Course summary

Zhuoqi Cheng
zch@mmmi.sdu.dk
SDU Robotics

Topics which have been covered in this course

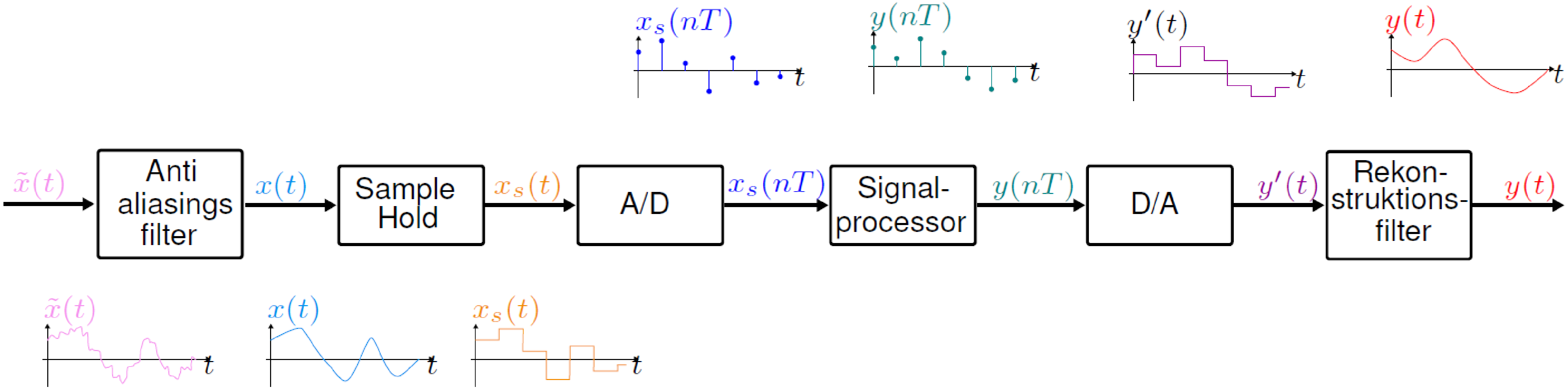
Domains:
→ Time & frequency conversion (DFT, FFT)
→ S domain & Z domain (continuous time signal v.s. discrete time signal, analog signal v.s. digital signal)

Signal & system:
→ Sampling and reconstruction
→ System analysis

Filter design:
→ Analog filter
→ Digital filter (FIR and IIR)

Realization & Implementation:
→ Realization
→ Quantization and multi-rate

Overview: signal processing workflow



Time & frequency conversion

Time-frequency domain

→ DFT & inverse DFT

$x(n)$ is a sequence sampled with interval T
The N-points DFT of $x(n)$ is given as

$$X(m) = \sum_{n=0}^{N-1} x(n) W_N^{mn}$$

for m = 0, 1, …, N-1
Where $W_N = e^{-j2\pi/N}$

The sequence $x(n)$ can be found from the spectrum $X(m)$ as

$$x(n) = \frac{1}{N} \sum_{m=0}^{N-1} X(m) W_N^{-mn}$$

for n = 0, 1, …, N-1

Fast Fourier Transform

$\frac{N}{2}$ point DFT $\frac{N}{2}$ point DFT

$$Y(m) = Y_{even}(m) + W_N^m Y_{odd}(m)$$

m from 0 to N-1

The complexity for calculating DFT_N equals to 2 times $DFT_{N/2}$ + N times Multiplication + N times Addition

Calculating $DFT_{N/2}$ requires 2 times $DFT_{N/4}$ + $\frac{N}{2}$ times Multiplication + $\frac{N}{2}$ times Addition
Therefore, overall, we need to calculate
$N/2$ times $DFT_{N/(N/2)}$ + $N * (\log_2 N - 1)$ times Multiplication + $N * (\log_2 N - 1)$ times Addition

It is about
$N * (\log_2 N - 1)$ times Multiplication + $N * \log_2 N$ times Addition

Calculate a DFT by direct use of the formulas, it requires N^2 times computation
When using the FFT algorithm, only $N * \log_2(N)$ times computation.

Discrete time signal in s domain & z domain

Laplace transform of sequence $x(n)$

$$X_s(s) = \sum_{n=0}^{\infty} x(n)e^{-snT}$$

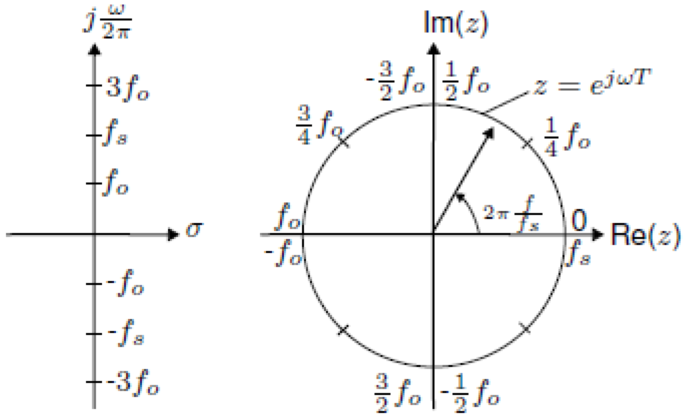
z transform of sequence $x(n)$

$$X(z) = \sum_{n=0}^{\infty} x(n)z^{-n}$$

$X_s(s) = X(z)$ when $\boldsymbol{z = e^{sT}}$

Where $s = \sigma + j\omega$

$$z = e^{sT} = e^{\sigma/f_s}\angle 2\pi\frac{f}{f_s}$$

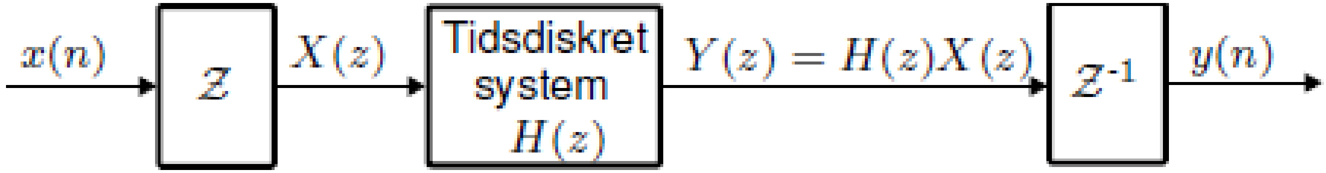


Transfer function (system) in z domain

Discrete-time systems can be described by a **transfer function** as

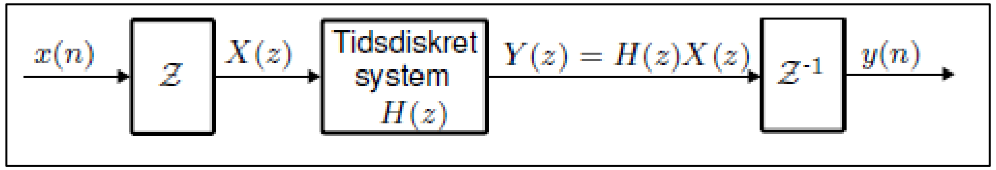
$$H(z) = \frac{Y(z)}{X(z)}$$

$H(z)$ is a transfer function, and $X(z)$, $Y(z)$ is the input and output sequence



The roots of $Y(z)$ are called **zeros**
The roots of $X(z)$ are called **poles**

Inverse z-transform



Inverse z-transformation is used to determine the output response $y(n)$ of a discrete-time system for a given input $x(n)$.

The processing of discrete system:

1. The input sequence $x(n)$ is z-transformed.
2. The system's transfer function $H(z)$ is set up with positive powers of z
3. The output response in z-domain is calculated $Y(z) = H(z)X(z)$
4. The output sequence $y(n)$ is calculated by inverse z-transform of $Y(z)$.

The processing of inverse z-transform:

1. Using partial fraction to set up expressions for $Y(z)$ as

$$\frac{\boldsymbol{Y(z)}}{\boldsymbol{z}} = \frac{T(z)}{zN(z)} = \frac{k_1}{z-p_1} + \frac{k_2}{z-p_2} + \cdots + \frac{k_N}{z-p_N}$$

where numerator coefficients k_i are calculated as

$$k_i = (z-p_i)\frac{Y(z)}{z}\bigg|_{z=p_i}$$

2. Write Y (z) z in fractional form and multiply by z

$$Y(z) = \frac{k_1 z}{z-p_1} + \frac{k_2 z}{z-p_2} + \cdots + \frac{k_N z}{z-p_N}$$

3. Inverse z-transform all the fractions. (from Table)

$$y(n) = k_1 \cdot p_1^n + k_2 \cdot p_2^n + \cdots k_N \cdot p_N^n$$

ZT4	a^n	$\frac{z}{z-a}$

SDU

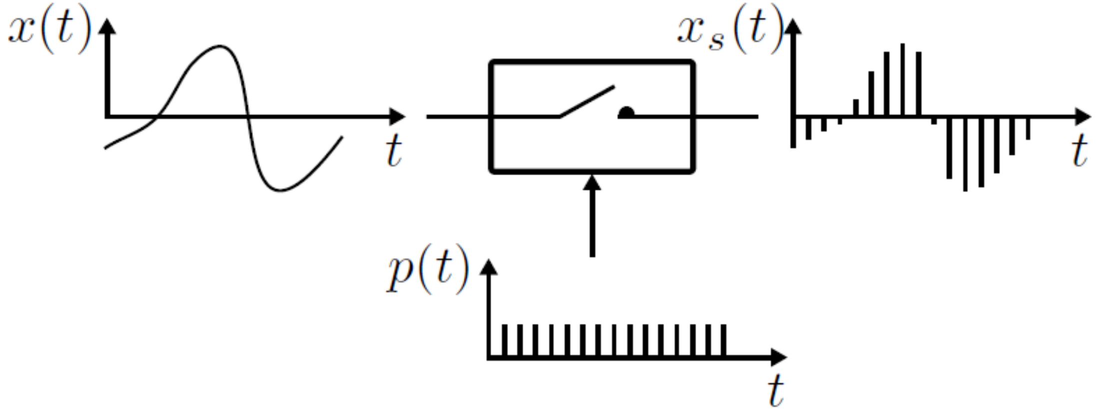
Common z-transform

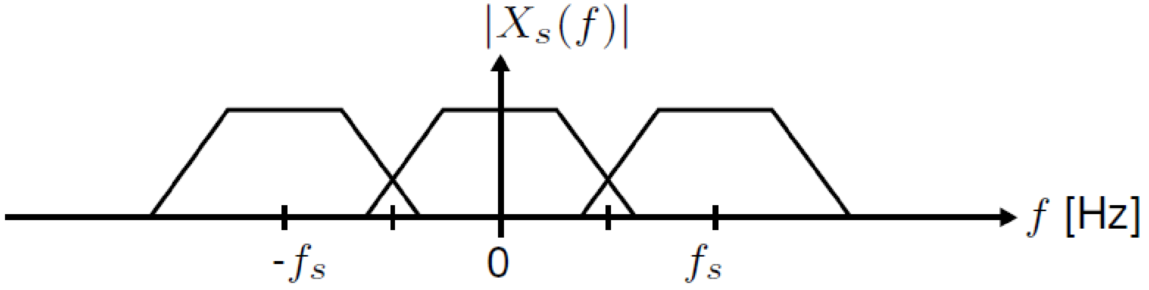
Par	$x(n)$	$X(z)$
ZT1	$\delta(n)$	1
ZT2	$u(n)$	$\frac{z}{z-1}$
ZT3	n	$\frac{z}{(z-1)^2}$
ZT4	a^n	$\frac{z}{z-a}$
ZT5	$e^{s_0 nT}$	$\frac{z}{z-e^{s_0 T}}$
ZT6	$\sin \omega_0 nT$	$\frac{(\sin \omega_0 T)z}{z^2-2(\cos \omega_0 T)z+1}$
ZT7	$\cos \omega_0 nT$	$\frac{z^2-(\cos \omega_0 T)z}{z^2-2(\cos \omega_0 T)z+1}$

Signal & system

Sampling

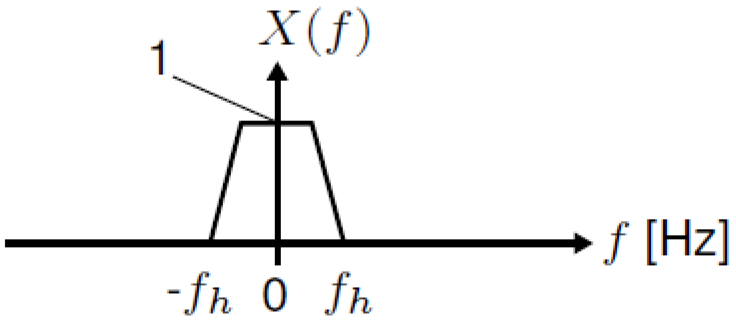
→ Sampling frequency f_s
→ Nyquist frequency $f_0 = f_s/2$
→ Aliasing



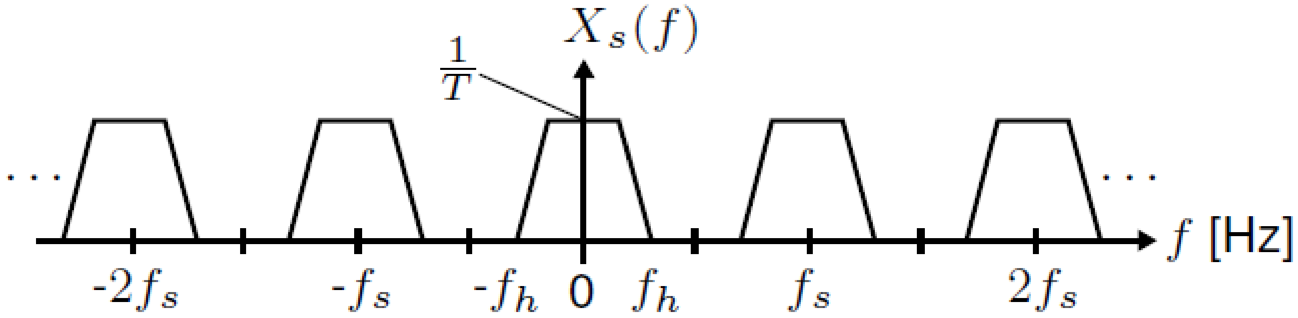


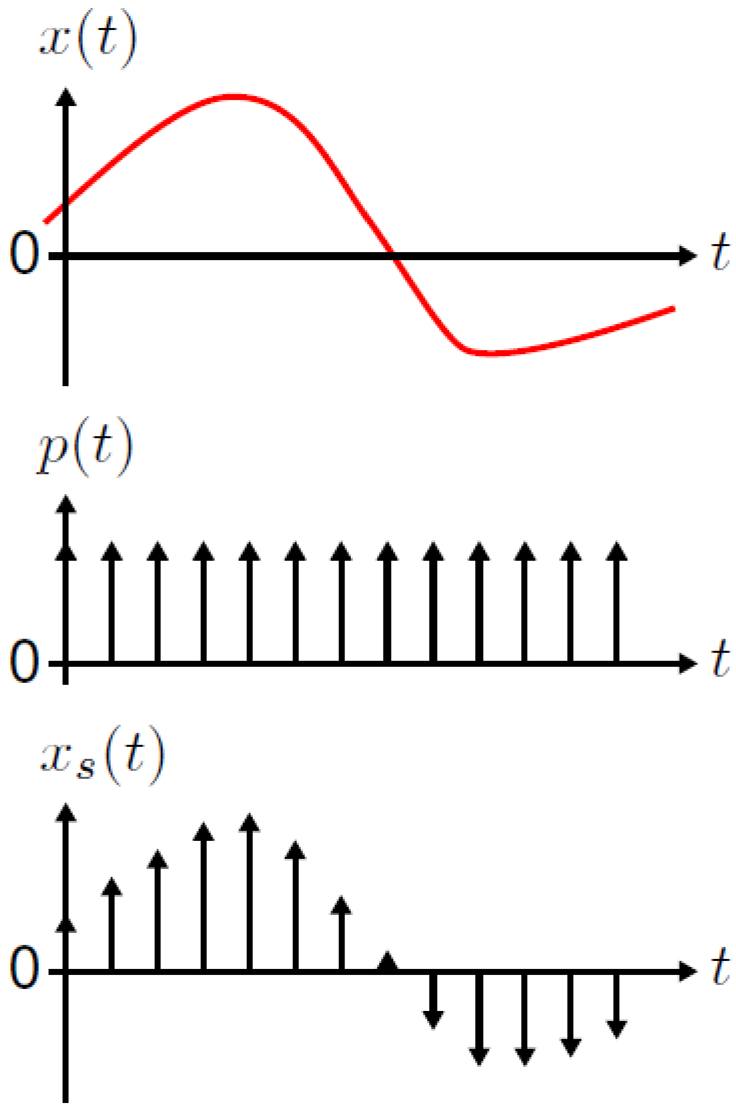
Impulse sampling

Example spectrum of continuous time signal $X(f)$



After pulse sampling, the amplitude spectrum becomes $X_s(f)$



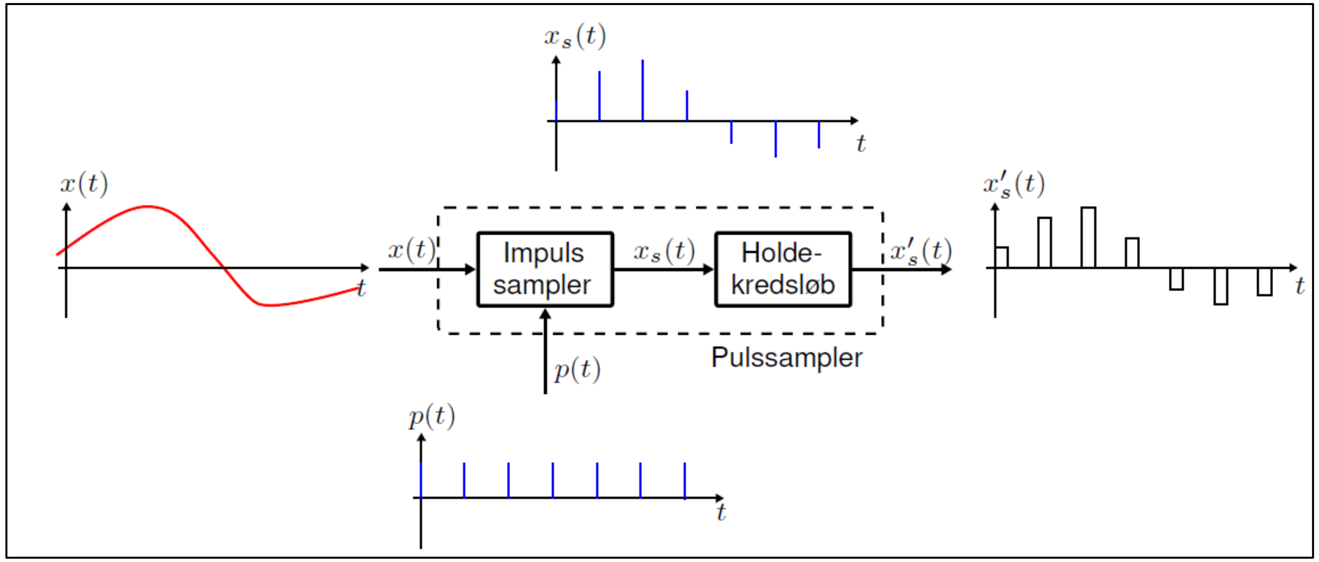


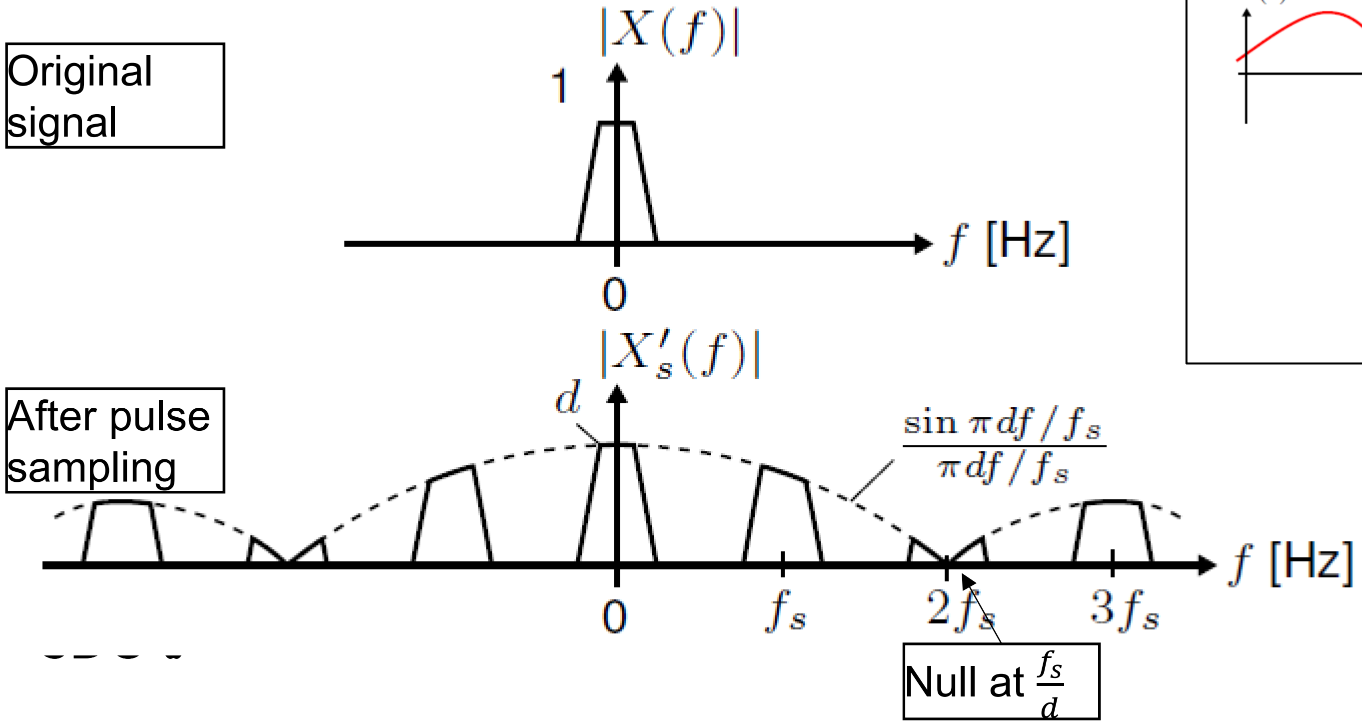
SDU

Pulse sampling

We use T to denote sampling interval [s], and τ to denote the pulse width [s].

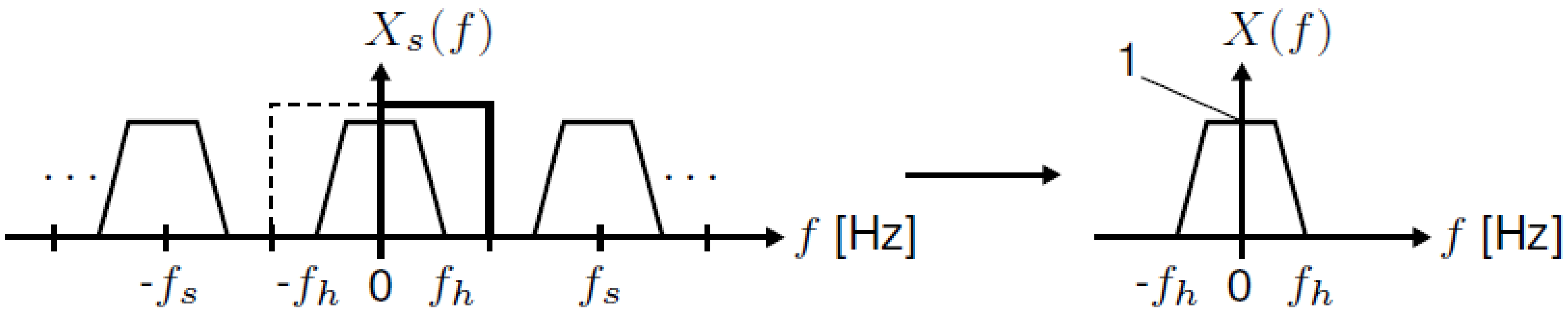
The duty factor of the pulse sampled signal $d = \frac{\tau}{T}$

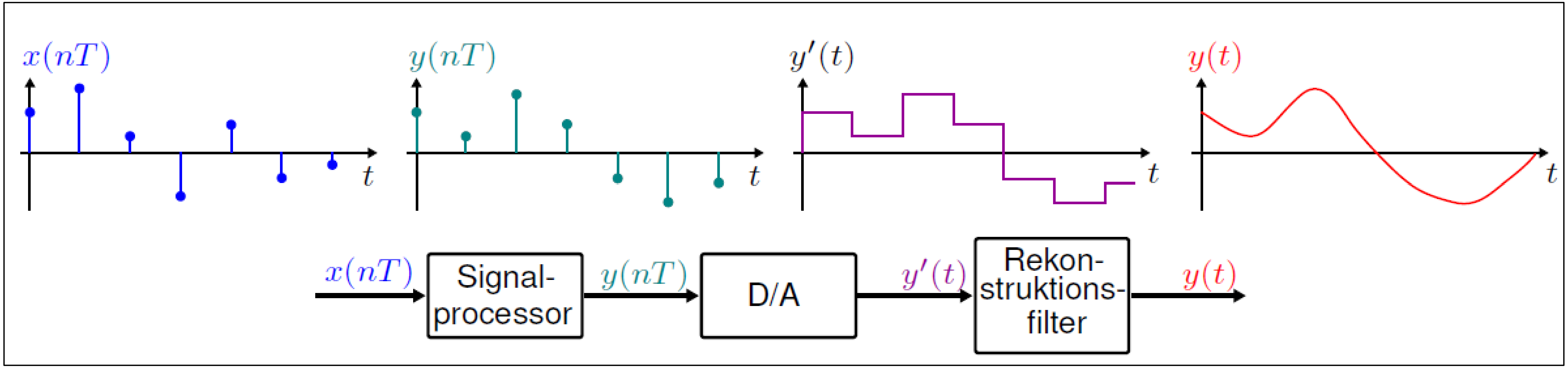




Reconstruction

The amplitude spectrum X(f) can be recreated from Xs(f) by **low-pass filtering**.

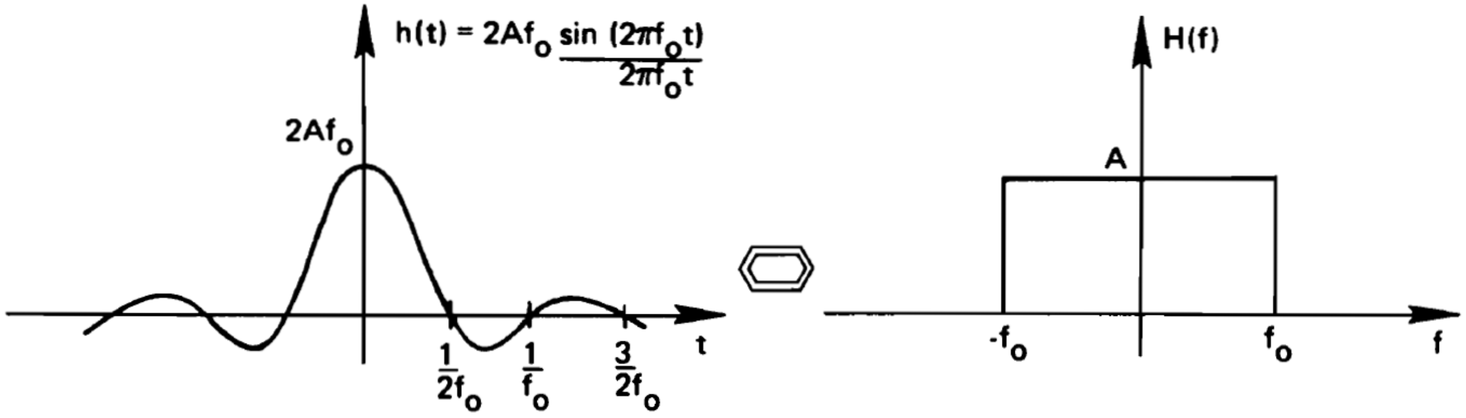




Reconstruction in the time domain

$$x_r(t) = h_r(t) * x_s(t) = \sum_n x(n)\ h_r(t - nT) = \cdots + x(-1)h_r(t+T) + x(0)h_r(t) + x(1)h_r(t-T) + \cdots$$

where $h_r(t)$ is called the reconstruction filter, normally is a $\mathrm{sinc}(\quad)$ function



System analysis

Transfer function $H(z)$ has poles $p_1, p_2, \dots, p_N$.

→ The system is **stable** if all poles lie within the unit circle, i.e.

$$|p_i| < 1 \quad for \quad i = 1, 2, \dots, N$$

→ The system is **marginally stable** if at least one pole (e.g. p_j) lies on the unit circle, while the other poles lie within the unit circle, i.e.

$$|p_i| \leq 1 \quad for \quad i = 1, 2, \dots, N$$

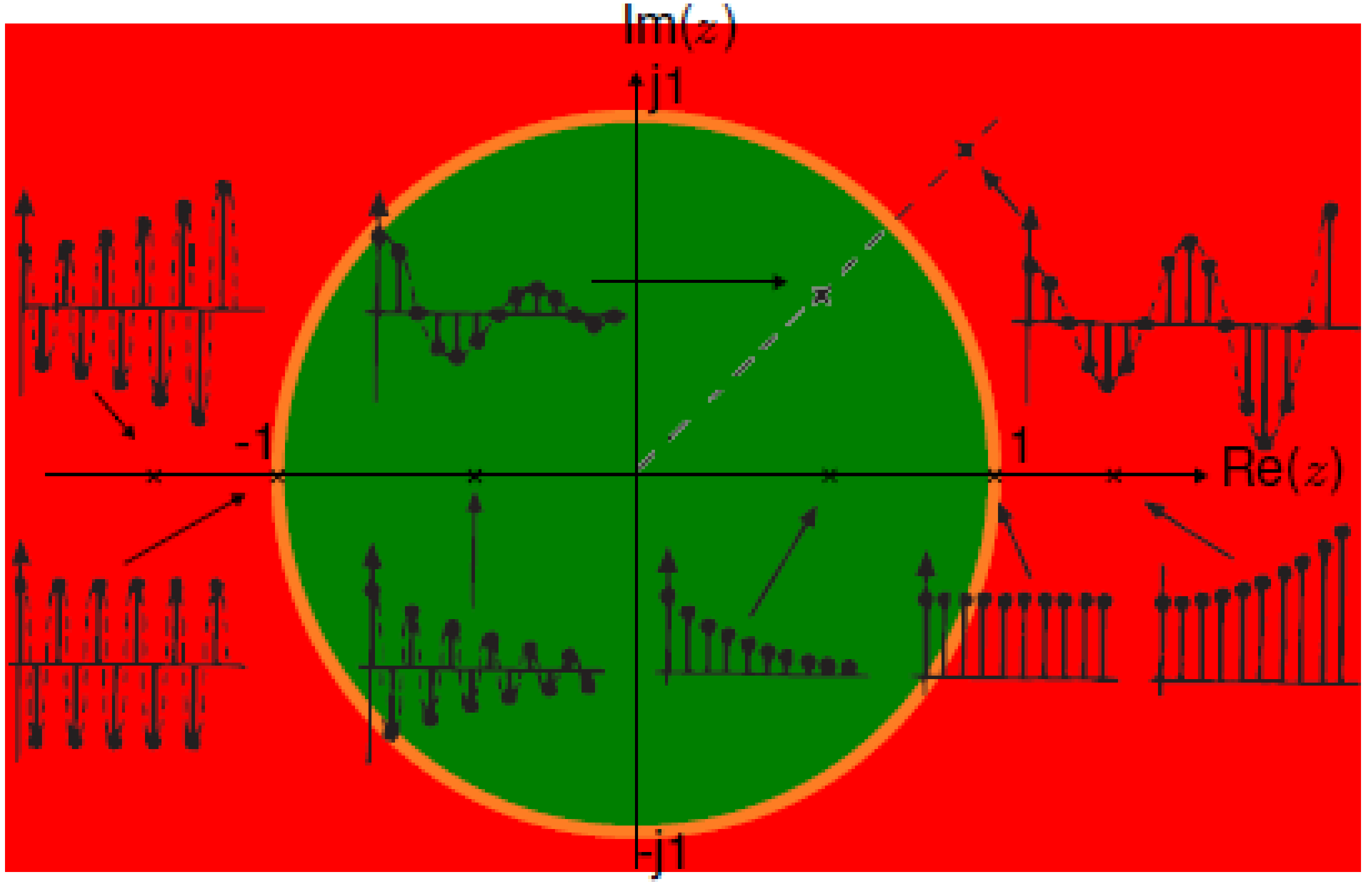
or

$$|p_j| = 1 \quad for \quad j \in \{1, 2, \dots, N\}$$

→ The system is unstable if a pole (e.g. p_j) lies outside the unit circle, i.e.

$$|p_j| > 1 \quad for \quad j \in \{1, 2, \dots, N\}$$

Determination of stability



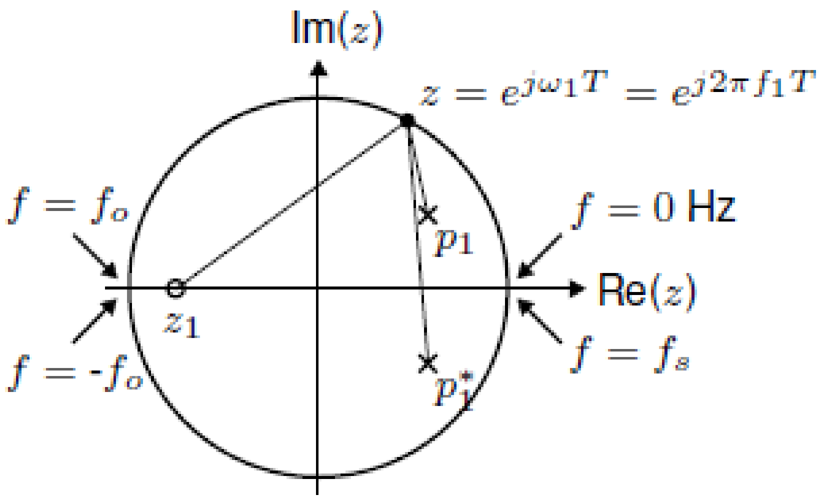
Stable

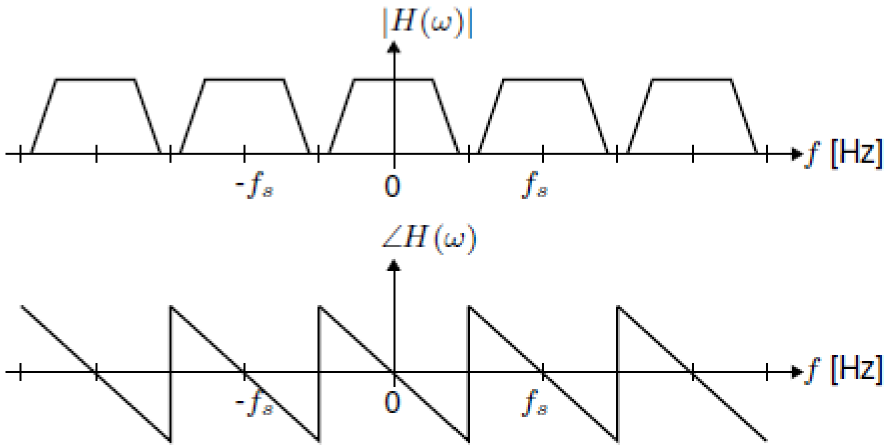
Marginally stable

Unstable

Frequency response

The frequency response is found by finding the amplitude and phase of $H(z)$ lies on the **unit circle, i.e.** $\boldsymbol{z = e^{j\omega T}}$





$$|H(z)| = a_0 \frac{|z - z_1|}{|z - p_1||z - p_1^*|}$$

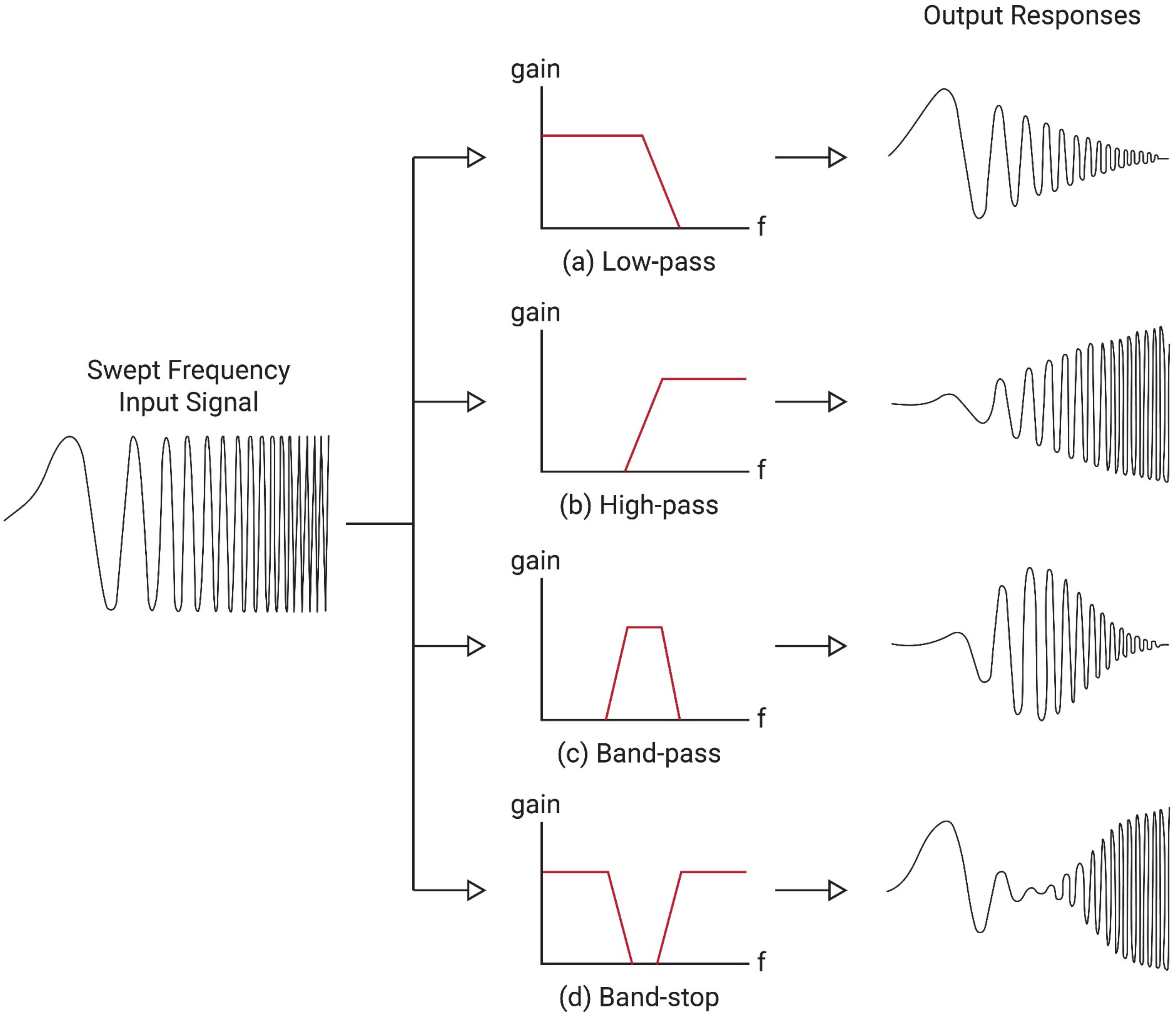
$$\angle H(z) = \psi_1 - \theta_1 - \theta_2$$

SDU

Filter design

Filter types

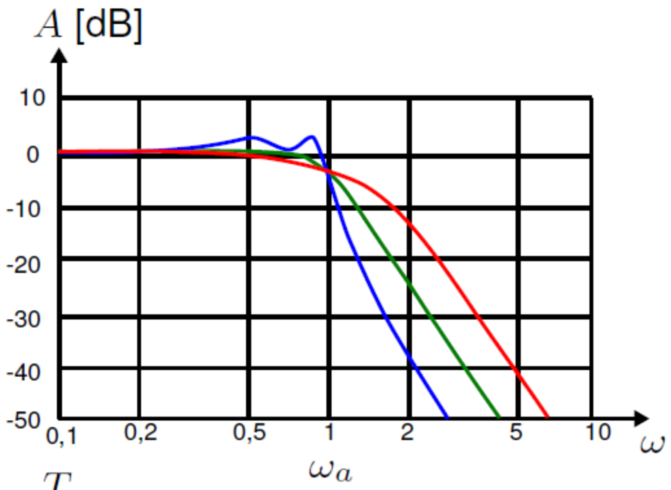
→ Lowpass filter
→ Highpass filter
→ Bandpass filter
→ Bandstop filter

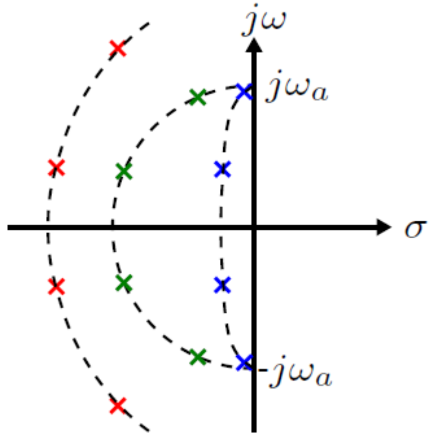


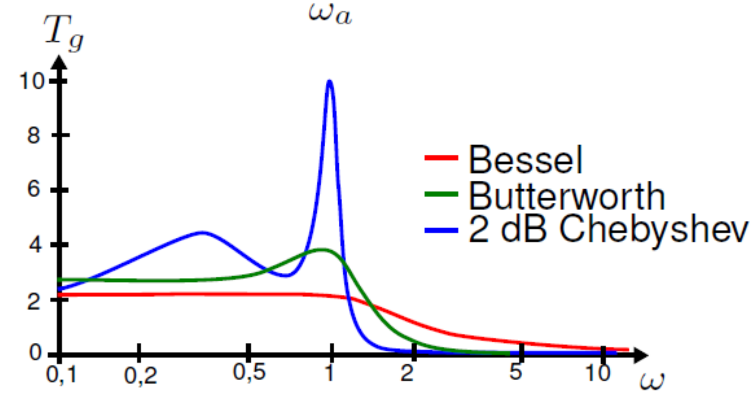
SDU

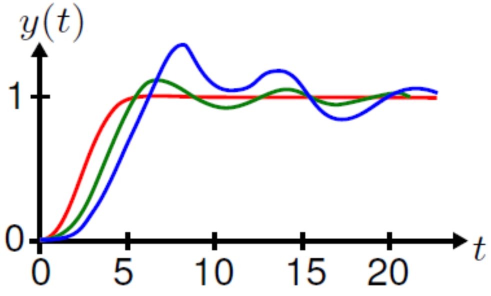
Analog filter

Basic filter functions





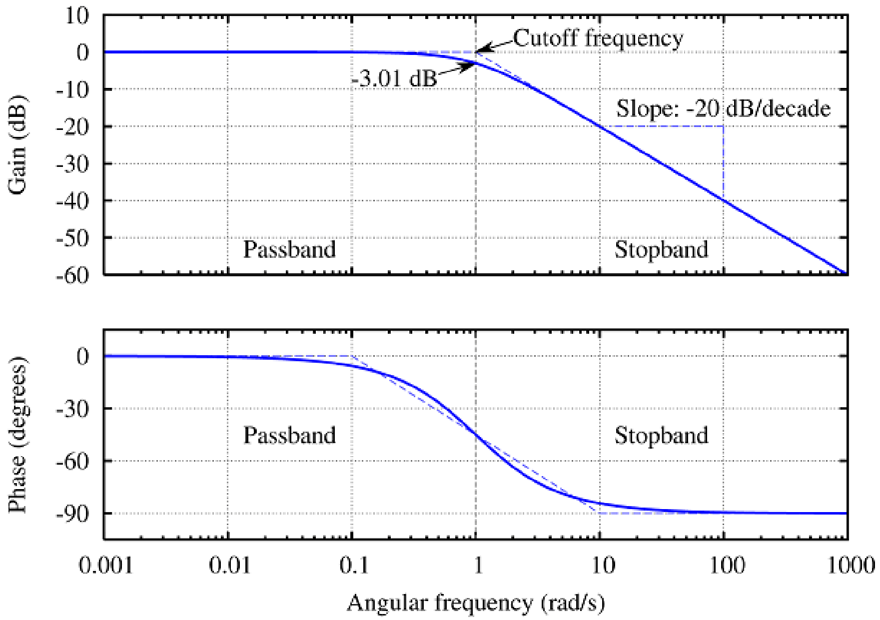


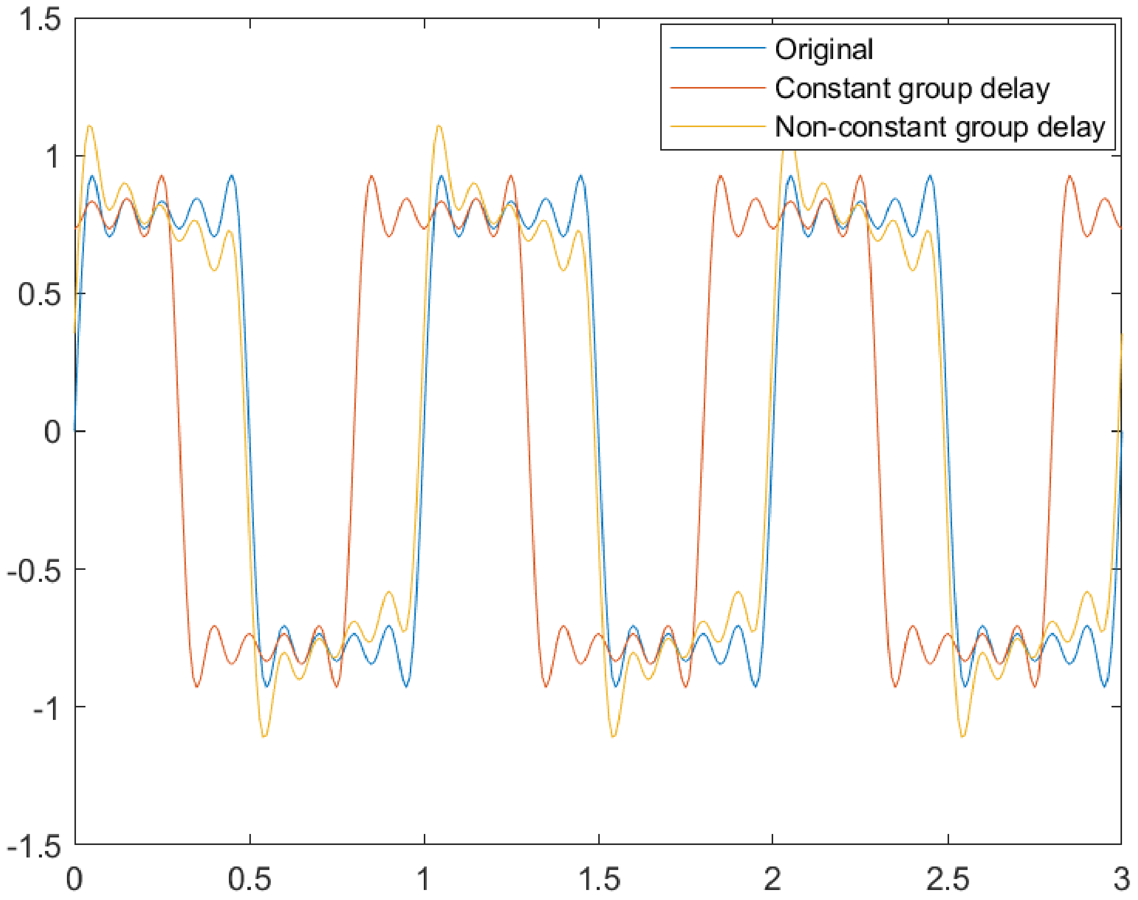


Group delay

→ $T_g = -\frac{d\phi(\omega)}{d\omega}$ [s]

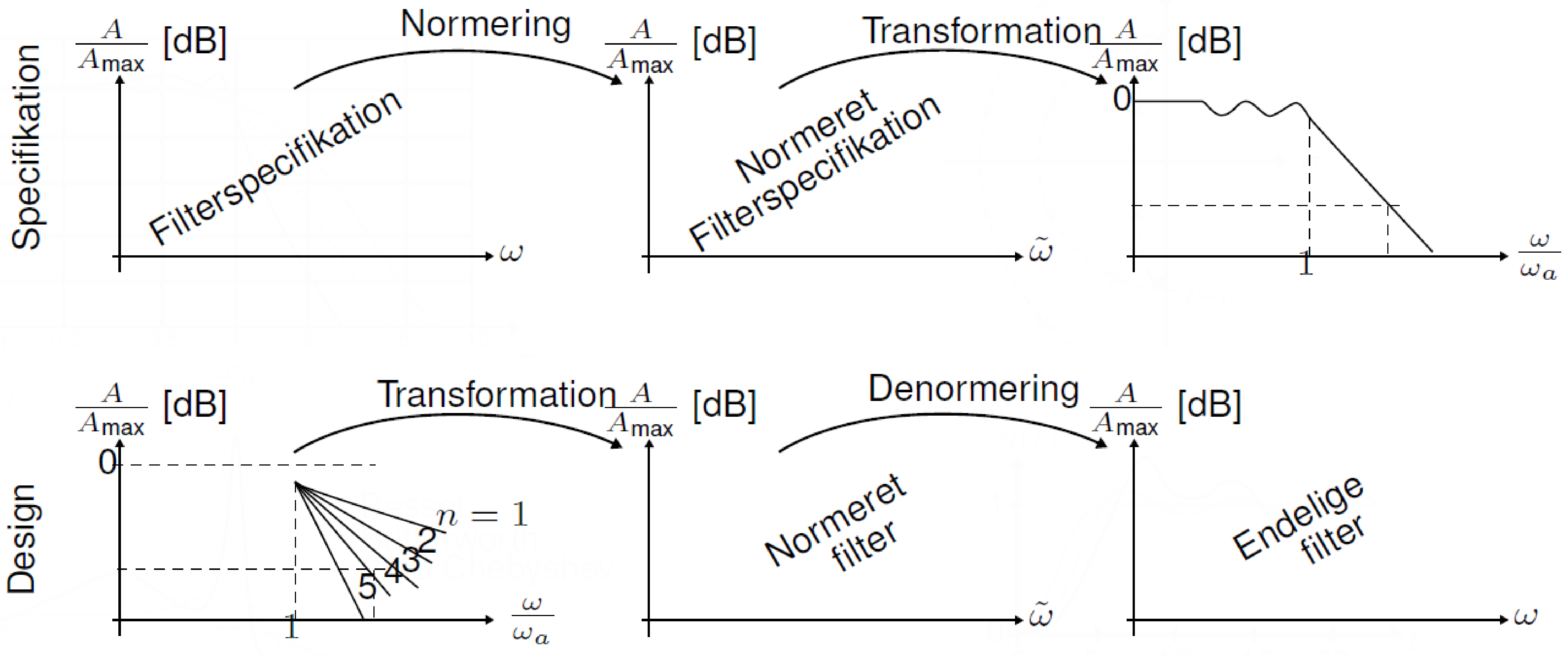
→ A filter has linear phase means it have constant group delay.





SDU

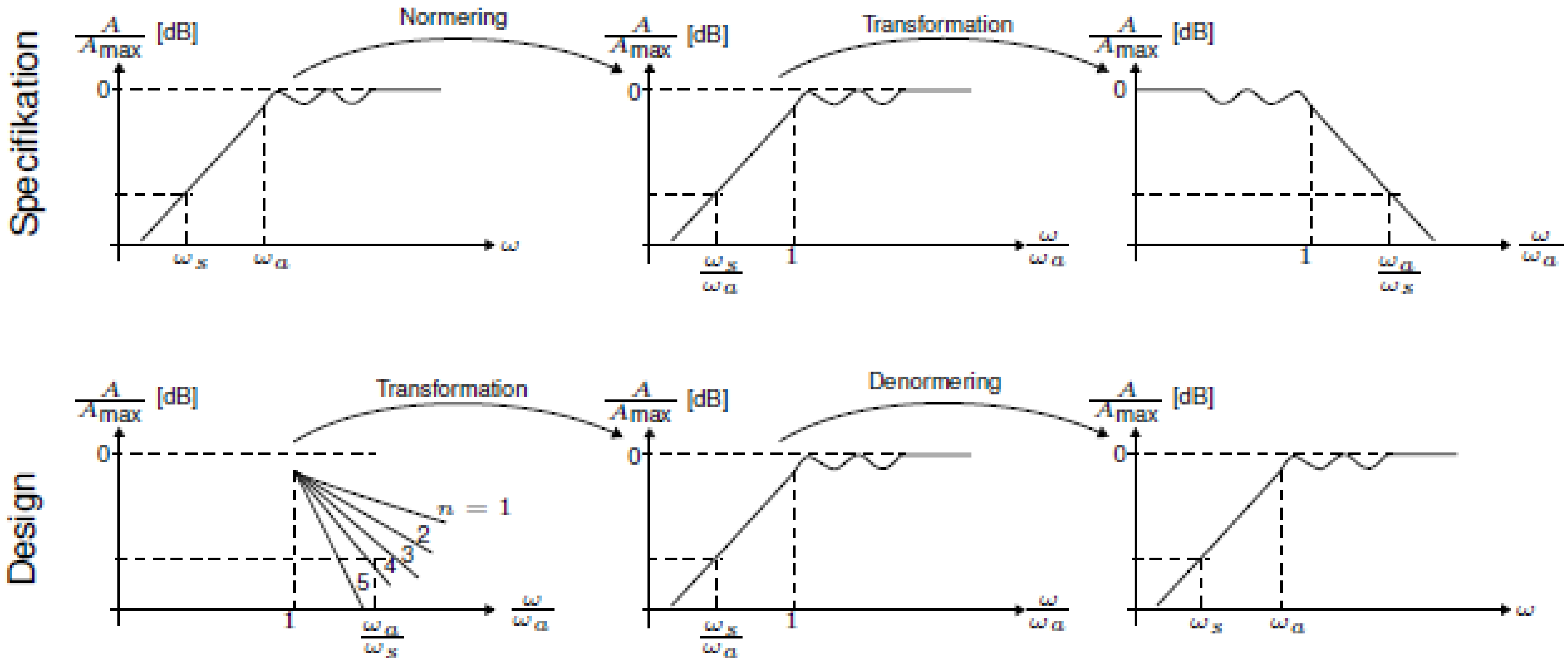
Analog low pass filter design



SDU

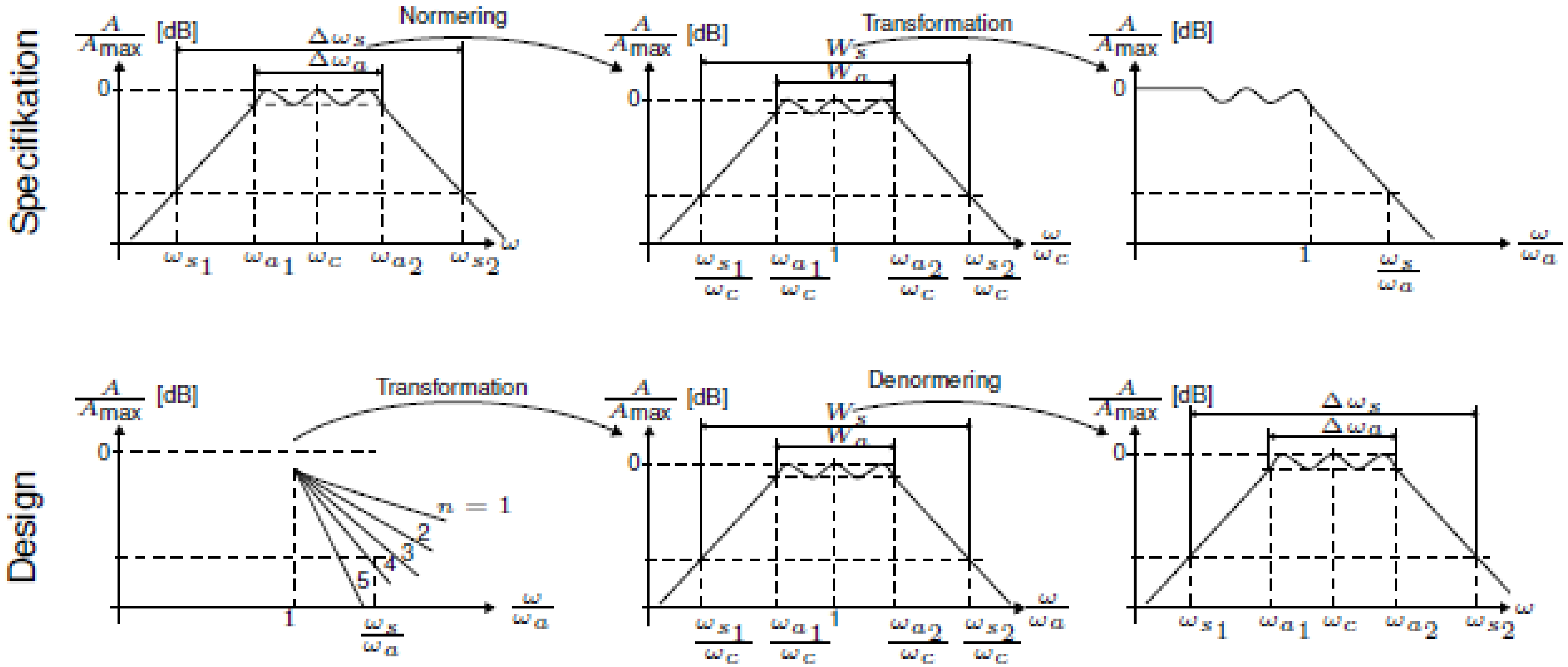
From Lowpass filter to highpass filter

$$H_{hp}(s) = H_{lp}(\bar{s})\Big|_{\bar{s}=\frac{1}{s}}$$



From Lowpass filter to bandpass filter

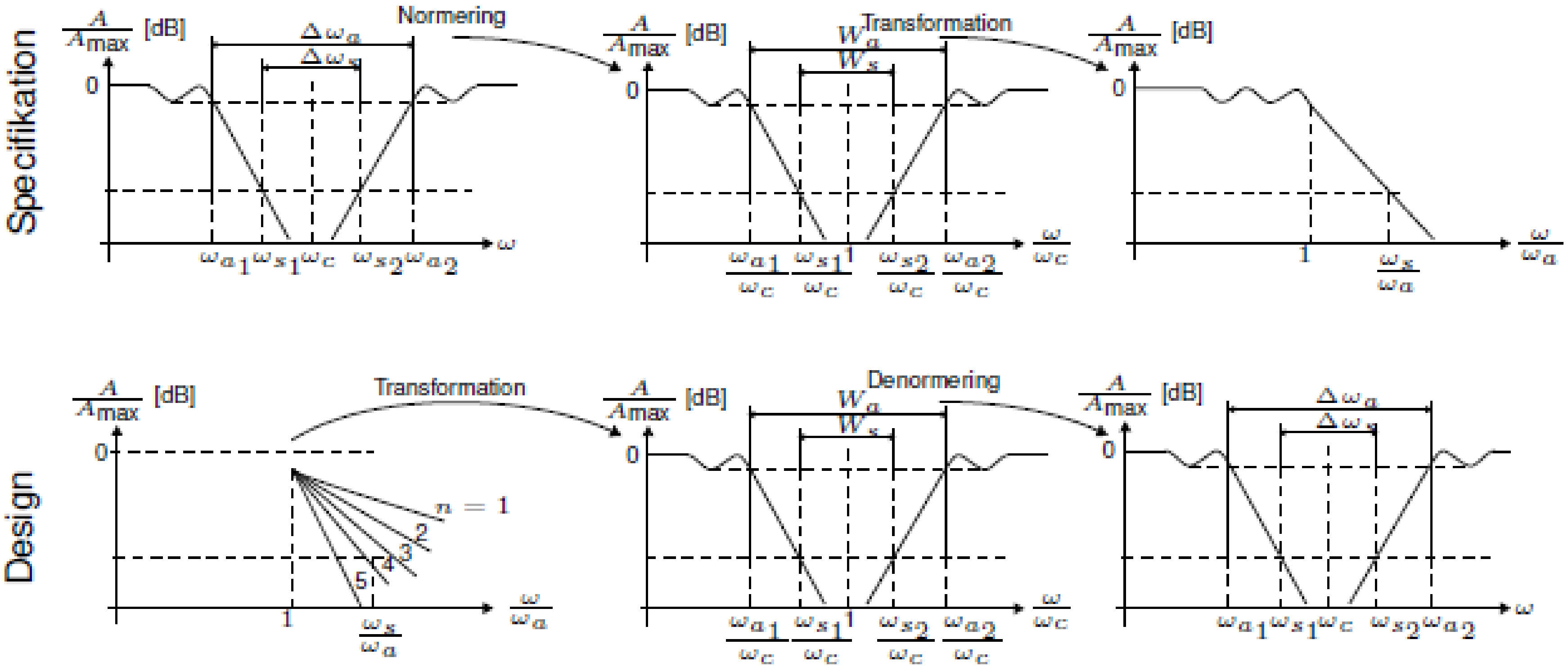
$$H_{bp}(s) = H_{lp}(\bar{s})\Big|_{\bar{s}=\frac{1}{W_a}(s+\frac{1}{s})}$$



SDU

From Lowpass filter to bandstop filter

$$H_{bs}(s) = H_{lp}(\bar{s})\Big|_{\bar{s}=\frac{W_a}{s+\frac{1}{s}}}$$



Digital filter designs

The goal is to make it satisfy the desired frequency response $H_d(e^{j\omega})$

IIR

$$H(z) = \frac{\sum_{k=0}^{M} b_k e^{-jk\omega}}{\sum_{k=0}^{M} a_k \, e^{-jk\omega}}$$

FIR

$$H(z) = \sum_{k=0}^{M} b_k e^{-jk\omega}$$

→ An FIR filter has 5 to 10 times larger realization structure than a corresponding IIR filter.
→ An FIR filter is always stable as it only has zero points.
→ An FIR filter is called a **non-recursive structure**, while an IIR filter is called a **recursive structure**.
→ An FIR filter is less sensitive to coefficient changes and rounding errors than an IIR filter.

IIR filter design

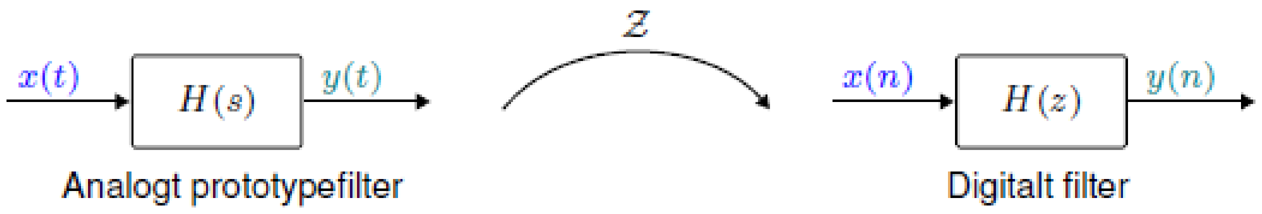
An IIR filter is designed by following the procedure

1. The filter's specifications are drawn up (analog filter)
2. Convert the analog filter to digital filter: z-domain transfer function is made
3. Choose a realization structure
4. Implement the design through program or hardware

Design methods:
→ Matched z-transform
→ Impulse invariant z-transform
→ Bilinear z-transform

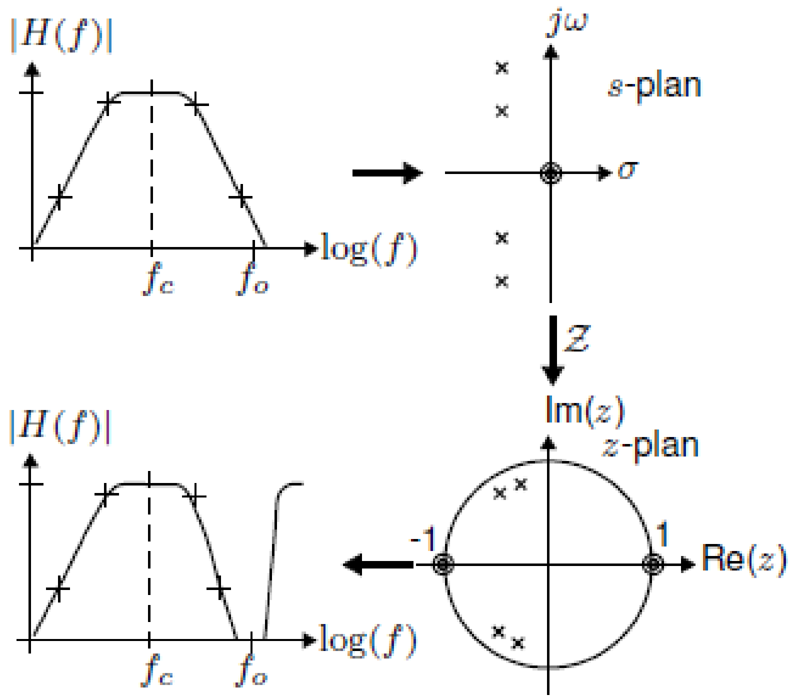
$$H(s) = \frac{\sum_{i=0}^{M} A_i s^i}{\sum_{i=0}^{N} B_i\, s^i}$$

$$H(z) = \frac{\sum_{i=0}^{M} a_i z^{-i}}{\sum_{i=0}^{N} b_i\, z^{-i}}$$



Matched z-transform

The transformation from s-domain to z-domain:
Transfer the **poles** and **zeros** from the s-plane to the z-plane. This will give a similar filter response.



Design procedure:

1. Determine the analog filter's transfer function $H(s)$.
2. Determine the analog filter’s poles and zeros.
3. Convert the **poles** in s-domain to z-domain $\boxed{z = e^{sT}}$
4. Determine the coefficients of the digital transfer function.
The numerator may be modified to have $H(z=1)=1$
5. Realize the transfer function as a **cascade structure**.

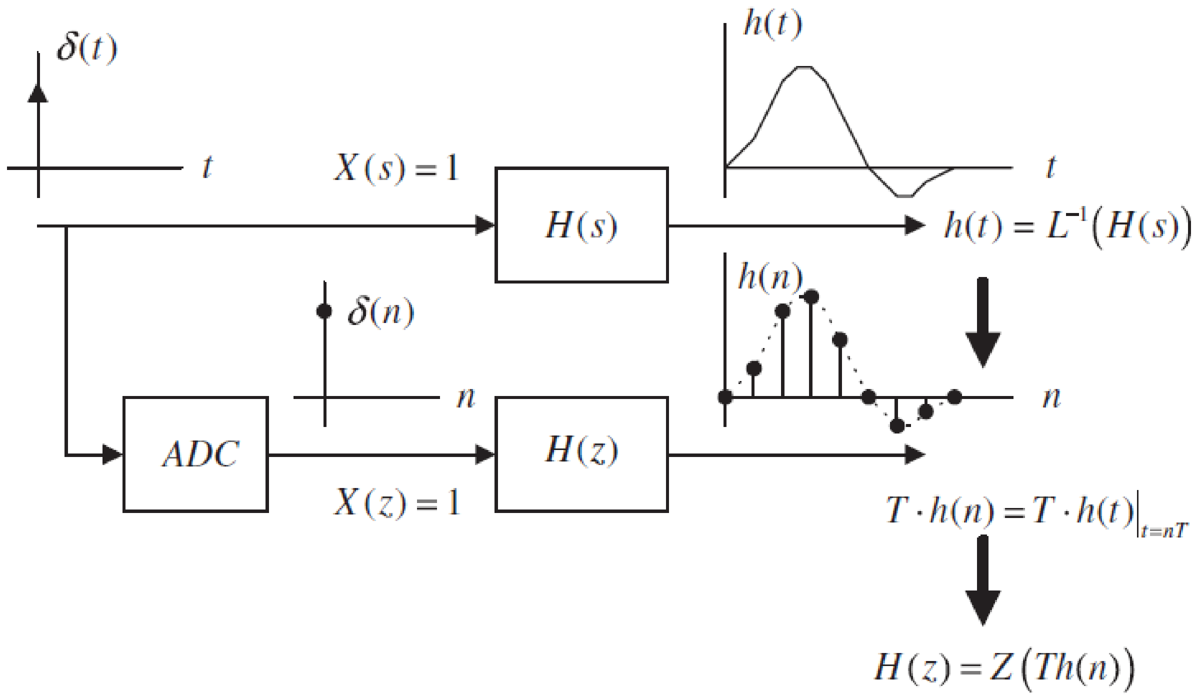
Impulse invariant z-transform

The analog impulse response can be achieved by using inverse Laplace transform of analog filter $H(s)$.

$$h(t) = \mathcal{L}^{-1}[H(s)]$$

Then we sample this analog impulse response with a sampling interval of T. Also, T is used as a scale factor.

$$T \cdot h(n) = T \cdot h(nT), \quad n \geq 0$$



Design procedure:

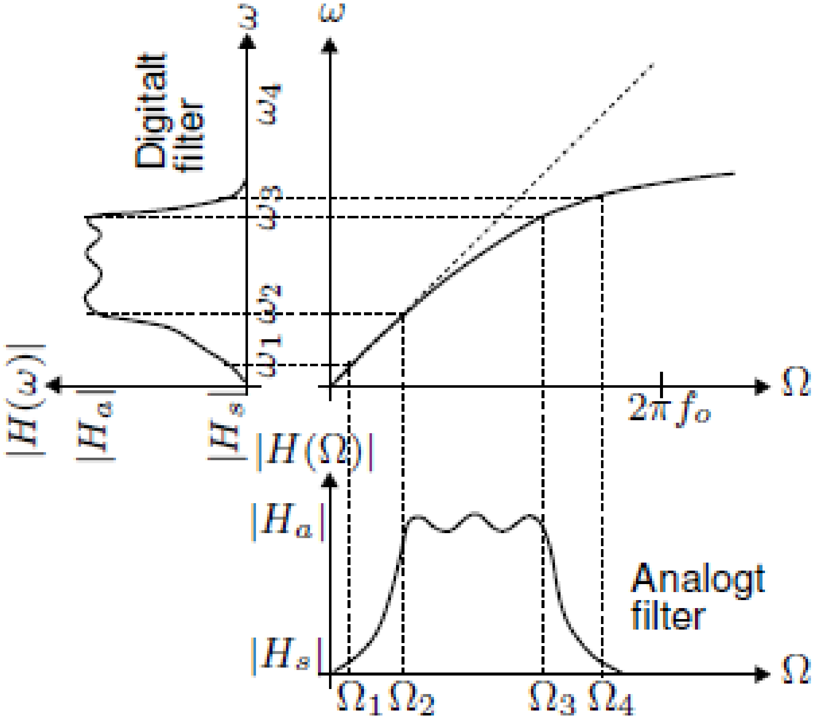
1. Determine the analog filter's transfer function $H(s)$.
2. Conduct partial fraction decomposition $H(s) = \sum_{i=1}^{N} \frac{r_i}{s-p_i}$.
3. Convert each section from s-domain to z-domain

$$H(z) = T\sum_{i=1}^{N} \frac{r_i}{1 - e^{p_i T} z^{-1}}$$

4. Realize the transfer function as a **parallel structure**.

Bilinear z-transform

Frequency warping
Since the frequency range $0 < f < \infty$ for the **analog filter** is transformed to the frequency range $0 < f < f_o$ for the **digital filter**, the frequency axis is deformed.



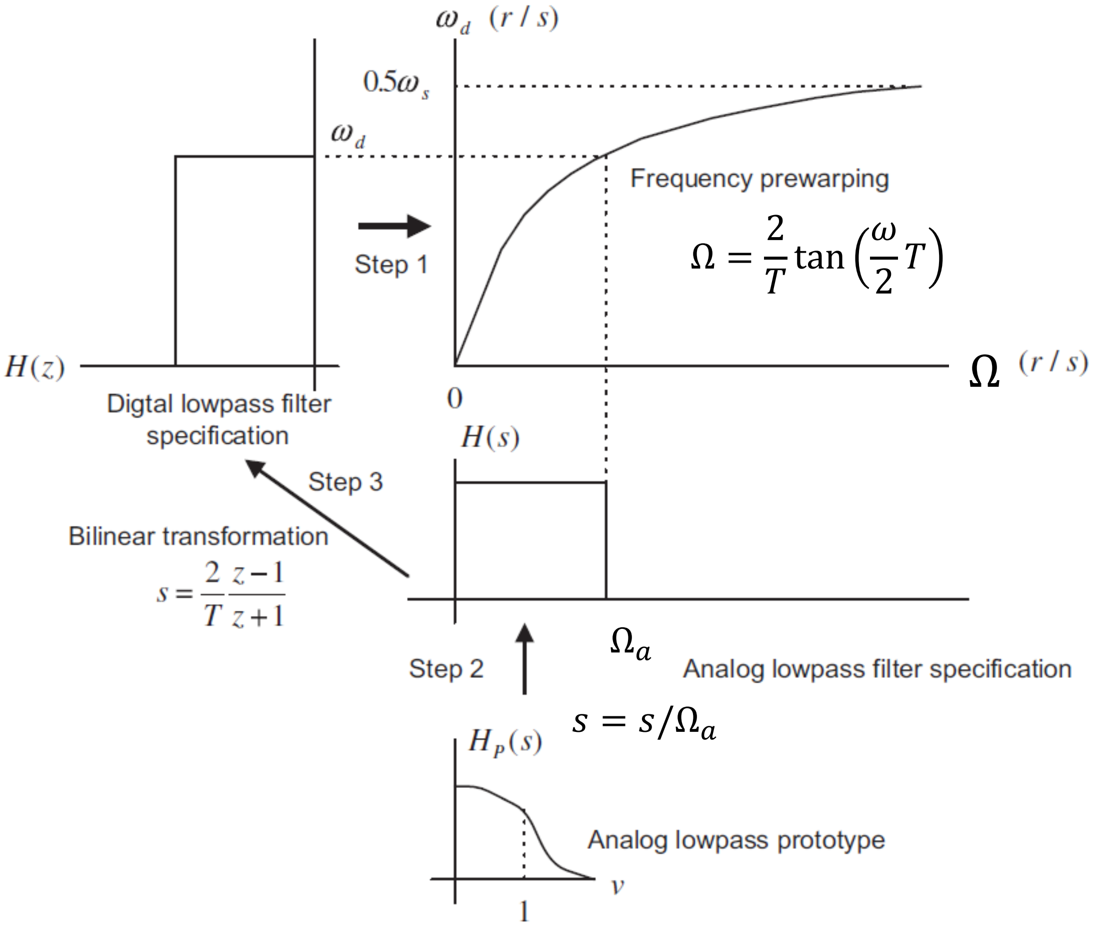
$$H(z) = H(s)\Big|_{s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}}$$

$$\Omega = \frac{2}{T}\tan\left(\frac{\omega}{2}T\right)$$

Frequency-normalized design

A lowpass digital filter with a cutoff frequency of 300 Hz is to be designed with a sampling frequency of 16 kHz. Please use bilinear z-transform method, and the **frequency-normalized** analog prototype filter should be

$$H_p(s) = \frac{1}{s+1}$$



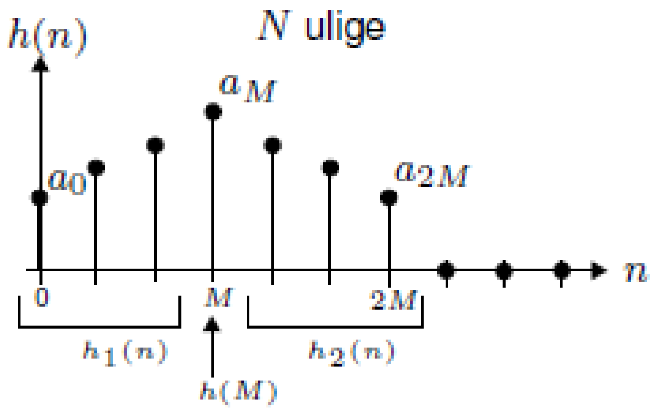
Design procedure:

1. Determine the transfer function of the analog filter $H_p(s)$
2. Compute pre-warp constant $C = \frac{1}{\tan\left(\frac{\omega_a T}{2}\right)}$
3. Convert the analog filter to the digital filter using

$$H(z) = H(s)\Big|_{s=C\frac{1-z^{-1}}{1+z^{-1}}}$$

4. The filter is realized as a **cascaded** realization structure

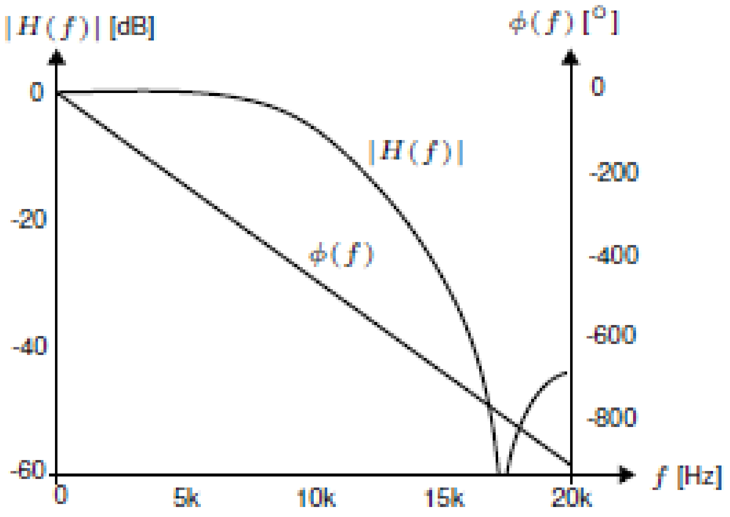
Design of FIR filter



An N^th order discrete time FIR filter has L=N+1 samples.

The design of FIR filter is to calculate the **filter coefficients** $\boldsymbol{a_i}$

Main idea: **Fourier transform of impulse response = frequency response of the desire filter**



FIR filter has linear phase:

$$\angle H(\gamma) = -M\pi\gamma$$

Design procedure - FIR filter (with window)

The construction of an FIR filter can proceed according to the following procedure

1. **Select window.** This selection is made according to the specified stopband and passband ripple.
2. **Determine filter order.** The filter order $2M$ is determined from the transition band Δf

$$\boxed{M = \frac{B_n f_s}{2\Delta f}}$$

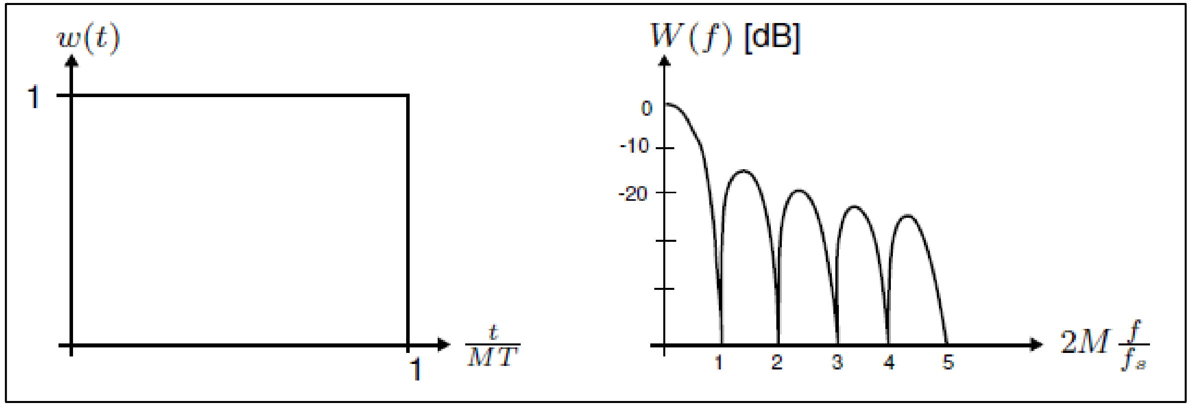
3. **Calculate filter coefficients.** The filter coefficients are calculated as $a_i = c_{M-i}\omega_{M-i}$
4. **Verification.** The amplitude characteristic of the filter is checked and, if necessary, the filter redesigned (M-value is corrected).

Design of FIR filter

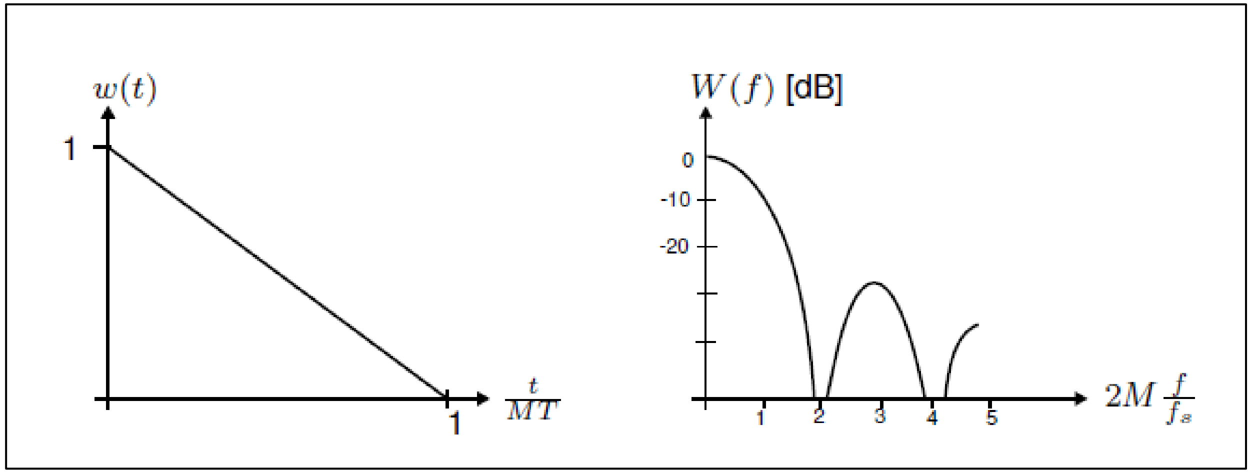
To calculate the filter coefficients for four different filter types.

Filtertype	c_0	$c_m = c_{-m}$	a_i
Lavpas	$2Tf_a$	$\frac{1}{m\pi}\sin(2\pi mTf_a)$	c_{M-i}
Højpas	$1-2Tf_a$	$\frac{1}{m\pi}(\sin(m\pi)-\sin(2\pi mTf_a))$	c_{M-i}
Båndpas	$2T(f_{a_2}-f_{a_1})$	$\frac{1}{m\pi}(\sin(2\pi mTf_{a_2})-\sin(2\pi mTf_{a_1}))$	c_{M-i}
Båndstop	$1-2T(f_{a_2}-f_{a_1})$	$\frac{1}{m\pi}(\sin(m\pi)+\sin(2\pi mTf_{a_1})-\sin(2\pi mTf_{a_2}))$	c_{M-i}

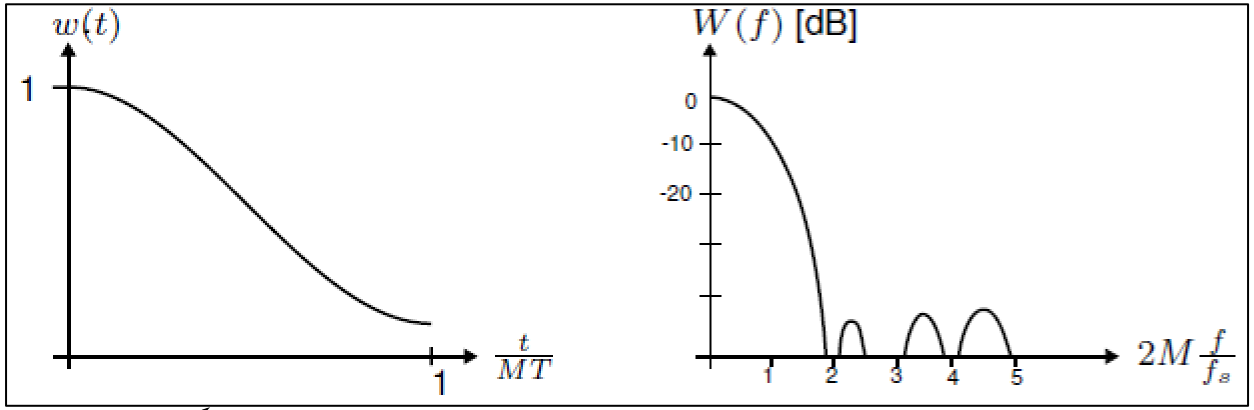
Window functions



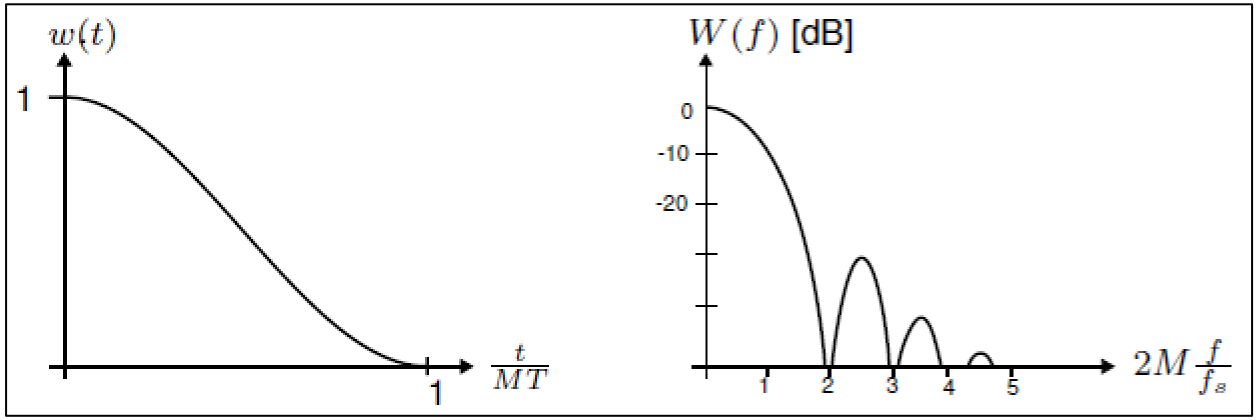
Rectangular window



Bartlett window

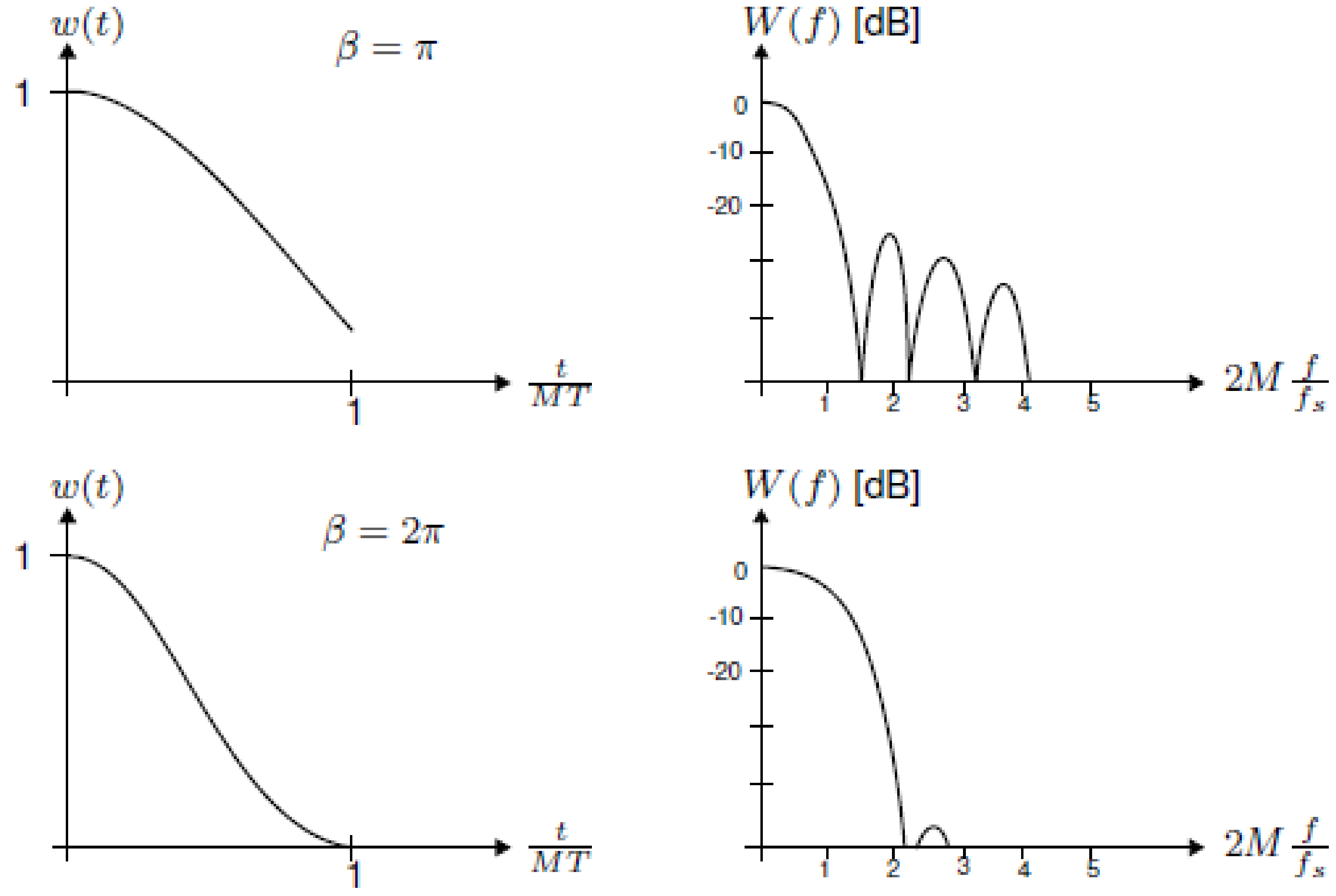


Hamming window



Hanning window

Kaiser window



SDU

Window function selection

Vindue	B_n	M_{min}	Min. stopbåndsdæmpning	Max. pasbåndsripple
Rektangulær	2	f_s/Δ_f	20 dB	1,5 dB
Bartlett	4	$2f_s/\Delta_f$	25 dB	0,1 dB
Hamming	4	$2f_s/\Delta_f$	50 dB	0,05 dB
Hanning	4	$2f_s/\Delta_f$	45 dB	0,1 dB
Kaiser ($\beta = \pi$)	2,8	$1,4f_s/\Delta_f$	40 dB	0,2 dB
Kaiser ($\beta = 2\pi$)	4,4	$2,2f_s/\Delta_f$	65 dB	0,01 dB

Følgende viser en oversigt over koefficienter for de betragtede vinduesfunktioner.

Vinduesfunktion	$w(n)$ for $-M \leq n \leq M$	$w(n)$ otherwise
Rektangulær	1	0
Bartlett	$1 - \frac{\lvert n \rvert}{M}$	0
Hamming	$0{,}54 + 0{,}46\cos(\frac{m\pi}{M})$	0
Hanning	$0{,}5 + 0{,}5\cos(\frac{m\pi}{M})$	0
Kaiser	$\frac{I_0\left(\beta\sqrt{1-(\frac{n}{M})^2}\right)}{I_0(\beta)}$	0

Parameteren β justerer primært side lobe amplituden (normalt er β mellem 1 og 10). Funktionen $I_0(x)$ er en nulte ordens Besselfunktion defineret som

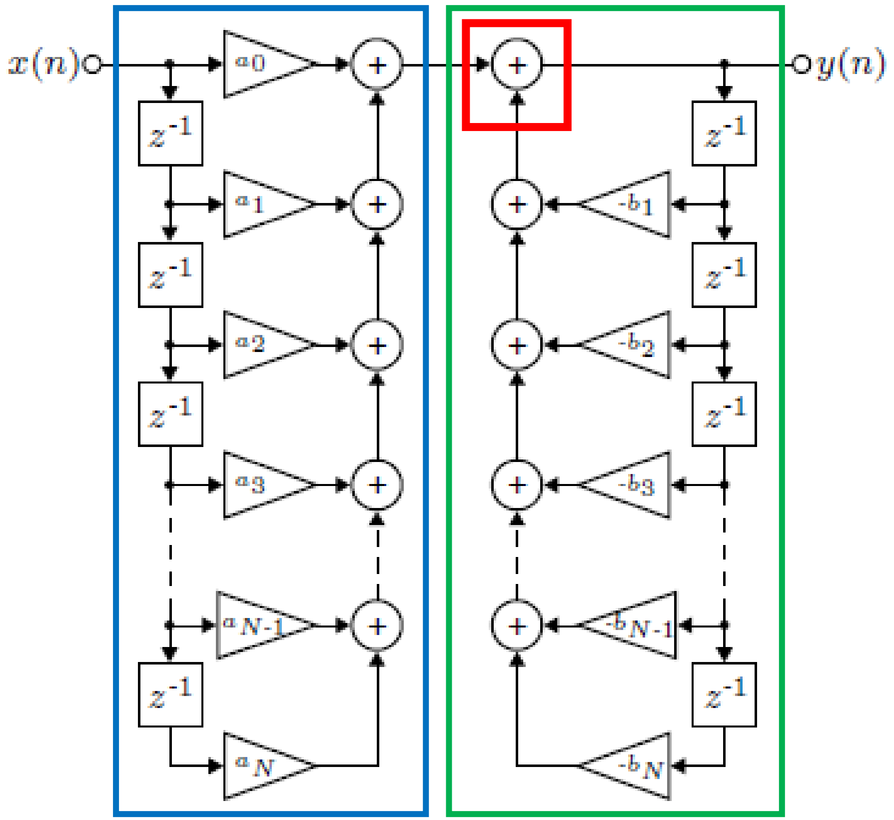
$$I_0(x) = 1 + \sum_{k=1}^{\infty} \left(\frac{1}{k!} \left(\frac{x}{2} \right)^k \right)^2$$

Not required.
You can find the values using matlab

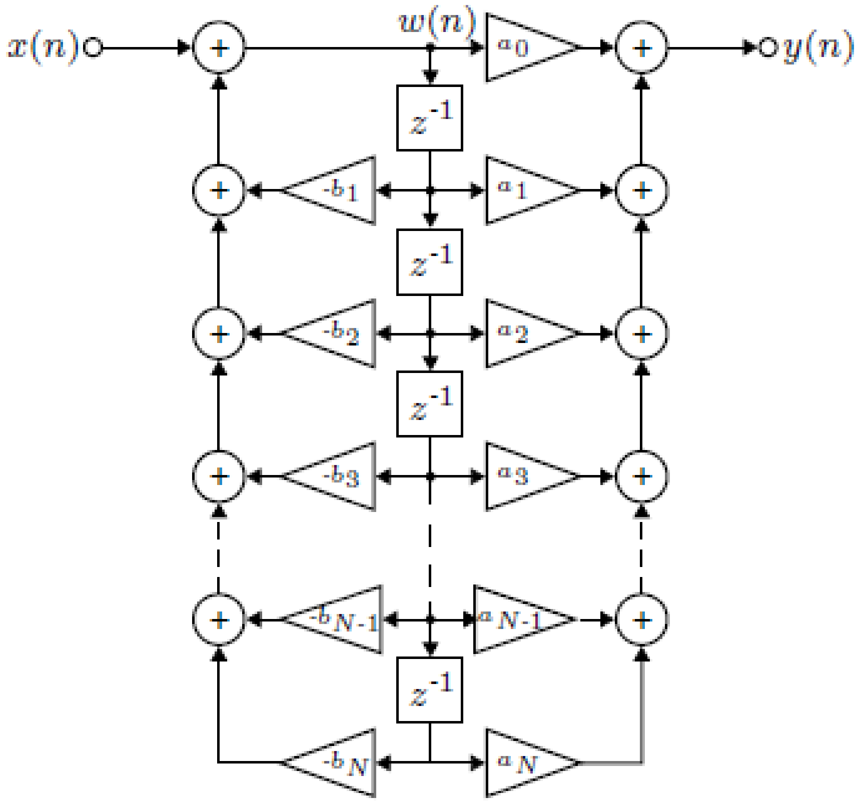
Realization & Implementation

Direct realization

The transfer function $H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{N} a_i z^{-i}}{1+\sum_{i=1}^{N} b_i z^{-i}}$, corresponds to difference equation $y(n) = \sum_{i=0}^{N} a_i x(n-i) - \sum_{i=1}^{N} b_i y(n-i)$



Direct Type I structure

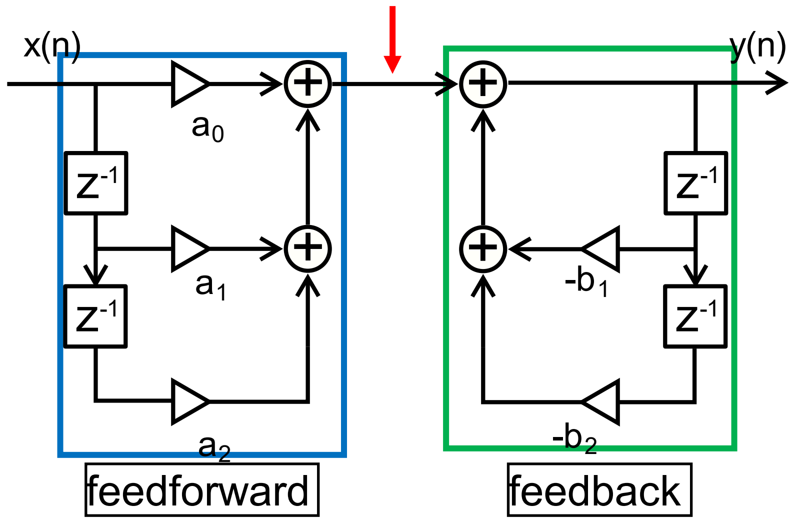


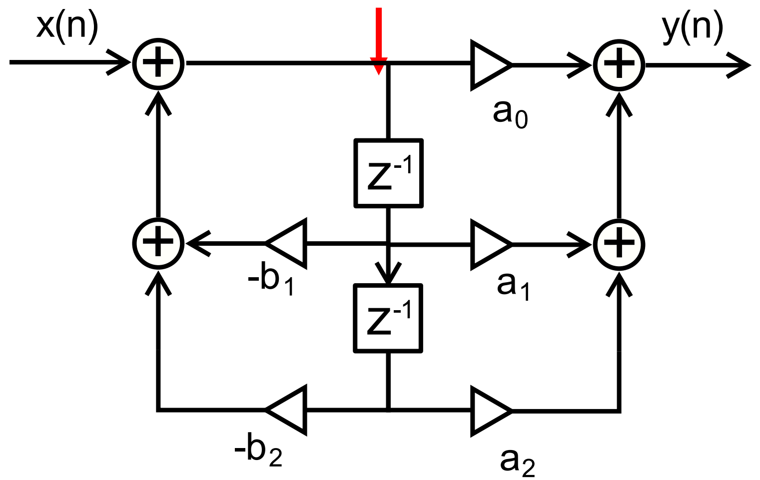
Direct Type II structure

SDU

Type I structure v.s. Type II structure

→ Type I uses separate feedforward and feedback loop, while Type II combines them together. The intermediate states in Type II can experience **larger signal swings**.





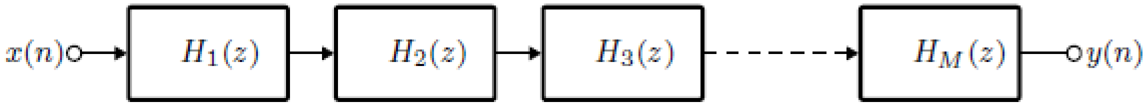
→ In Type II, the coefficient a_i act on the internal states, which make the system **more sensitive to the rounding errors**.

Cascade realization

A **higher-order system** is often realised as a cascade form

$$H(z) = H_1(z) \cdot H_2(z) \cdot H_3(z) \cdot \cdots \cdot H_M(z)$$

$$H(z) = H_1(z) \cdot H_2(z) \cdot H_3(z) \cdot \cdots \cdot H_M(z) = \frac{(z - z_1)(z - z_2)}{(z - p_1)(z - p_2)} \cdot \frac{(z - z_3)(z - z_4)}{(z - p_3)(z - p_4)} \cdot \cdots \cdot \frac{z - z_M}{z - p_M}$$



The M sections of the sub-system are 1st order or 2nd order transfer functions

$$H_k(z) = \frac{a_{0k} + a_{1k} z^{-1}}{1 + b_{1k} z^{-1}} \quad \text{or} \quad H_k(z) = \frac{a_{0k} + a_{1k} z^{-1} + a_{2k} z^{-2}}{1 + b_{1k} z^{-1} + b_{2k} z^{-2}}$$

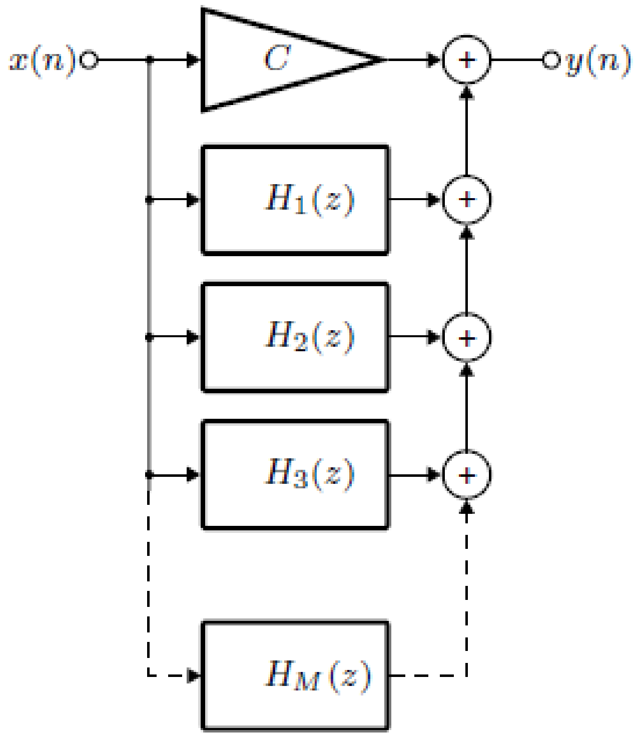
Parallel realization

A parallel realization of the filter is represented as

$$\mathrm{H(z)} = \mathrm{C} + H_1(z) + H_2(z) + \ \dots + H_M(z)$$

Partial fraction decomposition is used.

Compared to cascade structure, parallel realization is easier to tune each component.
Parallel structure can lead to sum of large signals, which can have higher risk of overflow.



Implementation

Quantization error

The signal varies from –V to +V, and has 2^N quantization levels.

$$2V = 2^N \Delta V$$

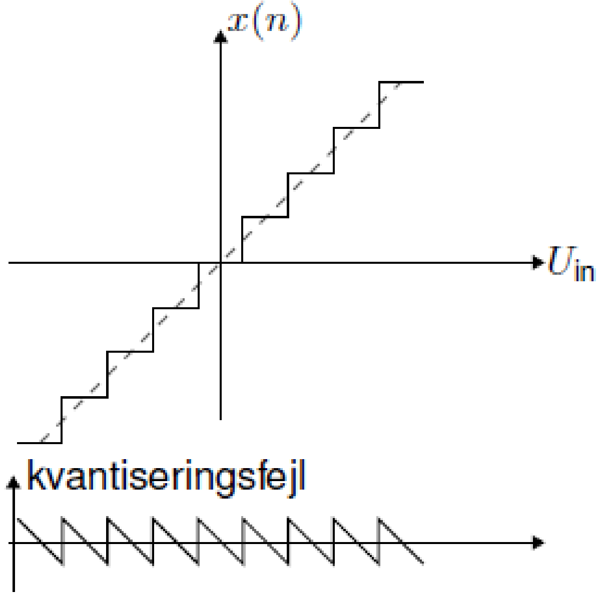
The quantization error is modeled as a uniform distribution in $-\frac{\Delta V}{2} \leq e_q \leq \frac{\Delta V}{2}$. According to the theory of probability and random variables, the quantization noise is calculated as

$$E(e_q^2) = \frac{\Delta V^2}{12}$$

Signal-to-Quantization Noise Ratio (SQNR) can be calculated as

$$SQNR = \frac{E((V/\sqrt{2})^2)}{E(e_q^2)} = \frac{\left(\frac{2^N \Delta V}{2\sqrt{2}}\right)^2}{\frac{\Delta V^2}{12}} = (2^N \cdot \frac{3}{2})^2$$

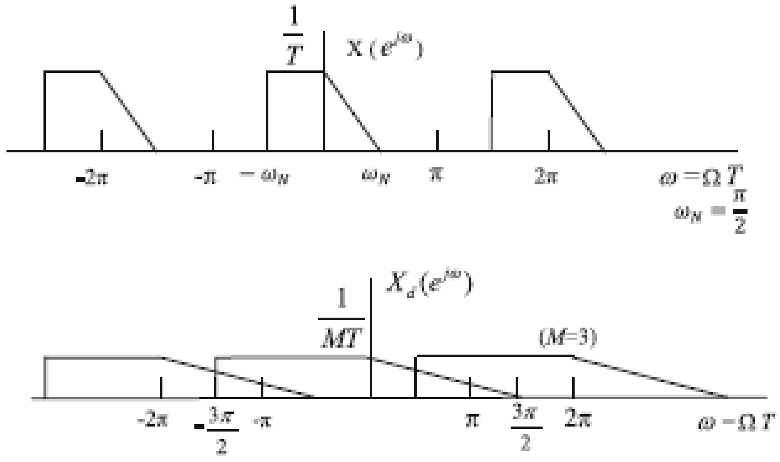
$$SQNR_{dB} = 20\log_{10}(2^N \cdot \frac{3}{2}) \approx 6N \ \ [\text{dB}]$$

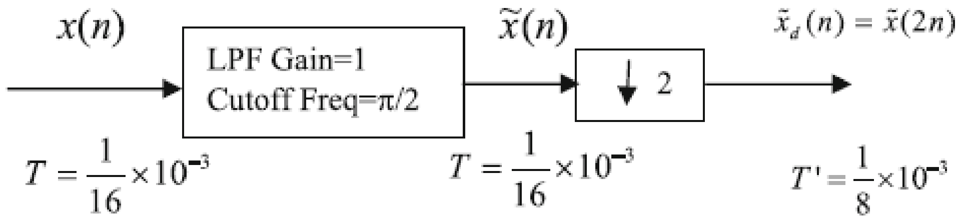


Multirate sampling

Downsampling:

$$x_d(n) = x(nM)$$



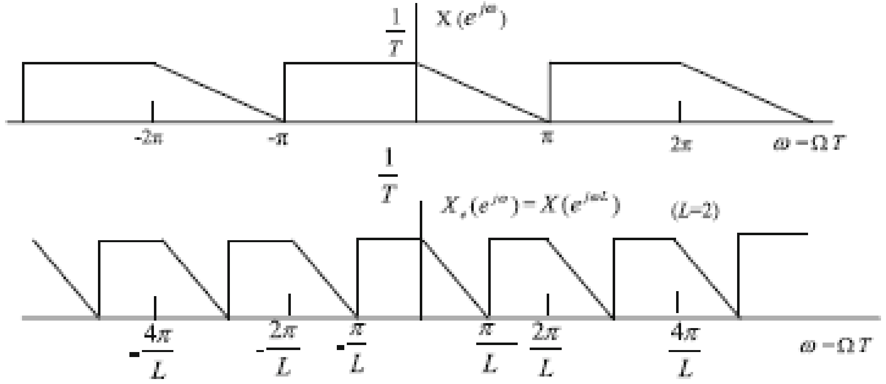


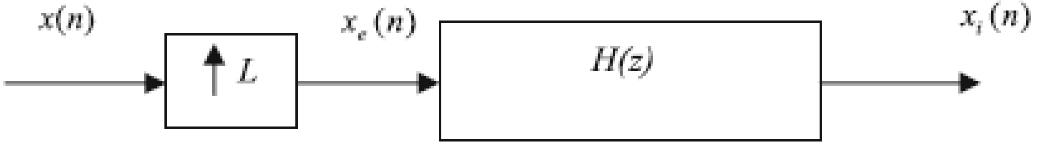
SDU

Multirate sampling

Upsampling:

$$x_u(n) = \begin{cases} x(n/L) & \text{hvis } n/L \in \mathbb{Z} \\ 0 & \text{ellers} \end{cases}$$





Sampling period T Sampling period $T' = T / L$ Sampling period $T' = T / L$