$

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

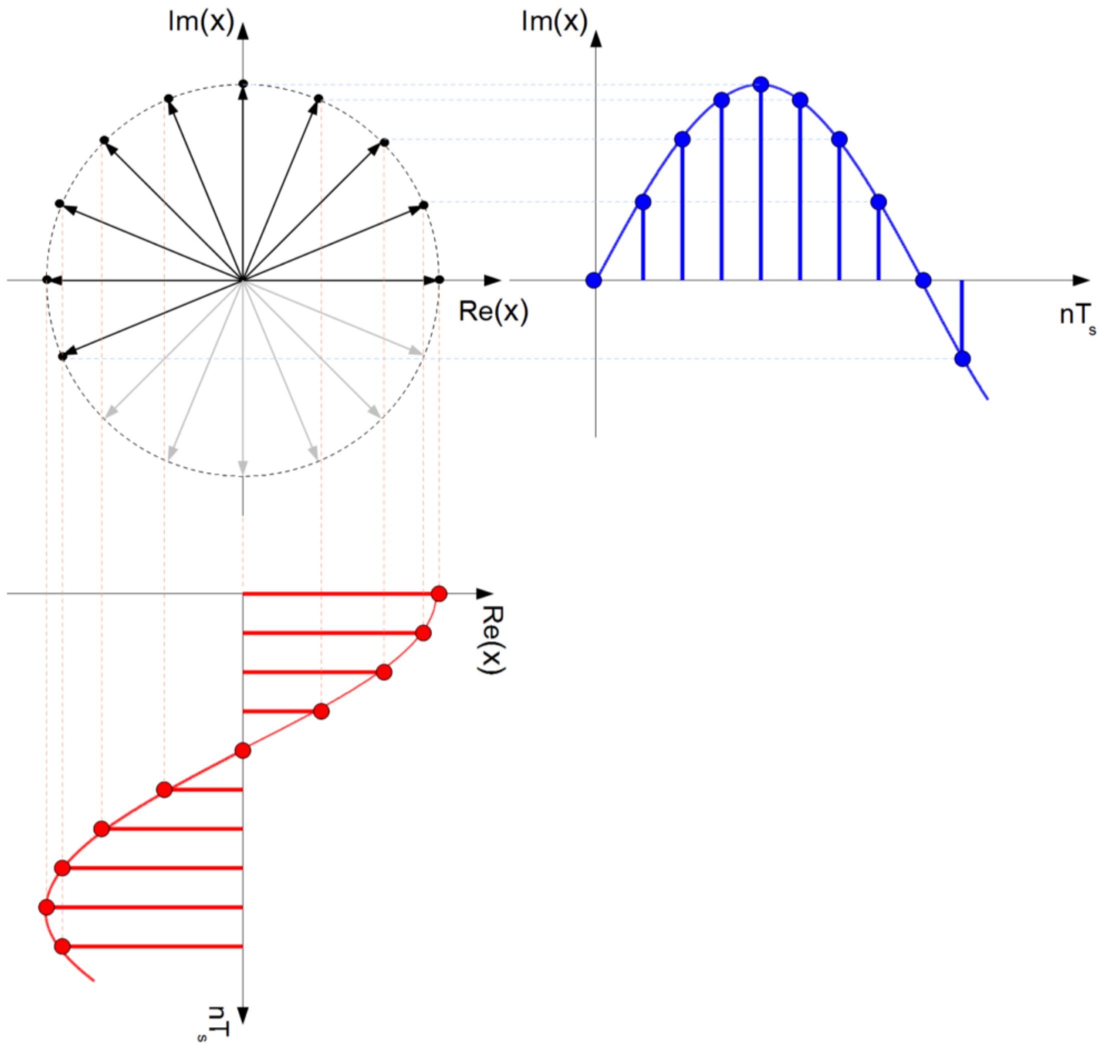


Figure 1: Complex hamronic sequence with period duration $T_0 = 16 \cdot T_S$

Statistical Signal Parameters

Stochastic signals must be qualified by *statistical signal parameters* within the **observation interval** $T = N \cdot T_S$.

expected / mean	quadratic mean	variance
DC-component	average power *(w/ DC)*	average power *(w/o DC)*
$\mu_x = \frac{1}{N}\sum_{i=0}^{N-1} x[i]$	$\rho_x^2 = \frac{1}{N}\sum_{i=0}^{N-1} x[i]^2 = P_{avg}$	$\sigma_x^2 = \frac{1}{N}\sum_{i=0}^{N-1} (x[i] - \mu_x)^2 = P_{AC}$

Signal Operations

Correlation

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

	cross-correlation	auto-correlation
Linear	$r_{xy}[n] = \sum_{i=-\infty}^{\infty} x[i]y[i+n]$	$r_{xx}[n] = \sum_{i=-\infty}^{\infty} x[i]x[i+n] = P_{avg}$

For **linear correlation** the resulting length of r_{xy} equals

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

and the range of shifts for the computation is given by

$$-N_x + 1 \leq n \leq N_y - 1$$

For signals differing in length, zero-padding can be applied.

Convolution

The *Convolution* involves folding the time-displaces signal around the point $n = 0$

$$z[n] = \sum_{i=-\infty}^{\infty} x[i]y[-i+n] \tag{0.1}$$

A convolution equals a polynomial multiplication.

The range of shifts for the computation is given by

$$0 \leq n \leq N_x + N_y - 2$$

The Convolution described in Equation 0.1 is called a **linear convolution** and can be applied to two signals of different length

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

```
z = conv(x, y)
```

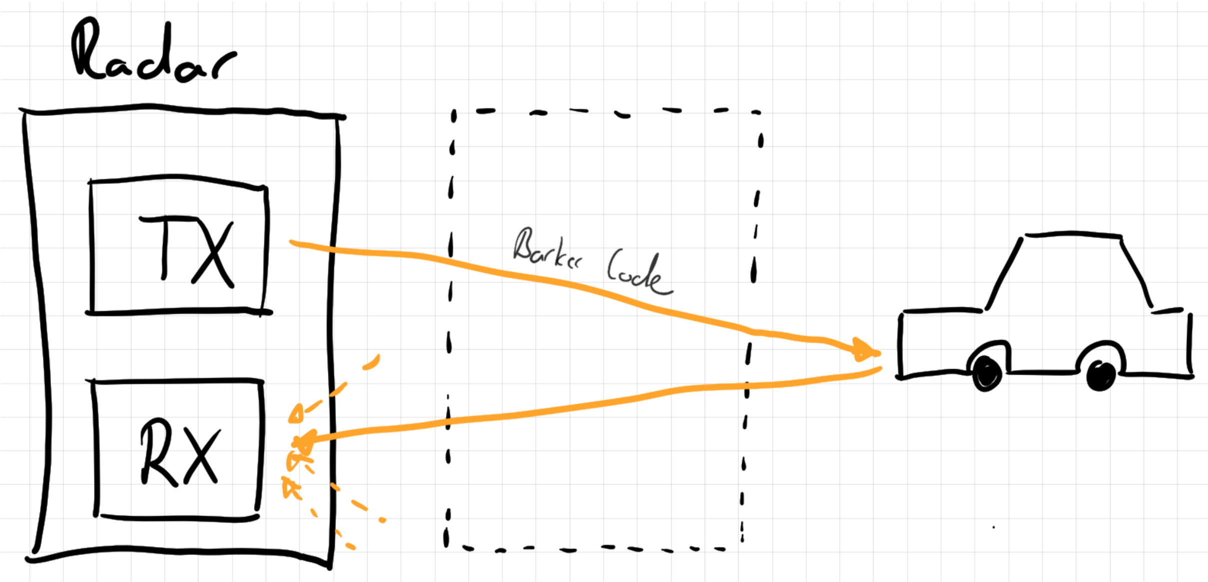
There is also the **circular convolution** which requires both signals to be of equal length N. If necessary, *zero padding* can be applied. The resulting signal then also is of length N.

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

The circular convolution corresponds to matrix multiplication In order to compute $x[n] \circledast_N y[n]$, the NN-matrix constructed from circular shifting y must be multiplied with vector x.

```
z = convmtx(x,y)
```

Anwendung: Radar



Um bei einem Radar nur auf das gewünschte Signal zu reagieren, also auf das eigene, wird vom Radar ein Barker-Code ausgesendet. Über Korrelation kann so die Laufzeit eindeutig zugeordnet werden.

> **Barker-Code**
>
> Es können auch andere Codes ausgesendet werden, die verwendeten Signale müssen jedoch sehr gute Autokorrelationseigenschaften aufweisen.

Analog-to-Digital & Digital-to-Analog Conversion

Steps of A/D- and D/A-Conversion

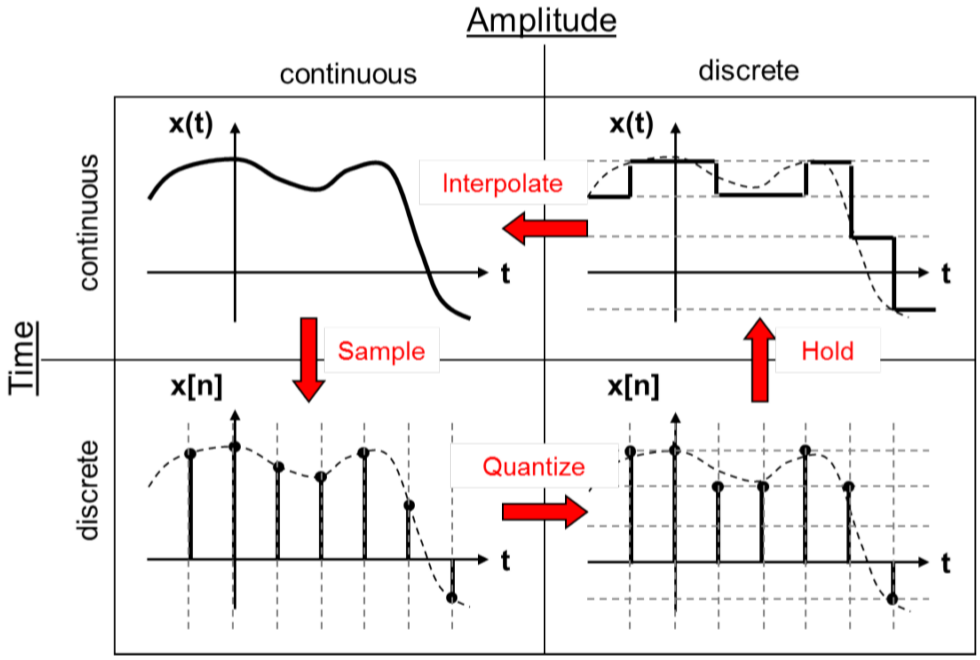


Figure 2: Signal classification in a/d- and d/a-conversion

A/D

Sample: Signal values are recorded at sampling rate f_S. This yields a train of pulses.

Quantize: The discrete signal values are mapped to a given number of quantization levels.

Code: The quantified values can be stored in a coded way. DSPs most often store the quantified values.

D/A

Decode: The coded samples are converted back into a suitable representation for the digital-to-analog conversion method used.

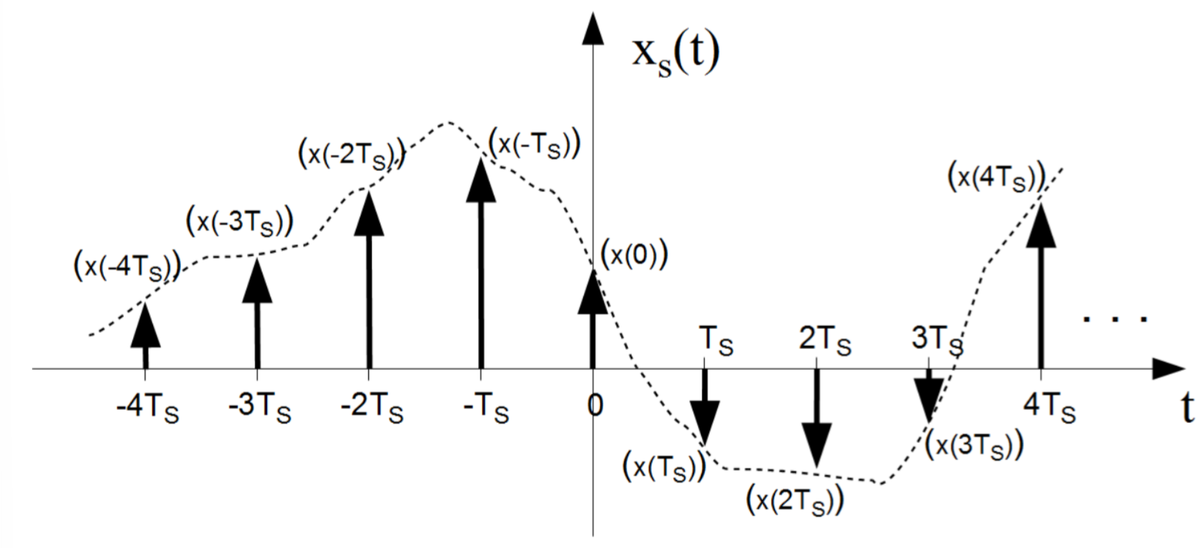
Hold: A momentary discrete signal value is constant over the sample period T_S.

Interpolate: The continuous staircase signal form is smoothed by a low-pass-filter.

Sampling and Aliasing

Sampling a time-continuous signal $x(t)$ corresponds to a multiplication with a Dirac impulse series. The resulting signal $x_S(t)$ can be regarded as a train of weighted Dirac impulses.

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



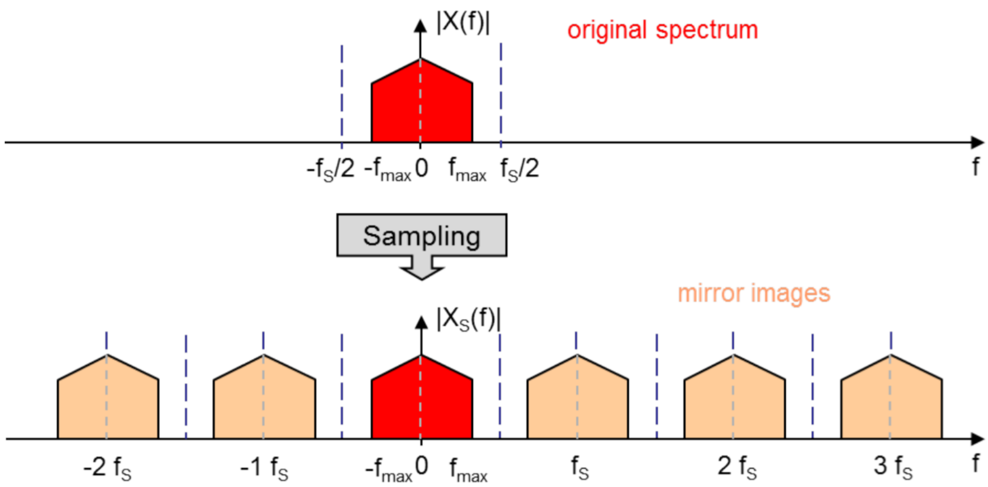
Through the application of the Fourier property $x(t)e^{j2\pi f_0 t} \circ\!\!-\!\!\bullet X(f - f_0)$ we obtain the frequency spectrum of the sampled signal as

$$X_S(f) = \frac{1}{T_S} \sum_{k=-\infty}^{\infty} X(f - kf_S)$$

! Observation

The frequency of the analog signal $x(t)$ consists of the original spectrum $X(f)$ superimposed (*überlagert*) by mirror images of the spectrum

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

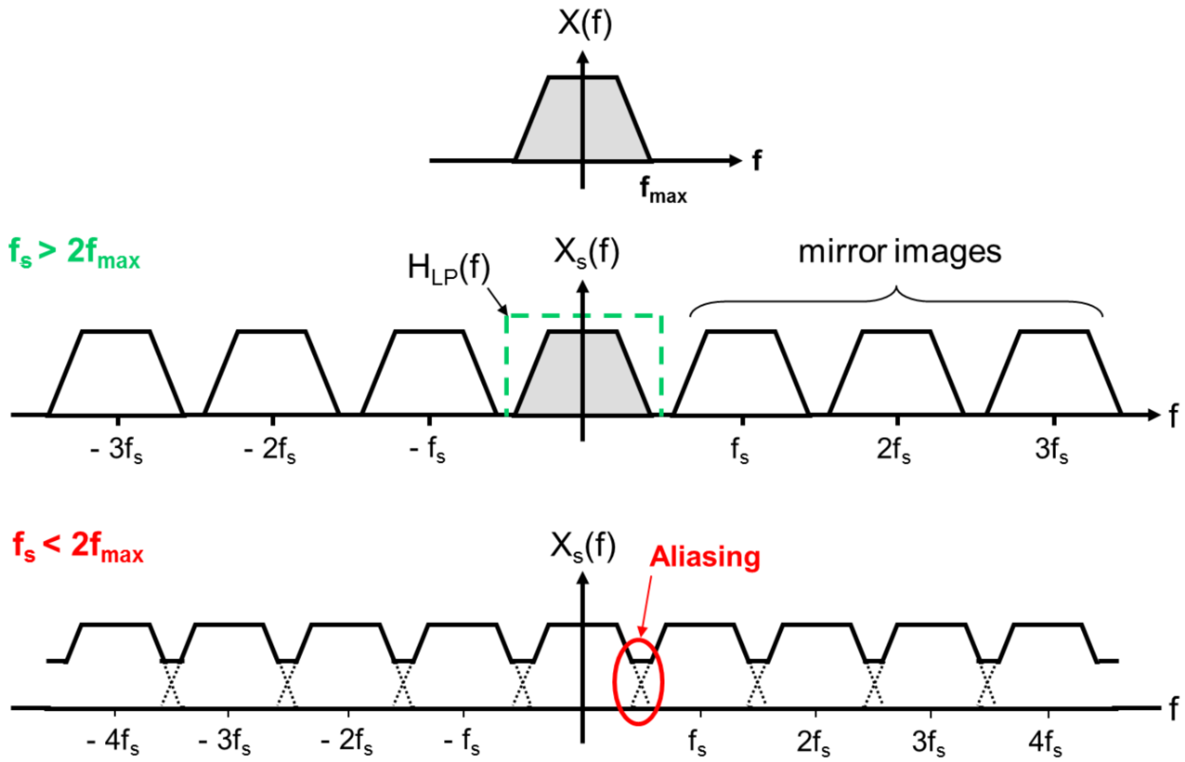


Aliasing

Sampling Theorem

An analog signal $x(t)$ with $X(f) = 0$ for $|f| > |f_{max}|$ is uniquely defined by its sample values $x[n] = x(nT_S)$, if for the sampling frequency $F_S = \frac{1}{T_S}$ holds:

$$f_s > 2 \cdot f_{max}$$



Band-Pass Sampling

$x(t)$ can be perfectly reconstructed if an integer $N \geq 0$ exists, such that $X(f) = 0$ holds for all frequencies f outside

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

For a given band-pass signal with given limits f_{min} and f_{max} it can be checked if the sampling frequency f_S can be used ($N \geq 1$)

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

For sampling with $N =$ **even** we get the mirror spectrums

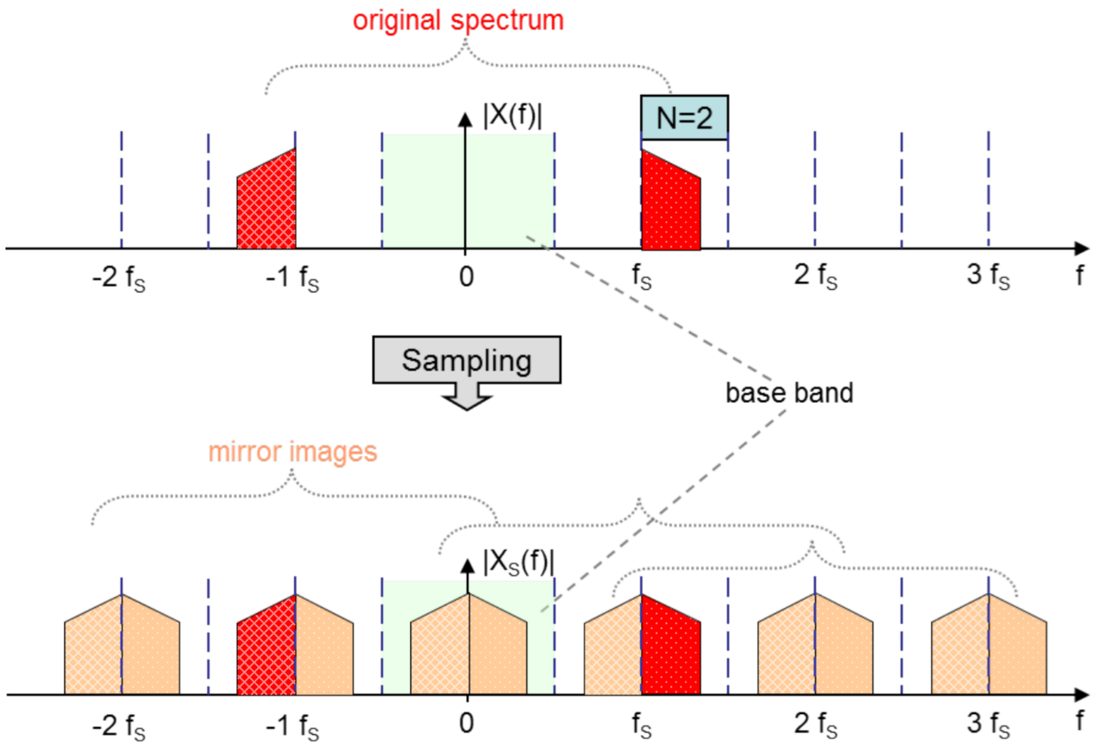


Figure 3: Band-pass sampling for even N

For sampling with $N =$ **odd** we get the mirror spectrums

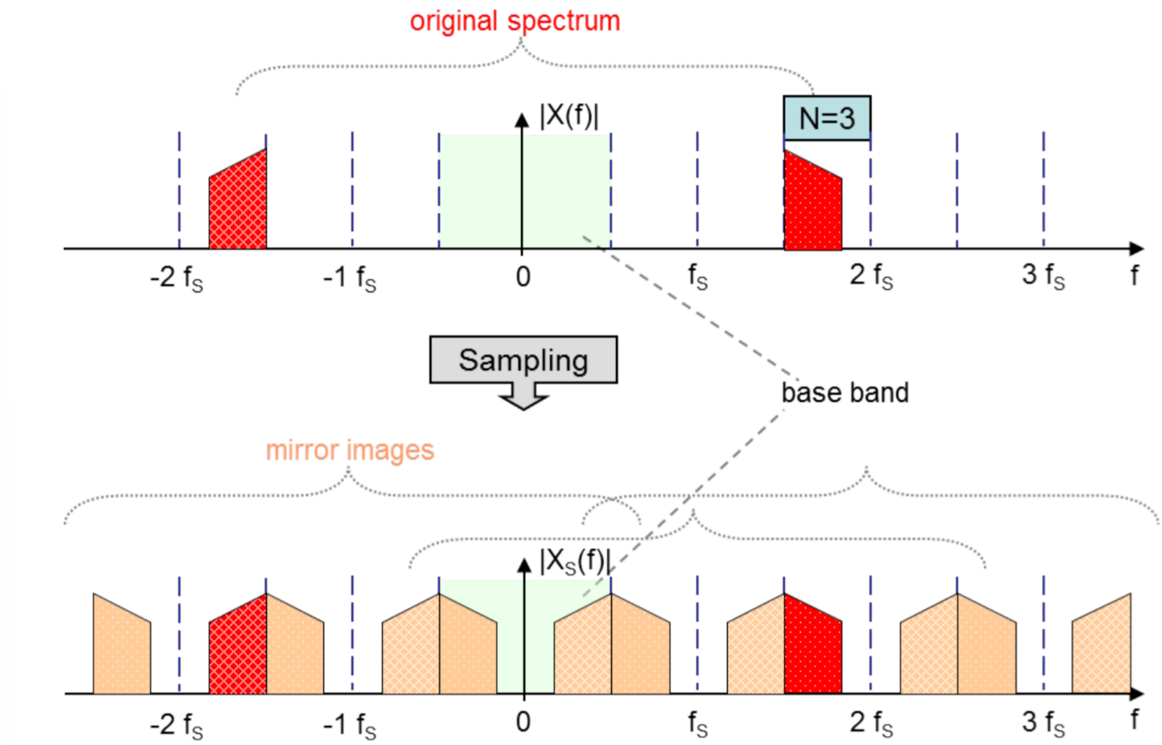


Figure 4: Band-pass sampling for odd N

! Spectrum Correction

Note that for N odd, the original spectrum appears "inverted" in the base band. The original structure of the spectrum can be re-obtained by changing the sign of every second sample of the time-domain sequence, i.e.

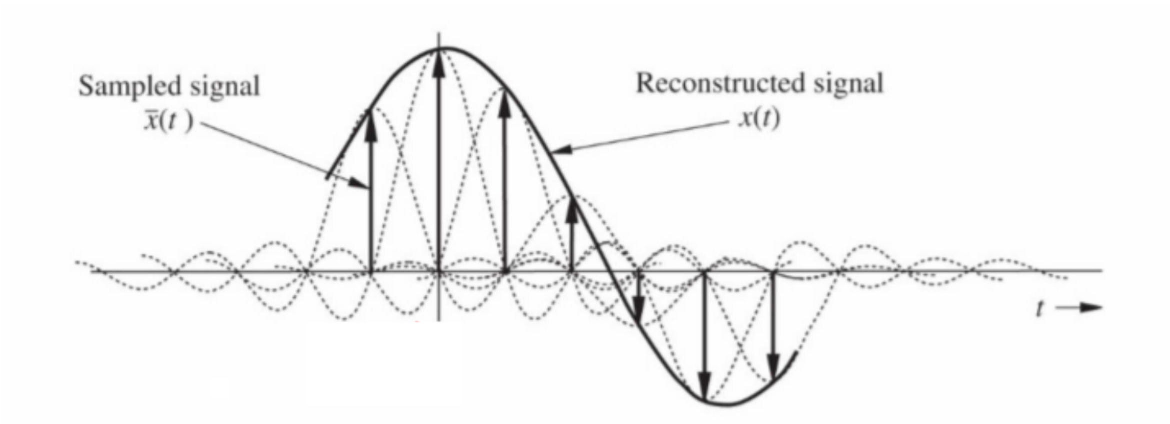
$$\tilde{x} = (-1)^n \cdot x[n]$$

Reconstruction

Ideal Reconstruction

Sampled signals w/o Aliasing can be *theoretically* reconstructed error-free. For this all mirror-spectra must be eliminated by a ideal low-pass filter. Because of the property $\text{rect}\left(\frac{t}{T}\right) \circ\!\!-\!\!\bullet |T| \cdot \text{si}(\pi T f)$ the ideal interpolation equals

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT_S) \cdot \text{sinc}(\pi f_S(t - nT_S))$$



i Ideal values

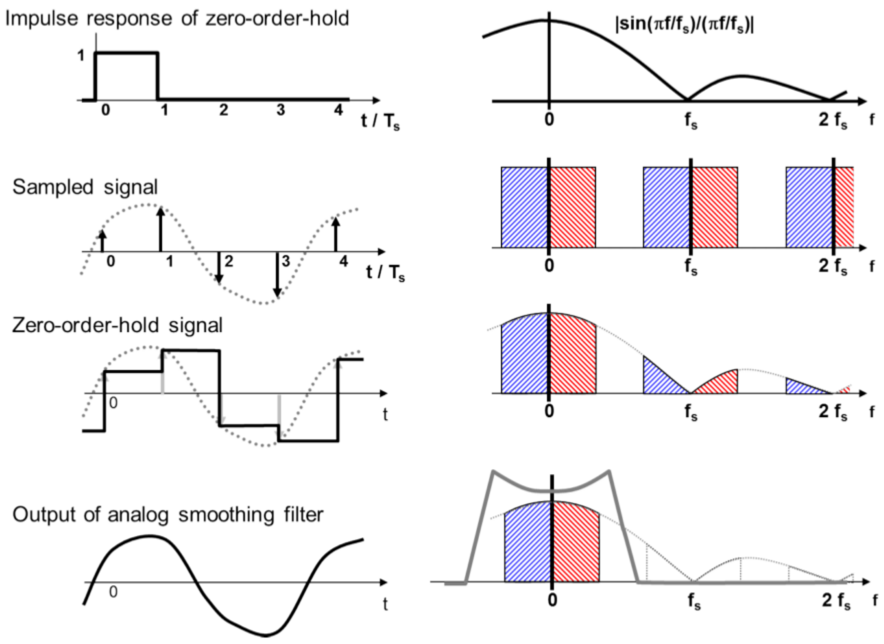
At the points $t = nT_S$ all values of except of $x(nT_S)$ equal 0. Thus at every point of $x(nT_S)$ the signal reaches the right value.
Caution! because of the infinit sum of sinc-pulses, the values between $x(nT_S)$ aren't particularly correct. Also the further to the "edge" of x you get, the more inaccurate it gets.

Practical Reconstruction

In practice Reconstruction is very often done with a simple *zero-order-holder* (ZHO). Such operation holds each sample value constant over a subsequent sample interval T_S. This results in a stair-case waveform, thus making a very poor low-pass filter. For this reason a analog low-pass filter is usually implemented.

Without analog filtering the **SNR** can be approximated as

$$\text{SNR} \approx 6dB \cdot \log_2\left(\frac{f_S}{f_0}\right) - 11dB$$

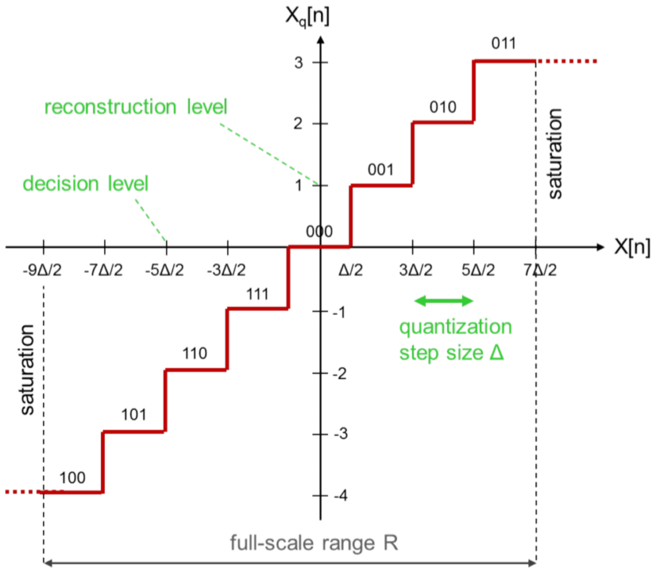
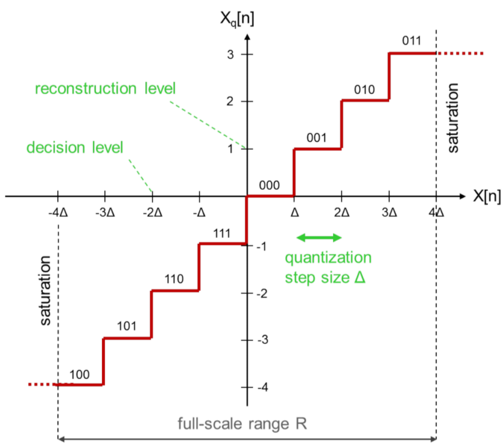


Quantization of Signals

Uniform Quantization

When quantizing sample values with W bits, the dynamic range R of the sampled signal $x[n]$ can be divided into 2^W intervals of equal width. Thus, the width of one **quantization step** is given by

$$\Delta = \frac{R}{2^W}$$

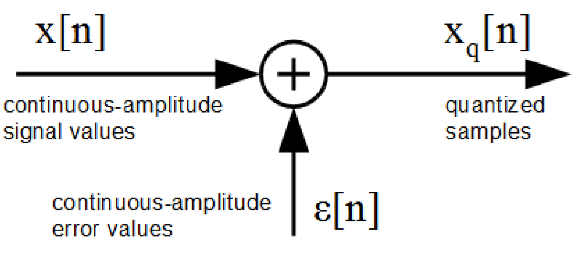
Mid-Tread	Mid-Rise
Decision levels at $\pm 0.5\Delta, \pm 1.5\Delta, \pm 2.5\Delta, \ldots$	Decision levels at $0, \pm\Delta, \pm 2\Delta, \ldots$
	
$-\frac{\Delta}{2} < \epsilon \leq \frac{\Delta}{2}$	$-\Delta < \epsilon \leq 0$

Furthermore **Clipping** occurs if the signal values of $x[n]$ are outside of the full-scale range R.

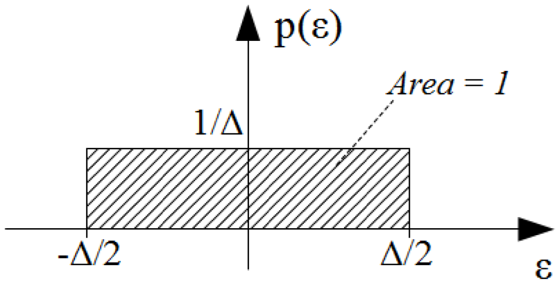
Quantization noise

The quantization error ϵ manifests itself as noise superimposed to the quantized signal

$$\epsilon[n] = x_q[n] - x[n]$$



(a)



(b)

Figure 5: Model of quantization noise (Figure 5a) and probability density function (Figure 5b)

The power of the quantization noise signal is

$$P_\epsilon = \frac{1}{\Delta} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \epsilon^2 d\epsilon = \frac{\Delta^2}{12}$$

The signal-to-noise ratio expressed in dB yields

$$SNR_{dB} = 6 \cdot W + 10 \cdot \log_{10}\left(\frac{12 \cdot P_x}{R^2}\right)$$

For Harmonic- & Full-Scale-Signals

$$SNR_{dB} = 6 \cdot W + 1.76 \approx 6 \cdot W$$

For every additional Bit, the SNR can be sixfold in $[dB]$.

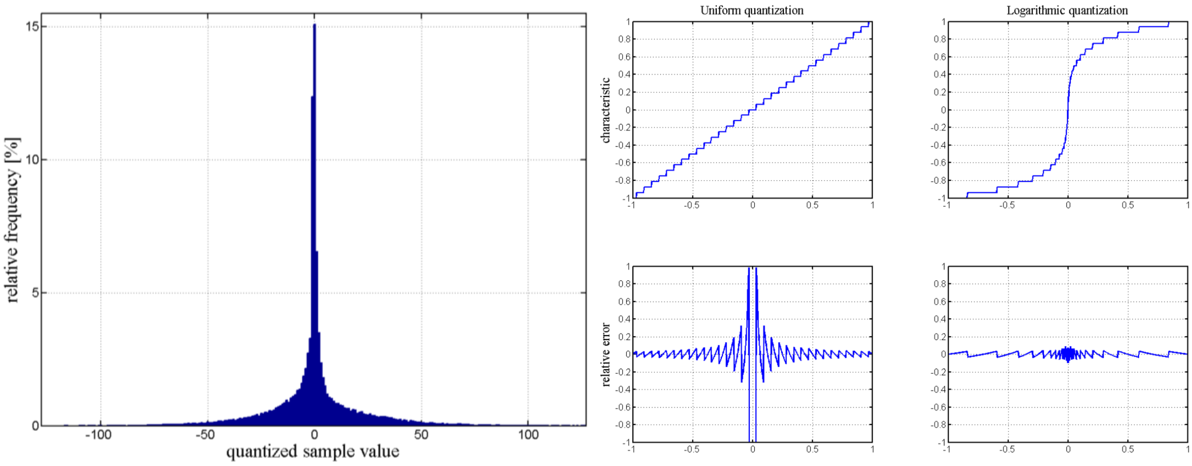
Logarithmic Quantization

One way to increase the SNR associated with quantization, is to adapt the quantizer characteristics to the statistical properties of the signal being quantized. One kind of signal with these properties are voice signals were very small amplitude values are orders of magnitude more likely than large amplitude values (Figure 6a).

There are several standards for implementing Logarithmic Quantization. One such standard is the μ-law algorithm

$$f_\mu(x) = \operatorname{sgn}(x) \cdot \frac{\ln(1 + \mu \cdot |x|)}{\ln(1 + \mu)} \qquad -1 \leq x \leq 1$$

With this applied the relative error can be significantly improved (Figure 6b).



(a) (b)

Figure 6: Comparison of uniform and logarithmic quantization

Digital Signals in the Frequency Domain

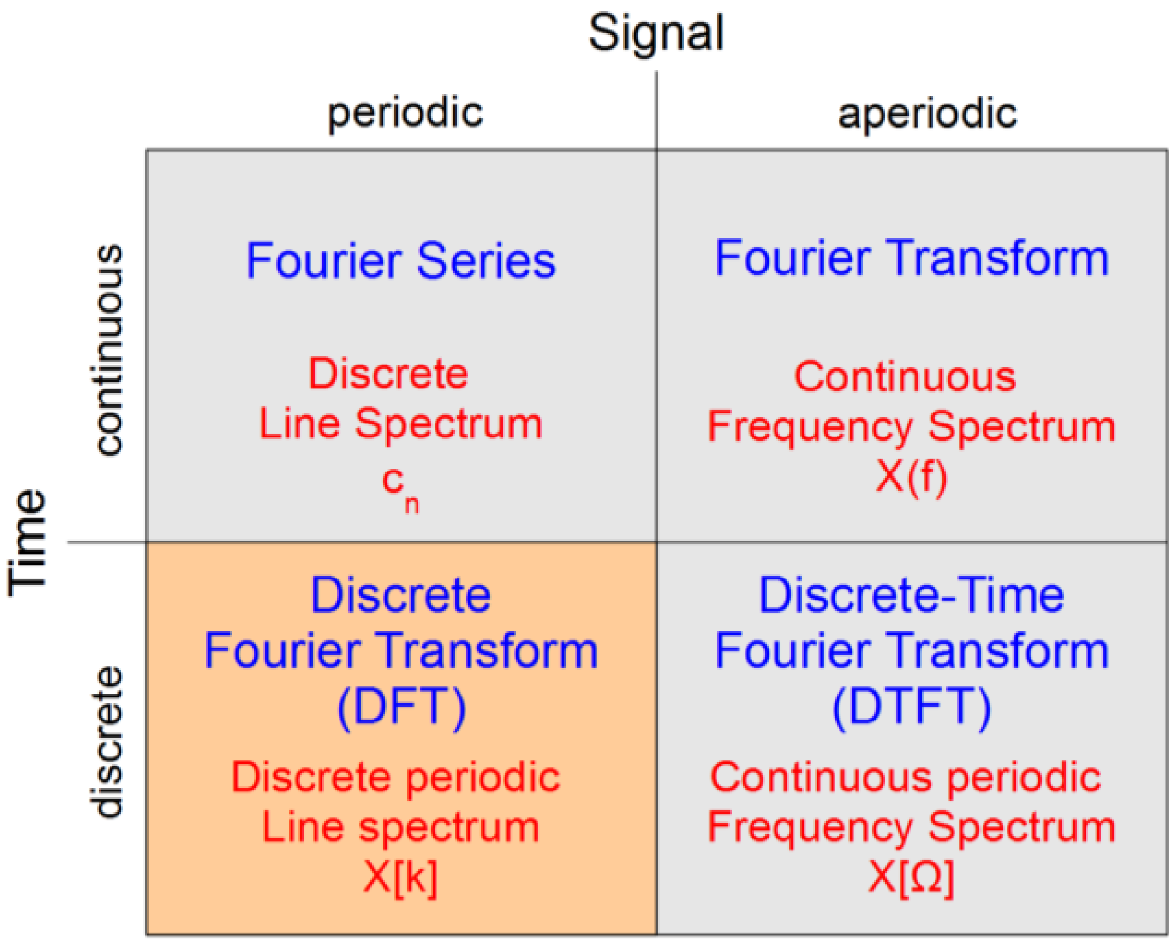


Figure 7: Comparison of Fourier methods

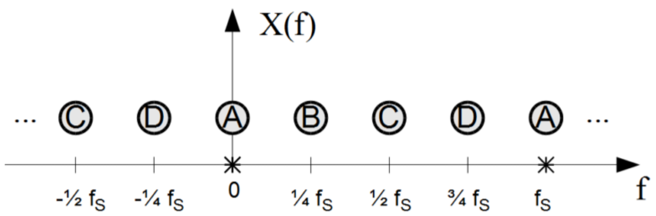
Fourier in Discrete Time

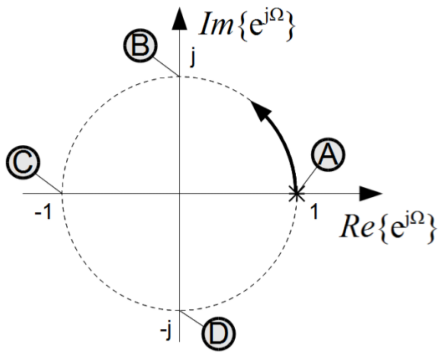
Discrete Time

Substituteing the continuous time t and integration with the discrete sample points nTs and summation, we get the frequency spectrum of the sampled signal $x_S(t)$. Thourgh the *normalized angular frequency*, we obtain the **Discrete-Time Fourier Transform (DTFT)**.

$$\underbrace{X_S(f) = \sum_{n=-\infty}^{\infty} x(nT_S)e^{-j2\pi f nT_S}}_{\text{Continuous Fourier Transform}} \overset{\Omega=2\pi f T_S = 2\pi \frac{f}{f_S}}{=} \underbrace{X(\Omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n}}_{\text{Discrete Fourier Transform}}$$

The **DTFT** produces a 2π-periodic, continuous spectrum. This is due to the mapping of the linear frequency axis onto the unit circle at the sampling rate of f_S.





Finite Measurement Interval

The lowest frequency we can capture in the measurement interval T, with a finite amount of sample points N is

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

> **! Resolvable frequencies**
>
> From the equation we learn that we can resolve lower frequencies if we increase N, on the other hand the highest frequency we can cover is f_S. The following generally applies
>
> $$f_k = k \cdot \frac{f_S}{N}$$

With this in mind we let the discrete-time-index n only run from 0 to $N-1$. We replace the $\frac{f}{f_S}$ in Ω with $\frac{k}{N}$ and get the **Discrete Fourier Transform (DFT)** where k is the discrete frequency index

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi n \frac{k}{N}} \qquad k = 0, 1, 2, \dots, N-1$$

> **i Observation**
>
> We get exactly as many *Fourier coefficients* $X[k]$ back, as we provide *samples* $x[n]$.
> The DFT produces a discrete and periodic line spectrum. The frequency resolution is $\frac{f_S}{N}$, i.e. we get the spectral values at the frequency points

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

Inverse Discrete Fourier Tranfsorm (IDFT)

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j2\pi n\frac{k}{N}} \qquad n = 0, 1, 2, \ldots, N-1$$

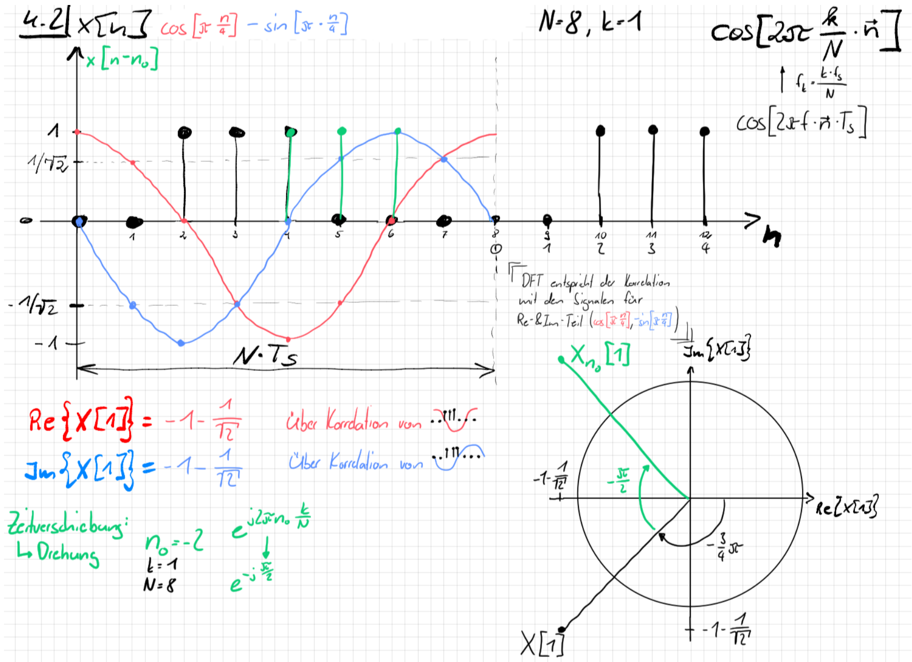


Figure 8: A intuitive approach to the DFT, also check chapter 4.2 of the main script by Wassner

Porperties of the DFT

Table 6: Periodicity

The *DFT* is f_S-periodic	The *IDFT* is periodic with $T = NT_S$
$X[k] = X[k+N]$	$x[n] = x[n+N]$

Table 7: Symmetry

DFT of a *real-valued signal* is symmetric around $k = \frac{N}{2}$
$X\left[\frac{N}{2}+m\right] = X^*\left[\frac{N}{2}+m\right]$

Table 8: Time/Frequency Shifting

Shifting by n_0 corresponds to a linear phase offset to all spectral values of the original time sequence	Multiplying the time sequence with a complex exponential, results in a constant frequency shift of the original spectrum
$x[n+n_0] \circ\!\!-\!\!\bullet\ e^{j2\pi n_0\frac{k}{N}} \cdot X[k]$	$e^{j2\pi k_0\frac{n}{N}} \cdot x[n] \circ\!\!-\!\!\bullet\ X[k-k_0]$

Table 9: Modulation

A direct consequence of the frequency shifting property		
$\cos(2\pi k_0 \frac{n}{N} \cdot x[n]$	$\circ\!\!-\!\!\bullet$	$\frac{1}{2}(X[k+k_0] + X[k-k_0])$

Table 10: Parseval Theorem

Between the signal samples $x[n]$ and the Fourier coefficients $X[k]$ following relationship exists
$\frac{1}{N}\sum_{n=0}^{N-1} x[n]^2 = \sum_{k=0}^{N-1} \left\lvert \frac{X[k]}{N} \right\rvert^2$

Table 11: Correspondence of Convolution and Multiplication

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

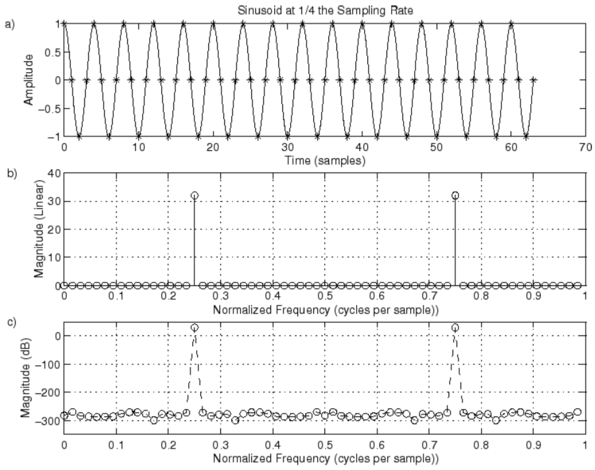
Range of Validity of the DFT

The Discrete Fourier Transform (DFT) accurately represents the spectrum of periodic signals within a finite measurement interval but requires an infinite interval for aperiodic signals to approach the correct result of the Discrete-Time Fourier Transform (DTFT). In the case of aperiodic signals, the DFT provides an approximation of the DTFT spectrum, and windowing functions are commonly employed to minimize the associated approximation errors in practical applications. In this case the DFT samples the DTFT at discrete points of the noramlized angular frequency Ω

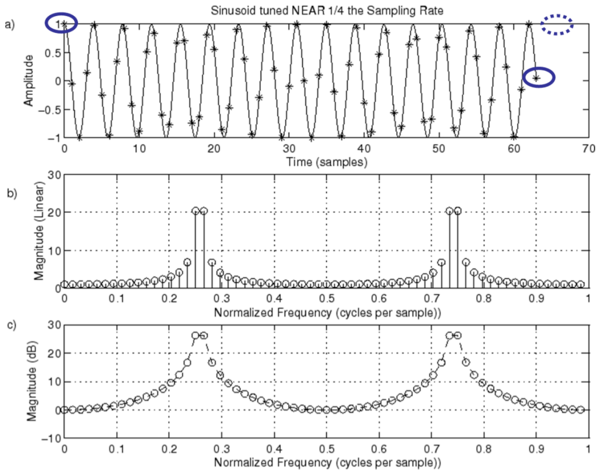
$$X[k] = X(\Omega)|_{\Omega = 2\pi\frac{k}{N}}$$

Practical Application Aspects of the DFT

Choice of Measurement Interval



(a)

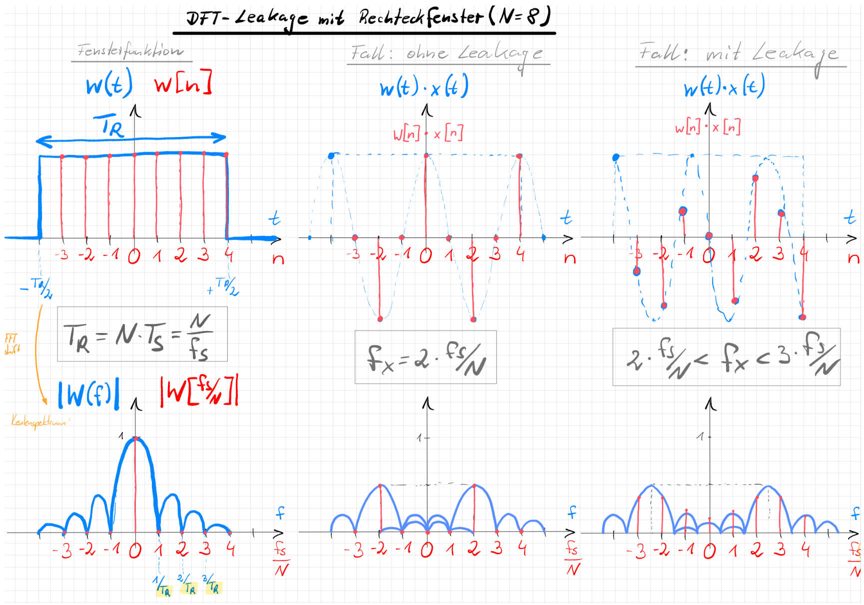


(b)

Figure 9: 9b shows the DFT of a cosine with correct measurement interval ($N = 64$). 9b shows the application of the DFT with incorrect measurement interval which yields jumps in the virtual periodic extension of the signal being analyzed and thus **leakage** in the frequency domain occures.

Leakage

The *leakage-effect* manifests itself in the main frequency "leaking" into several frequency indexes surrounding the two true frequency points (see Figure 9).



DFT and Zero-padding

To "increase" spectral resolution zero padding is added.

Zero-padding

Zero-padding **won't** increase spectral resolution, it just corresponds to a better interpolation between the N frequency points of the DFT spectrum.
To increase spectral resolution, additional samples of the signal $x[n]$ must be sampled.

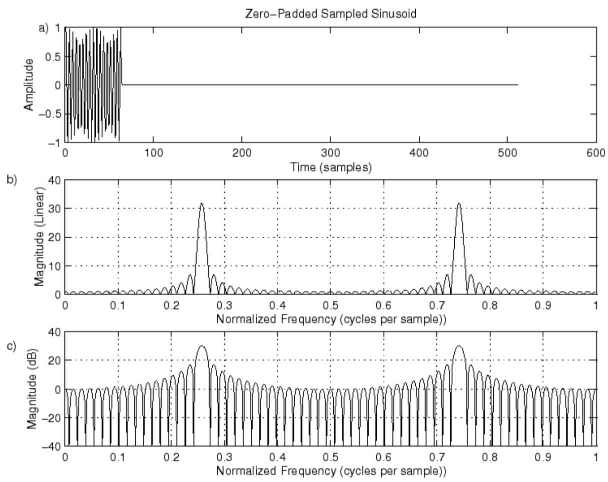
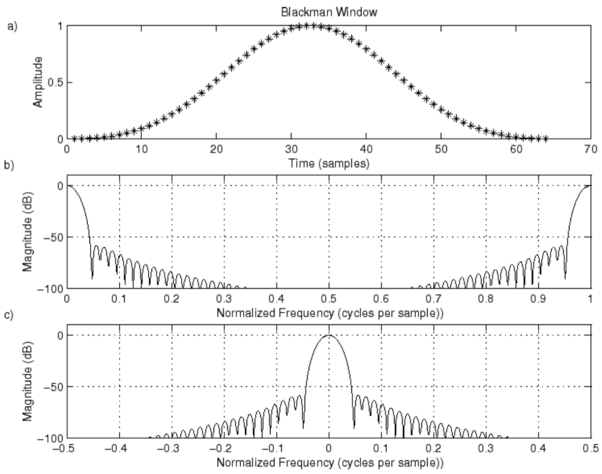


Figure 10: Zeropadded signal in time & frequency domain

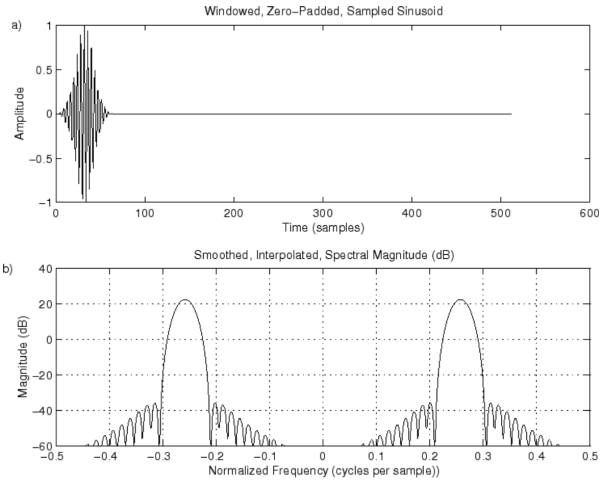
The finite measurement interval of the DFT corresponds to multiplying the signal $x[n]$ with a $rect$-Window, which corresponds to a a convolution of $X(\Omega)$ with the $\frac{\sin(x)}{x}$-function, thus introducing the form of lobes. This lobe-structure is independent of the form of the signal waveform $x[n]$.

DFT and Windowing

To overcome the problem of lobe-introduction, we apply **windowing**. A window can be any function, that is applied to the signal in the measurement interval.



(a)



(b)

Figure 11: 11a shows the Blackman Window for $N = 64$ in time and frequency. The Signal from Figure 10 with a Blackman window applied before zero-padding shows much smaller lobes in frequency domain (Figure 11b)

> Effects introduced by Windowing
>
> - Comparing the original signal Figure 10 to the windowed in Figure 11b that the first side lobe is significantly smaller ($12dB \rightarrow -60dB$), so the *leakage is minimized*
> - The width of the main lobe on the other hand has become wider ($0.03 \rightarrow 0.1$ cycles per sample), which equals to the *reduction of the frequency resolution*

Windowing functions Different windowing functions will have different influence on the time and frequency domains of signals and should be chosen there for.

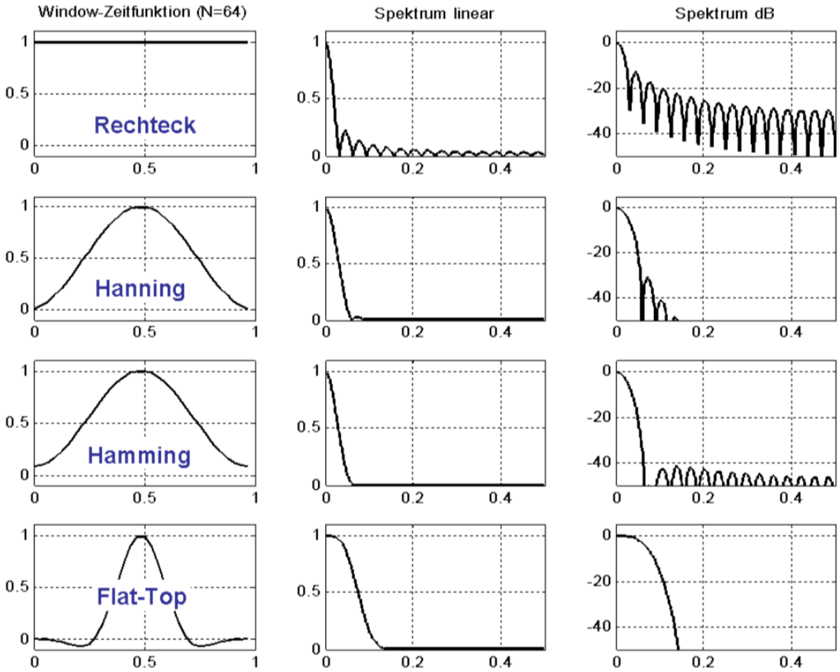


Figure 12: Comparison of windowing functions in the time and frequency domain

Short-Time DFT

When assessing the development of the frequency spectrum over time a **continuous** evaluation of the frequency spectrum of short signal sections is required **(short-time DFT)**.

> **!** short-time DFT
>
> Since such signal sections are unlikely to fit a signal period perfectly, windowing is applied, which results in *less leakage* but a *convolution of the signal with the window spectrum.*
> When setting the length N of the window, one compromises between:
> - high spectral resolution *(N large)*
> - high time resolution *(N small)*

*(*For both **good spectral and time resolution** we have to overlap the DFT windows, see Figure 13. In the limit, we could overlap two consecutive DFT windows by $N - 1$ sample values *if computationally feasible.*

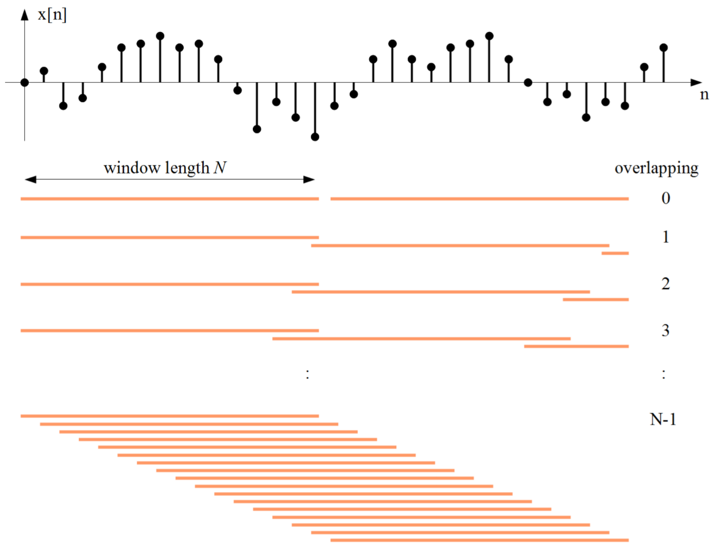


Figure 13: **Overlapping of windows in the short-time DFT**

Fast Fourier Transform FFT

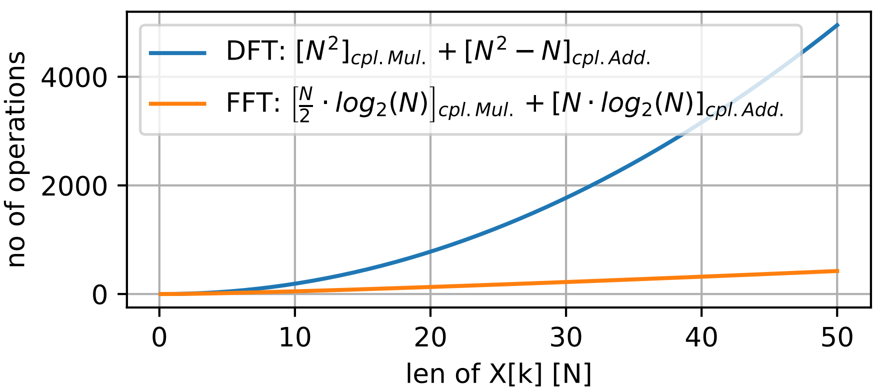
Computational effort DFT vs. FFT

To calculate a spectrum through *DFT* ($x * y$)

$$[N^2]_{cpl.Mul.} + [N^2 - N]_{cpl.Add.}$$

To calculate a spectrum through *FFT* ($ifft(fft(x) \cdot fft(y)$)

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



That gives us a speedup factor (assumption: real addition= real multiplication) of

$$\frac{8N-2}{5 \cdot log_2(N)} \approx 1.5\frac{N}{log_2(N)}$$

Twiddle Factors

The **twiddle factors** are the building part of every complex harmonic sequence in a N-point DFT and are defined as

$$W_N = e^{-j\frac{2\pi}{N}}$$

Periodicity: W_N^k can evaluate N different numpery only $\rightarrow$ N-periodic:

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

Symmetry: Apart from the sign, W_N^k only holds $\frac{N}{2}$ numbers:

$$W_N^{k+\frac{N}{2}} = -W_N^k$$

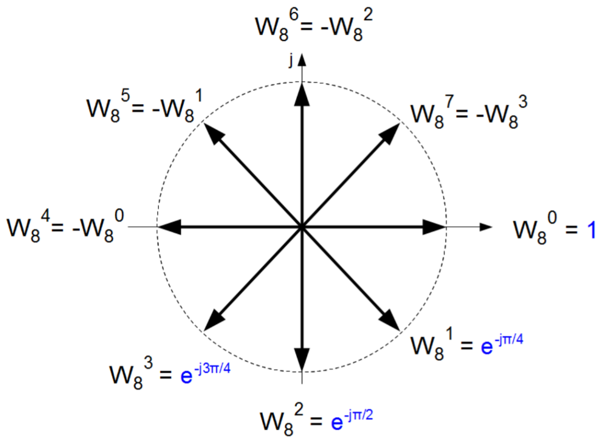


Figure 14: Example for $N = 8$

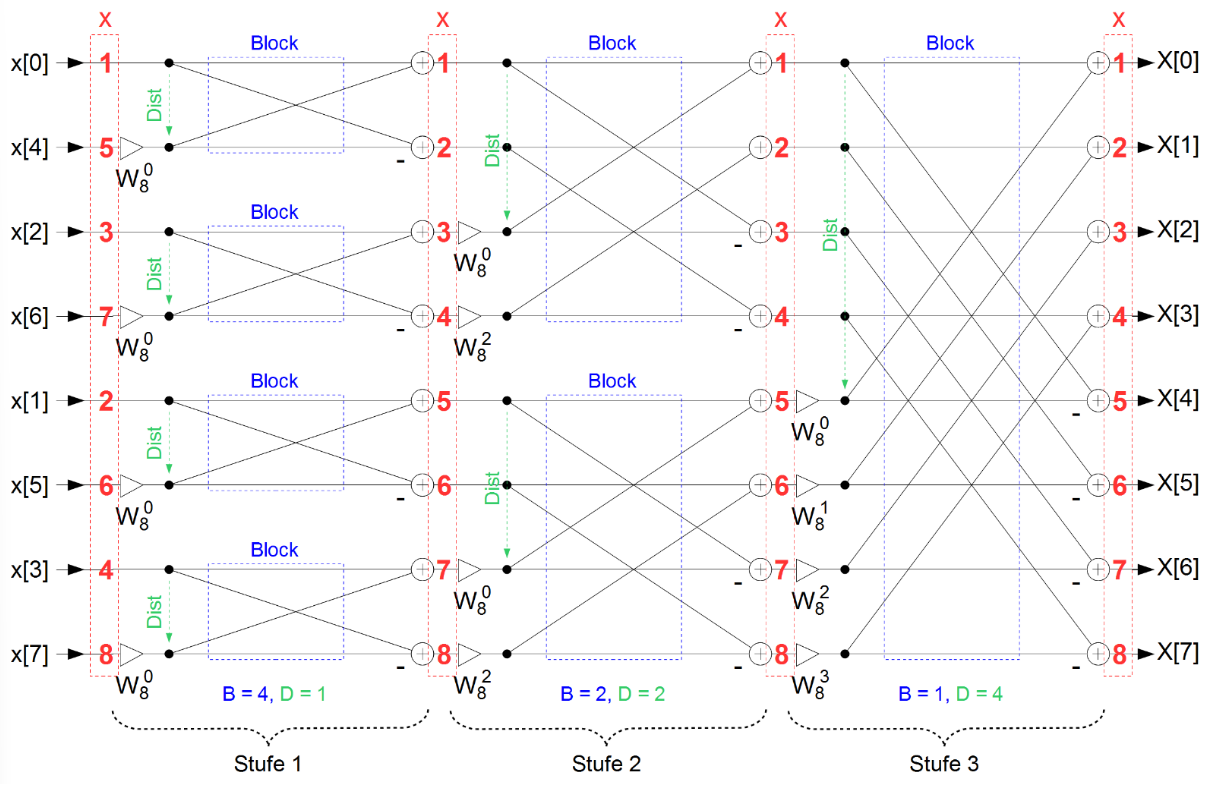
Butterfly operation

Using the twiddle factors we can process a FFT much faster. For this we obtain the **Radix-2 decimation-in-time.**

$$\begin{aligned}
X[k] &= \sum_{n=0}^{N-1} x[n] e^{-j2\pi n \frac{k}{N}}\,, \qquad k = 0, 1, \ldots, N-1 \\
&= \sum_{n=0}^{N-1} x[n] W_N^{nk} \\
&= \sum_{m=0}^{N/2-1} x[2m] W_N^{2mk} + \sum_{m=0}^{N/2-1} x[2m+1] W_N^{(2m+1)k} \qquad \Big| W_N^2 = W_{N/2} \\
&= \underbrace{\sum_{n=0}^{N/2-1} x_1[n] W_{N/2}^{nk}}_{X_1[\tilde{k}]} + W_N^k \cdot \underbrace{\sum_{n=0}^{N/2-1} x_2[n] W_{N/2}^{nk}}_{X_2[\tilde{k}]} \qquad \Big| \tilde{k} = k \mod N/2
\end{aligned}$$

$x_1[n]$ and $x_2[n]$ contain samples of $x[n]$ with even and odd time index, respectively.

Through this we get a calculation architecture that is built by *2-point-FFTs*



Allgemein kann das Vorgehen so beschrieben werden:

- Werte in *bit-reversed*-Reihenfolge in Verarbeitung geben (Stufe 1), anschliessend linear im Speicher halten *(Endergebniss stimmt ohne weitere umsortierung)*
- Aus verarbeitungsstufen $s = 1 \dots \log_2(N)$ die Anzahl Blöcke $B = \frac{N}{2^s}$ und Distanz $D = 2^{s-1}$ berechnen
- Inidizes $i_1 = (b-1) \cdot (2^s) + d$ und $i_2 = i_1 + D$ berechnen
- Index für Drehfaktor $i_W = (d-1) \cdot B + 1$ berechen
- Butterfly auf berechnete indizes anwenden

```
W = exp(-1j*2*pi*[0:N/2-1]/N);  % twiddle factors (only N/2 because of symmetry)

x = bitrevorder(x); % bit-reversed order to allow in-place computation
for s = 1:log2(N)   % for each stage to do ...
    B = N/(2^s);        % get # of blocks in current stage
    D = 2^(s-1);        % get # of butterflies per block in current stage
    for b = 1:B         % for each block in current stage to do ...
        for d = 1:D         % for each buttergly in current block to do ...
            i1 = (b-1)*(2^s)+d;     % get 1st operand idx
            i2 = i1+D;              % get 2nd operand idx
            iW = (d-1)*B+1;         % get twiddle factor idx
            [x(i1) x(i2)] = butfly(x(i1),x(i2),W(iW));      % compute butterfly
 ↪  in-place
        end
    end
end
```

Efficient FFT implementation

The entire FFT can be performed **in-place**, this is because the memory place of the input-pair can be filled after a butterfly execution with its result. Thus only $2N$ memory locations (N complex values) plus $N/2$ for the twiddle factors are needed. Additionally by the application of **bit-reversed addressing** the algorithm can work much faster and the result is correct without reordering.

linear order		bit-reversed order	
decimal	binary	binary	decimal
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

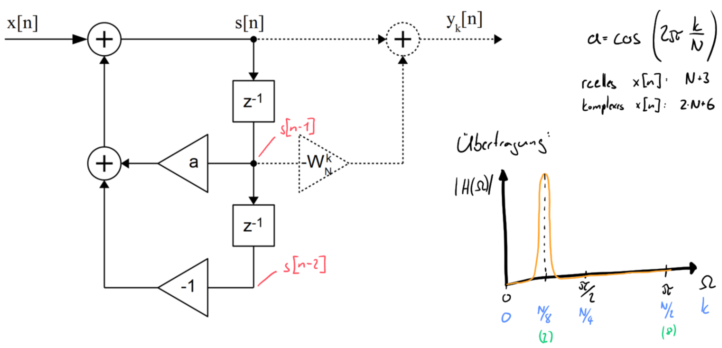
Figure 15: Bit-reversed order for $N = 8$. Indeces with symmetric binary representation remain unchanged.

Goertzel Algorithm

If only individual $X[k]$ out of all N spectral components are needed, the *Goertzel algorithm* can be applied. This is also a linear filtering technique. The IIR-system of difference equations is

$$s[n] = x[n] + a \cdot s[n-1] - s[n-2], \qquad a = 2\cos\left(2\pi\frac{k}{N}\right) \tag{0.2}$$

$$y_k[n] = s[n] - W_N^k \cdot s[n-1] \tag{0.3}$$



Parseval theorem

Goertzel gives us the DFT spectral component at frequency

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

Through Parsevals theorem we get the power content P_k in a real-valued signal $x[n]$ around f_k

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

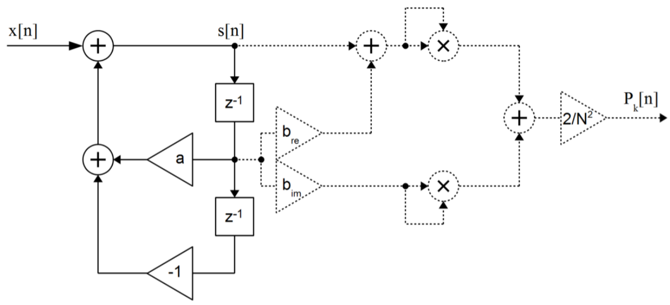


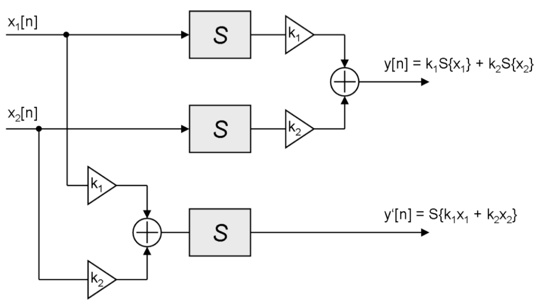
Figure 16: With $b_{re} = \Re\{-W_N^k\}$, $b_{im} = \Im\{-W_N^k\}$

Digital LTI Systems

Definition

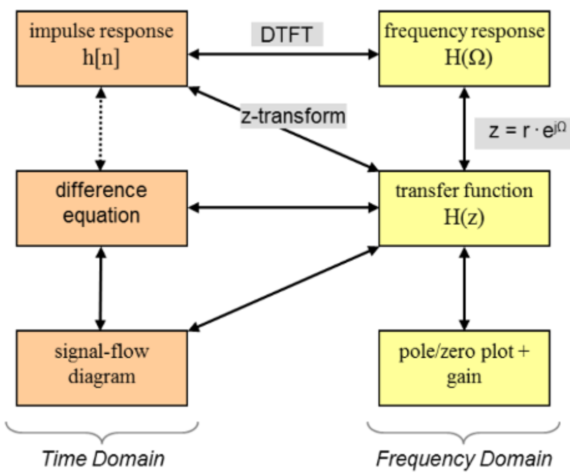
Linearity:

- Multiplication of a signal with a constant $y[n] = a \cdot x[n]$
- Addition of two signals $y[n] = x_1[n] + x_2[n]$



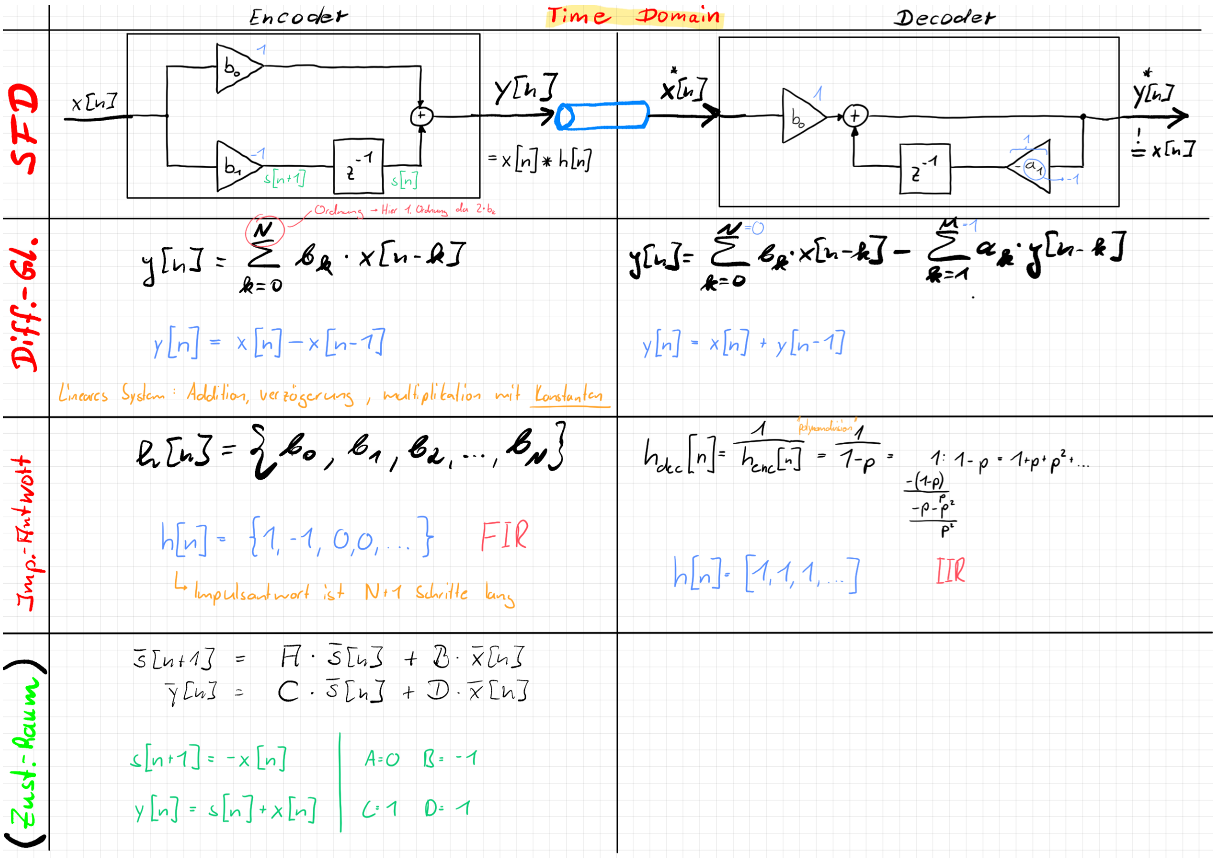
Time-Invariance: Time delay of a signal by $d = k \cdot T_S$, $x[n] \to y[n] = x[n-d] \to y[n-d]$

Description of LTI systems



	impulse response	system identification, measurement
Time Domain	**difference equation**	system implementation (algorithm)
	signal-flow diagram	system implementation (architecture)
	transfer function	coupling analysis and implementation
Frequency Domain	**pole/zero plot**	intuitive analysis and design
	frequency response	system identification, analysis and design

System Descriptions in the Time Domain



Impulse response

```
% plot impulse response of a system
impz([b0 b1 ...],[a0 a1 ...], n) % a, b: transfer function coeff, n: no. of samples
```

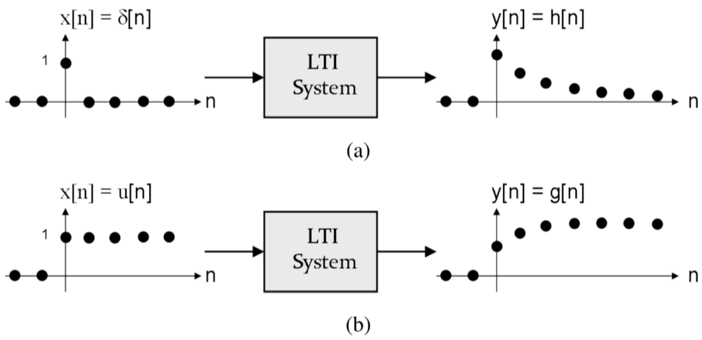


Figure 17: Impulse response (a) and step response (b)

A *LTI*-System is reacts to each weighted unit impulses $x_S(i) \cdot \delta[n-i]$ with a weighted impuls response $x_S(i) \cdot h[n-i]$. Through superposition we obtain the systems reaction

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

Dies entspricht der linearen Faltung.

Difference Equation

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

$max(N.M)$ is the **order** of the system. A system $M \geq 1$ is **recursive**. Directly convertible into *Signal-Flow Diagram*.

Signal-Flow Diagram

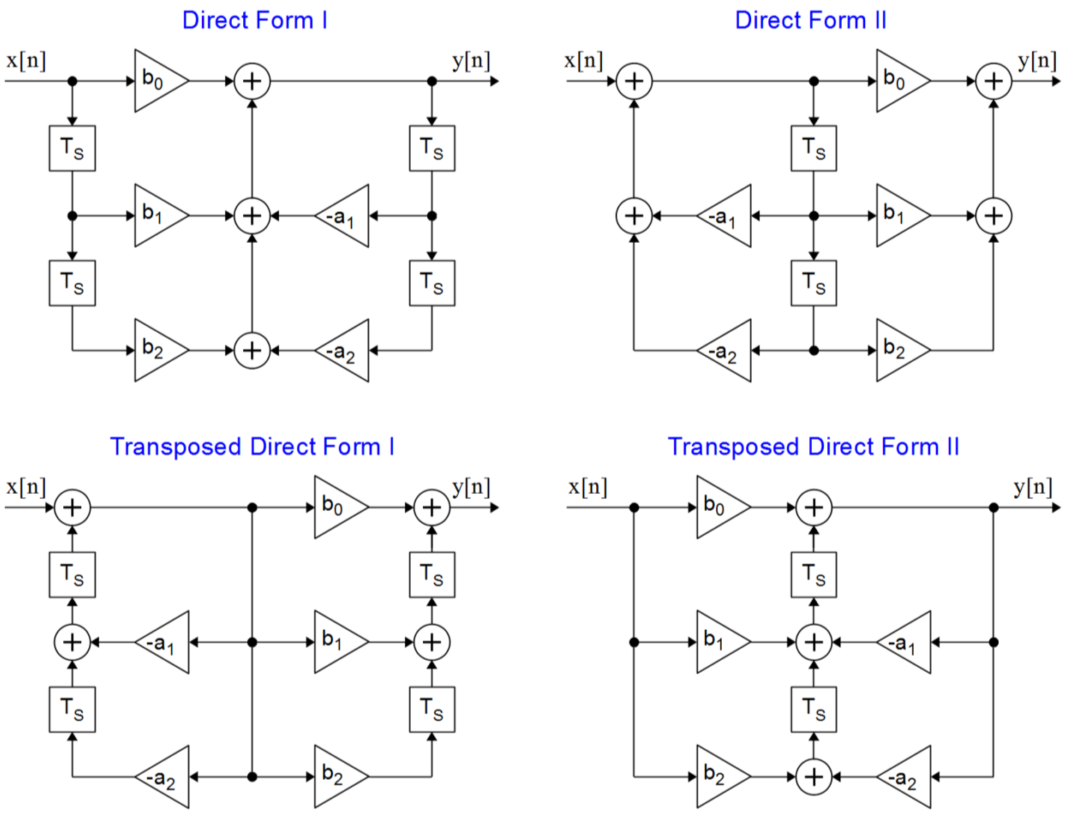
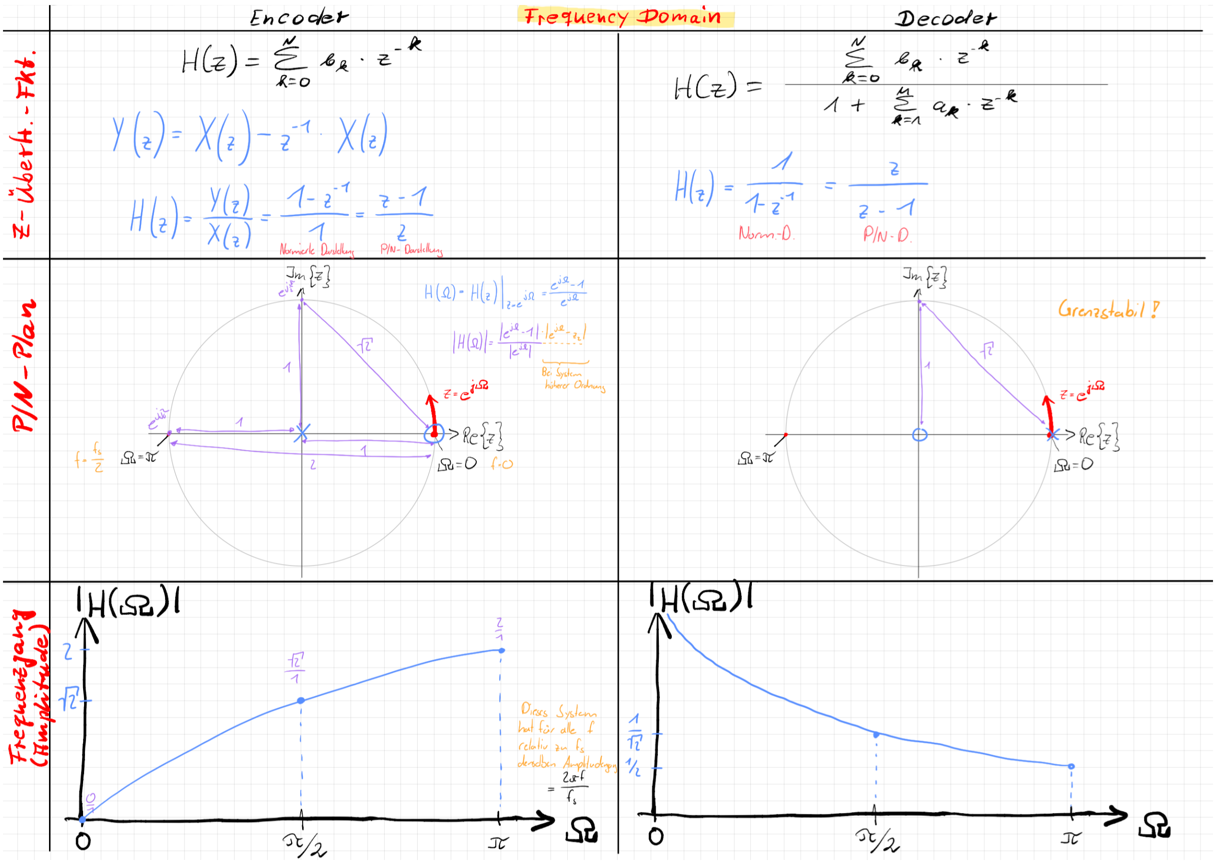


Figure 18: Normalized forms of signal-flow diagrams for digital LTI-systems

System Description in the Frequency Domain



Transfer Function

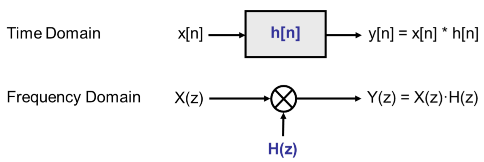
$$y[n]=\sum_{k=0}^{N} b_k x[n-k]-\sum_{k=1}^{M} a_k y[n-k] \circ\!\!-\!\!\bullet Y(z)=\sum_{k=0}^{N} b_k z^{-k} X(z)-\sum_{k=1}^{M} a_k z^{-k} Y(z)$$

by reordering we obtain the **z-transfer-function** *(normalized if* $a_0=0$*)*

$$H(z)=\frac{Y(z)}{X(z)}=\frac{b_0+b_1z^{-1}+b_2z^{-2}+\cdots+b_Nz^{-N}}{1+a_1z^{-1}+a_2z^{-2}+\cdots+a_Mz^{-M}}$$

The transfer function $H(z)$ of a digital LTI system is the z-transform of its impulse response $h[n]$, thus we get

$$y[n]=x[n]*h[n] \circ\!\!-\!\!\bullet Y(z)=X(z)\cdot H(z)$$



Pole/Zero-Plot

```
% plot the zero-poles-diagram
zplane(z,p) % with zeros and poles as in z=[z0 z1 ...], p=[p0 p1 ...]
zplane(b,a) % with transferfunction coeffs
```

To analyze the system behavior we use the **pole/zero-form**

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

Tip

- $N > M$: there are $N-M$ additional poles at $z=0$. If $b_0 \neq 0$, the inverse is true!
- $M > N$: there are $M-N$ additional zeros at $z=0$. In this case $K_0 = b_0$ is true

! Stability

A causal digital LTI-system is stable if all poles p_i are located within the unit circle.

Frequency Response

```
% Plot Bode-Diagram of a system
freqz(b,a) % a,b: transferfunction coeffs
```

Systembehavior as a function of its input signal *(Bode-Plot)*. **Measure**: Frequency sweep over all frequencies, **Indirect Measurement**: obtain impulse response and transform into frequency response $h[n] \circ\!\!-\!\!\bullet H(\Omega)$. The frequency response turns out to be f_S or 2π periodic and complex valued. We represent the frequency response in polar coordinates:

$$H(\Omega) = |H(\Omega)| \cdot e^{j\varphi(H(\Omega))}$$

Where the **amplitude response** is mostly written in dB

$$|H(\Omega)|_{dB} = 20 \cdot \log_{10}(|H(\Omega)|$$

For a *distortionfree* transmission a **linear phase response** $\varphi(H(\Omega)) = -K \cdot \Omega$ is desirable, thus will make all frequencies undergo the sampe time delay. This delay is called the **group delay** and is defined as

$$\tau_g = -\frac{d\varphi(H(\Omega))}{d\Omega} \cdot T_S = \frac{K}{2\pi}$$

A LTI-System will react to a sinusoidal input with a sinusoidal output of the same frequency

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

We can obtain the *amplitude* through (Attention: Add in dB!)

$$|Y(\Omega)| = |X(\Omega)| \cdot |H(\Omega)|$$

and the *phase*

$$\varphi(Y(\Omega)) = \varphi(X(\Omega)) + \varphi(H(\Omega))$$

Relation between frequency response and transfer function

In general: $z = re^{j\Omega}$, which gives us

$$H(\Omega) = H(z)|_{z=e^{j\Omega}}$$

We can obtain the bode plot from the pole/zero-form where the **amplitude** is

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

and the **phase**

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

Design of Digital Filters

Fourier Analysis of analog Signals

DFT Inside

The z-Transform