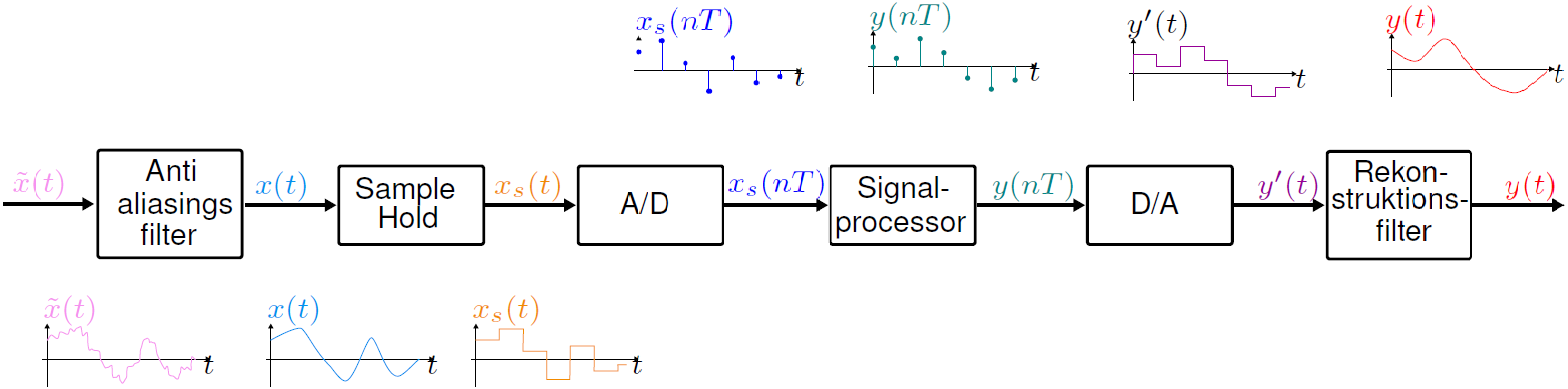
Filter

Zhuoqi Cheng
zch@mmmi.sdu.dk
SDU Robotics

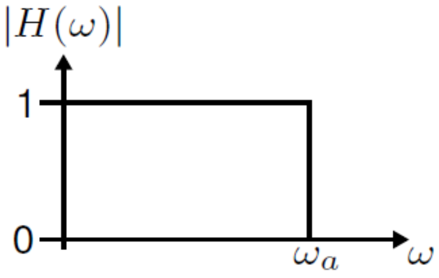
Topics to be covered in this course

→ Sampling and reconstruction
→ Aliasing
→ Quantization and dynamic range
→ Implementation
→ Conversion time-frequency domain
→ Z transform
→ Linear Time Invariant system (LTI)
→ System analysis
→ Window functions
→ Filter design
→ Impulse response (FIR and IIR)

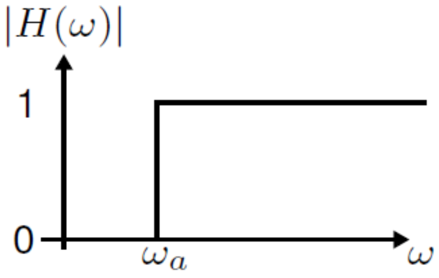
Digital signal processing workflow



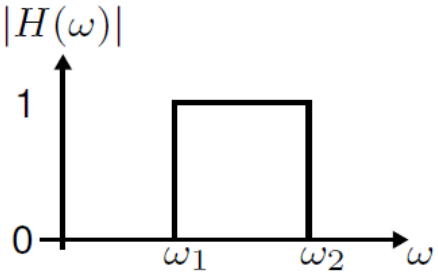
Filters



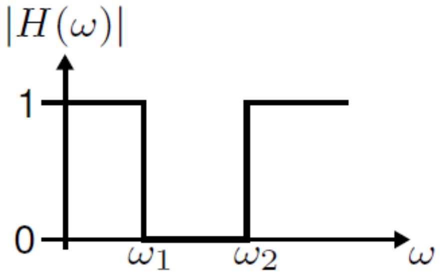
Low pass



High pass



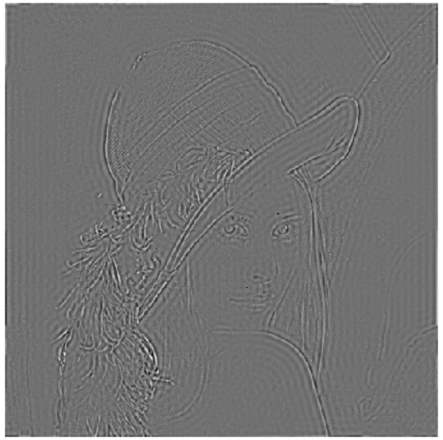
Band pass



Band stop

Low-pass Filtered Image

High-pass Filtered Image

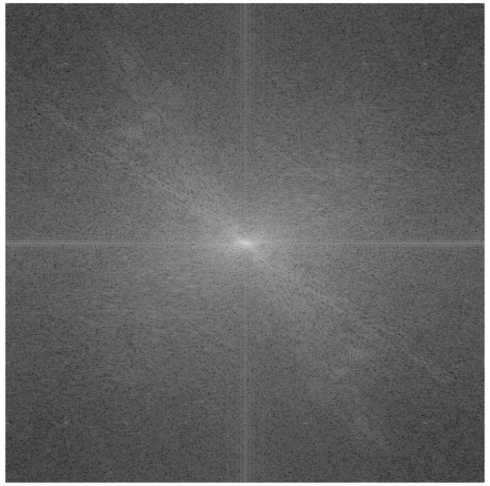


Band-pass Filtered Image

Band-stop Filtered Image

Original Image

Magnitude Spectrum

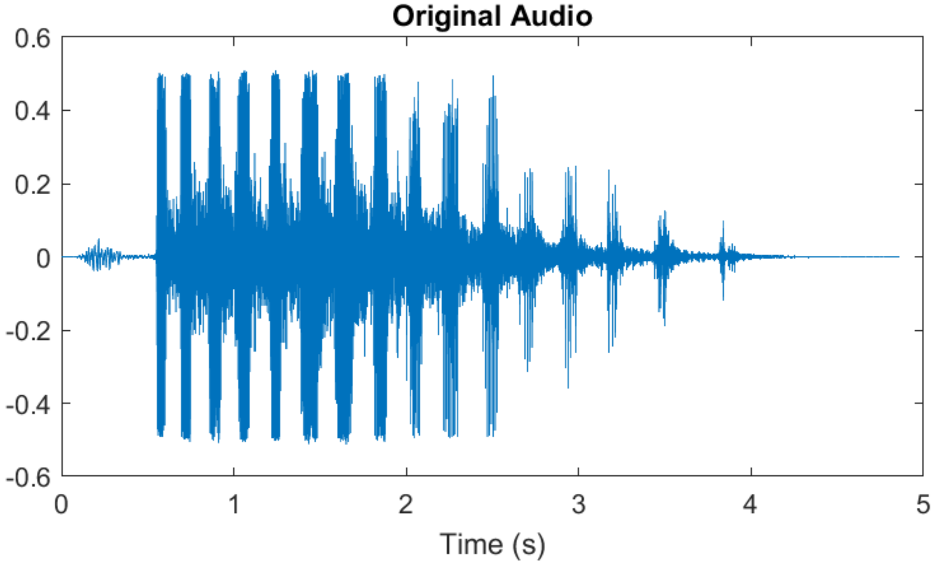


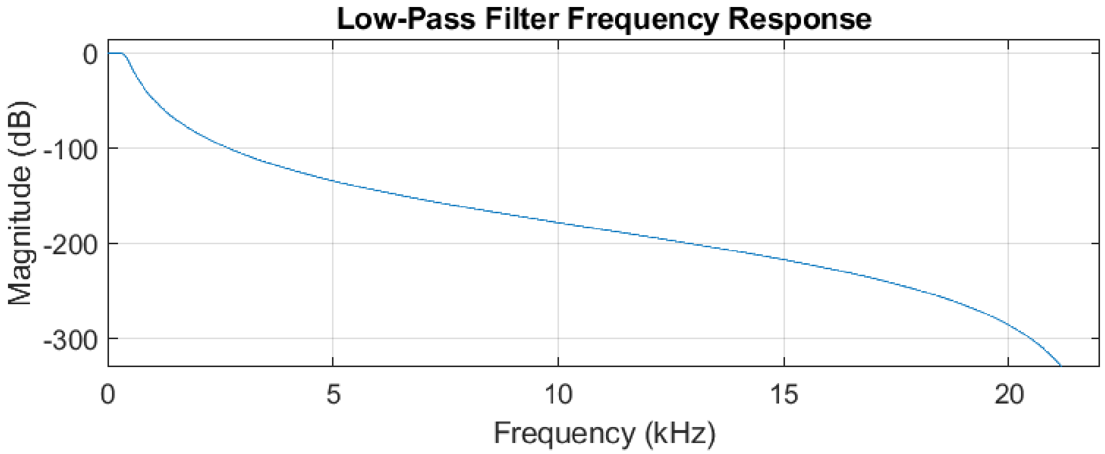
Sound signal processing

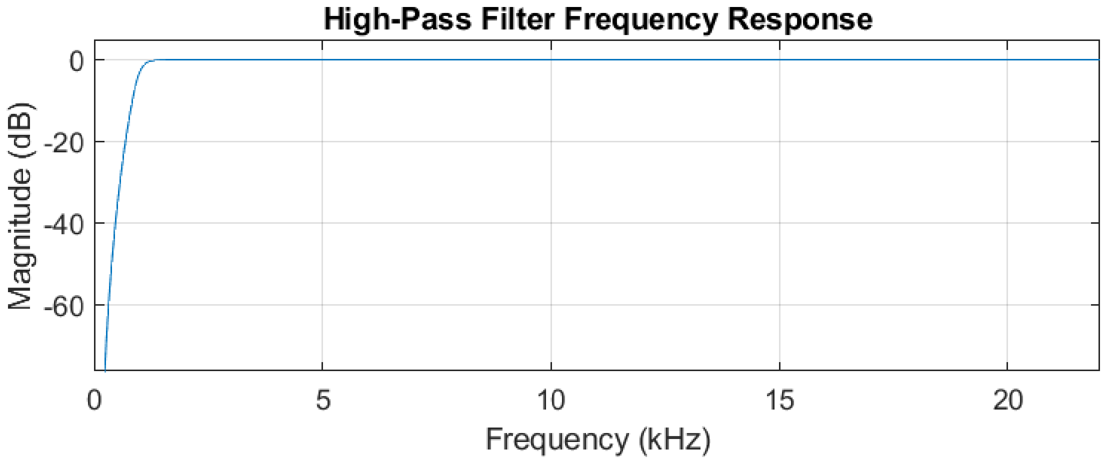
→ Original soundtrack

→ Low frequency soundtrack

→ High frequency soundtrack

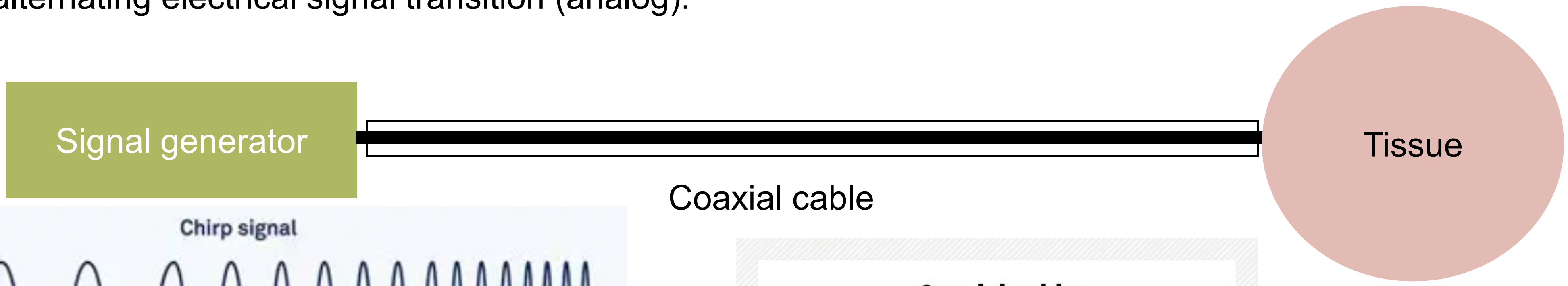


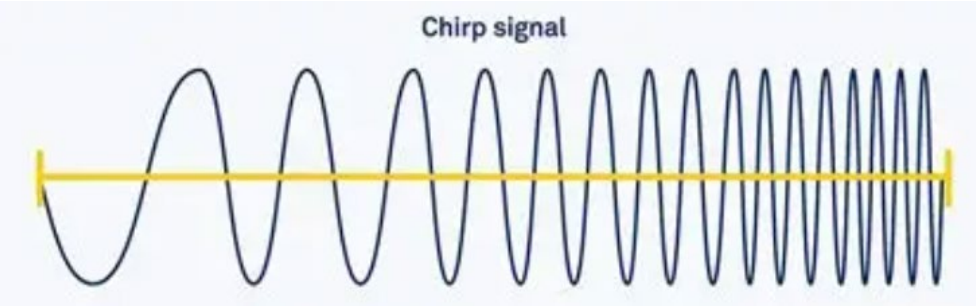


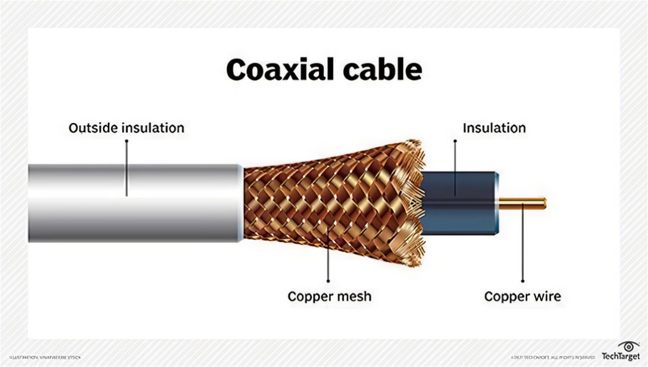


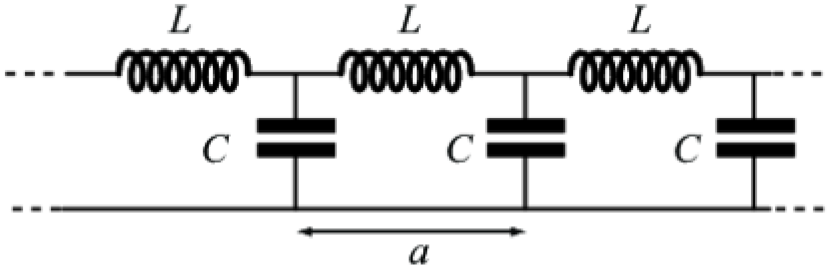
Example: analog filter

Example of alternating electrical signal transition (analog):

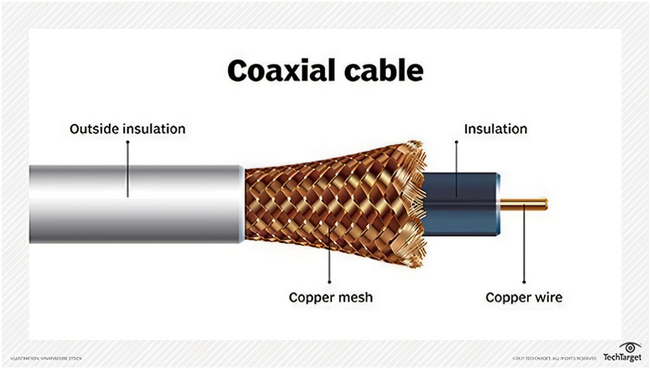


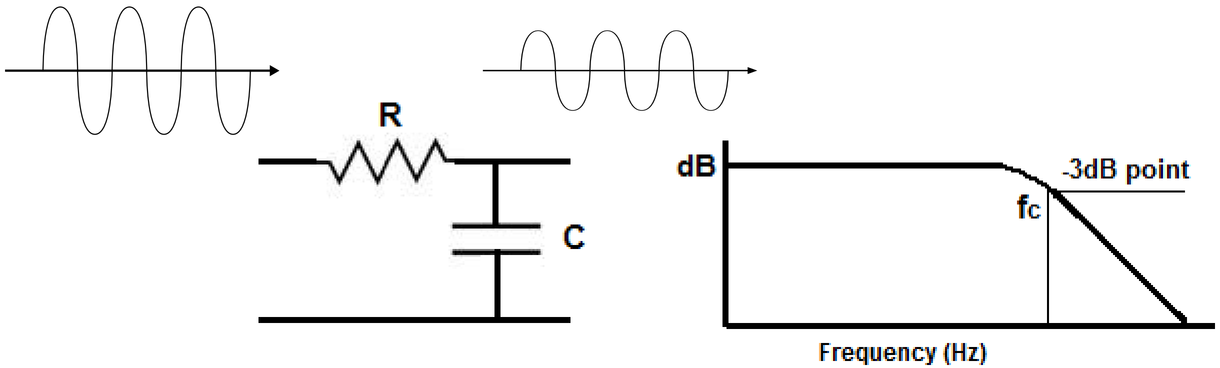






Analog low pass filter





$$f_c = \frac{1}{2\pi RC}$$

For analog signal, physical components are chosen to construct an analog filter. (**Hardware**)

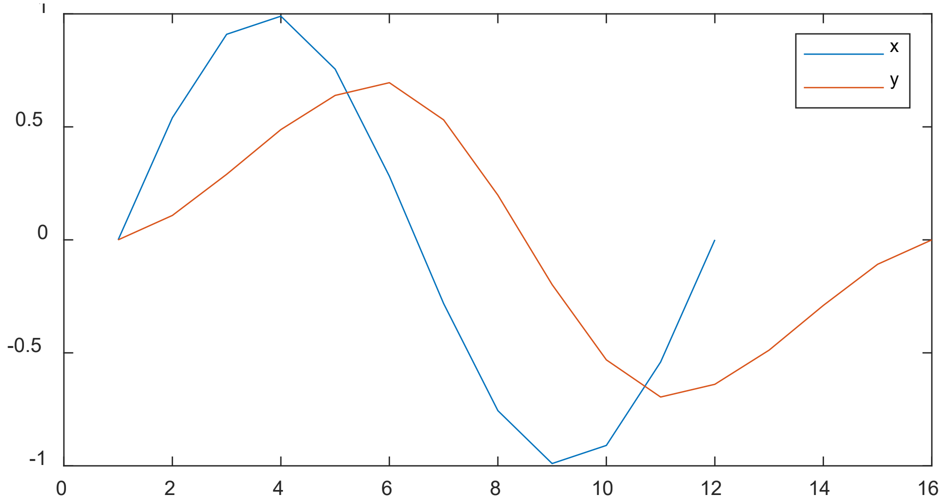
Digital signal and digital filter

Digital signal is a type of signal that **represents information using discrete values.**

Example of digital signal:
X = [0, 0.541, 0.910, 0.990, 0.756, 0.282, -0.282, -0.756, -0.990, -0.910, -0.541, 0]

Example of digital low pass filter (moving average):
H = [0.2, 0.2, 0.2, 0.2, 0.2]

Operation: convolution
Y = X*H

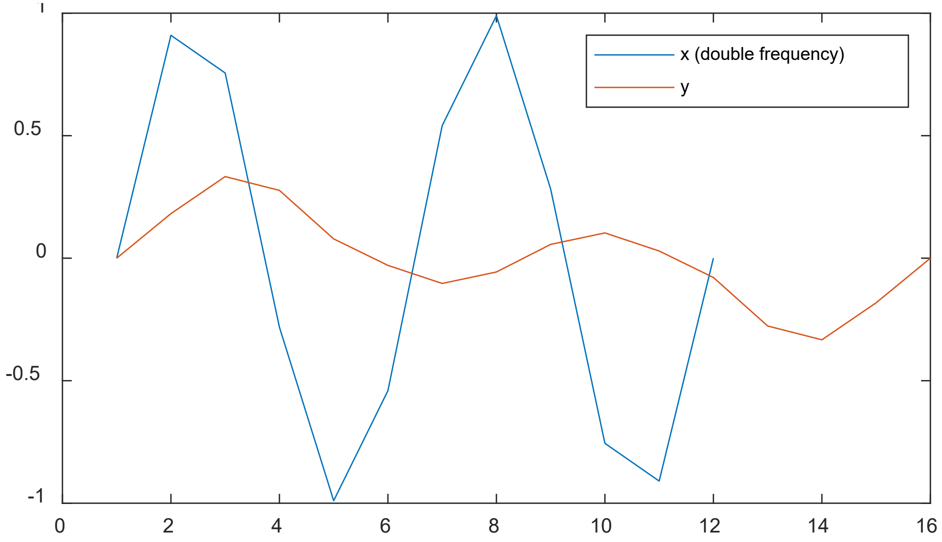


Digital signal and digital filter

If the input digital signal x becomes double frequency, and the same digital filter is used
H = [0.2, 0.2, 0.2, 0.2, 0.2]

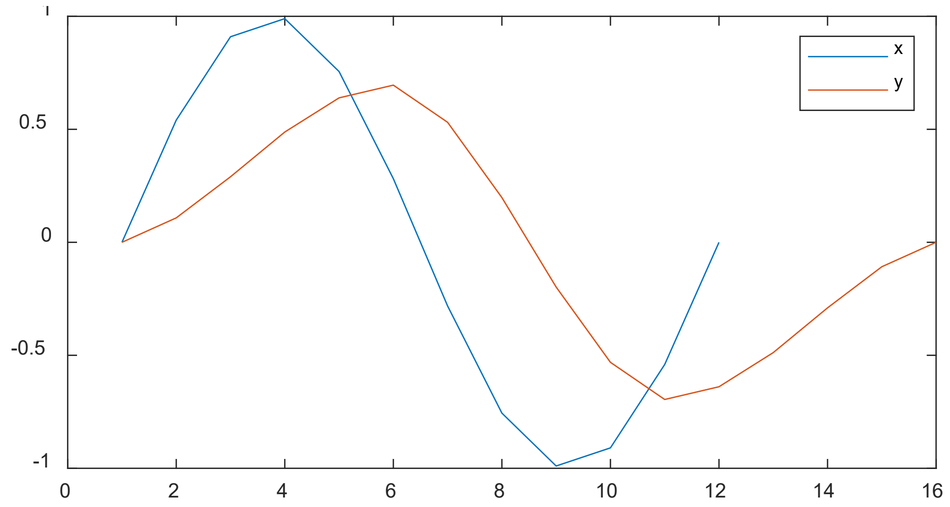
Operation: convolution
Y = X*H

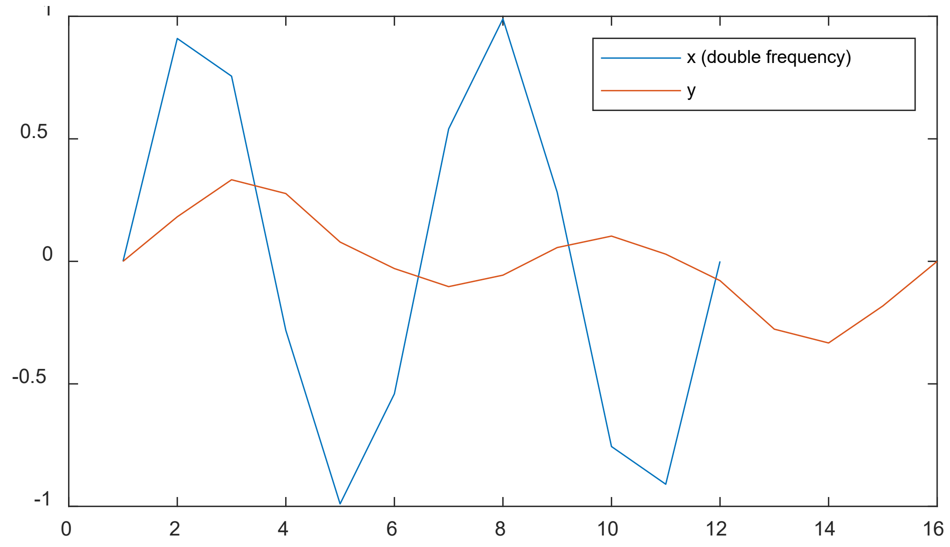
Output amplitude reduce!
Because H is a low pass filter!



Group delay

You also observe a delay?





We are talking about frequency domain here!

Group Delay

For most application, the magnitude is the main concern.

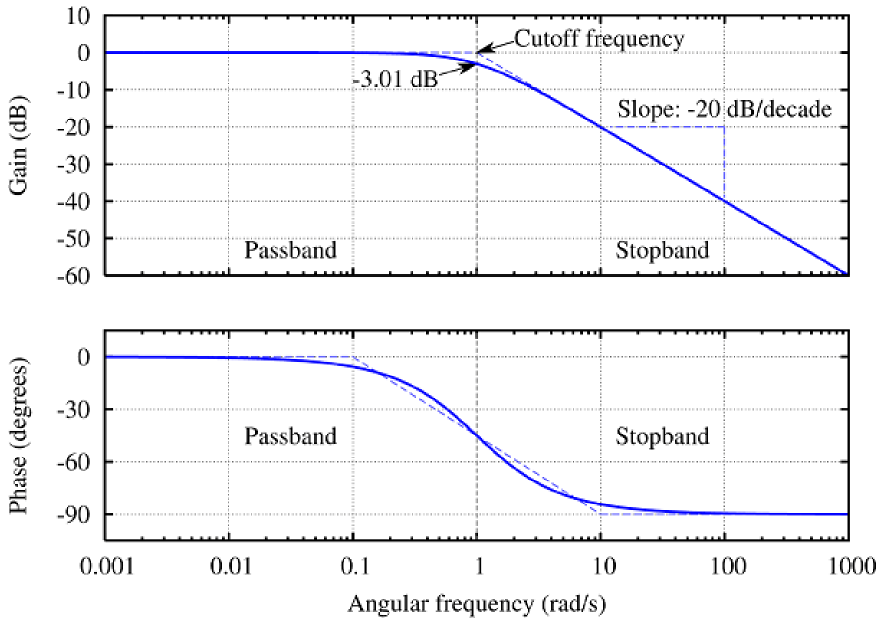
But the **phase characteristic** of a filter also influences the input-output behavior of the filter.
The group delay T_g is a measure of the time delay through the filter.
This value often depends on the frequency ω.

Group delay is defined as

$$T_g = -\frac{d\phi(\omega)}{d\omega}$$

where $\phi(\omega)$ is the filter phase [°].

To avoid pulse overshoot or damped oscillation, the filter must have a constant group delay.

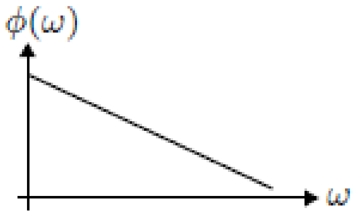


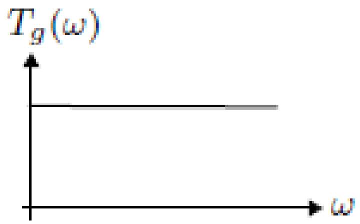
Group delay (Linear phase)

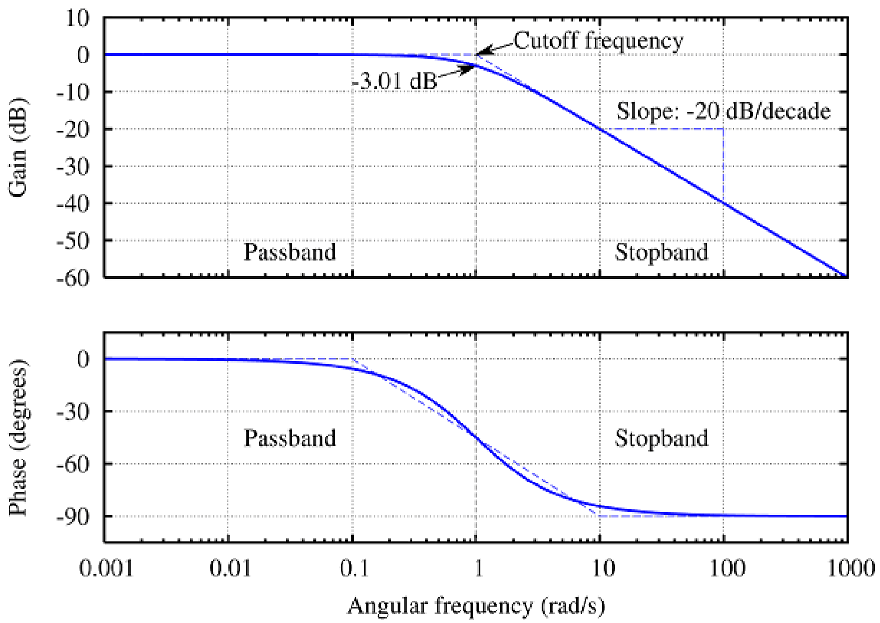
From the group delay, it can be determined whether the filter's step response has damped oscillation.

Let's consider a system with linear phase (constant group delay).

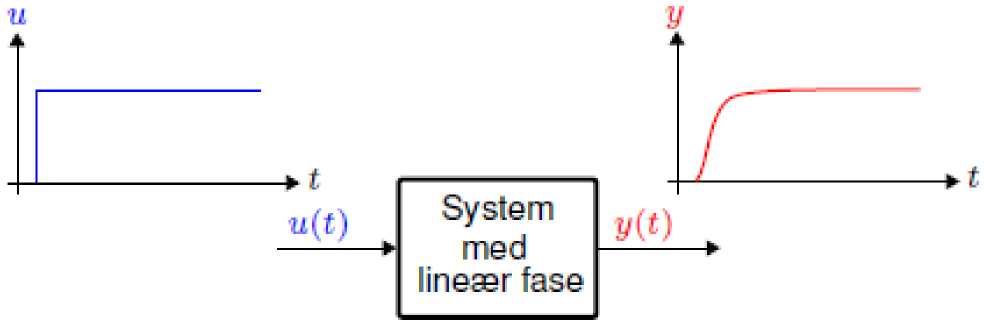
$$\boxed{T_g = -\frac{d\phi(\omega)}{d\omega}}$$





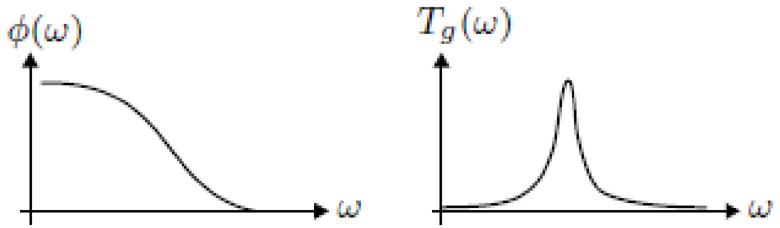


The step response of the system with linear phase does not have oscillation.

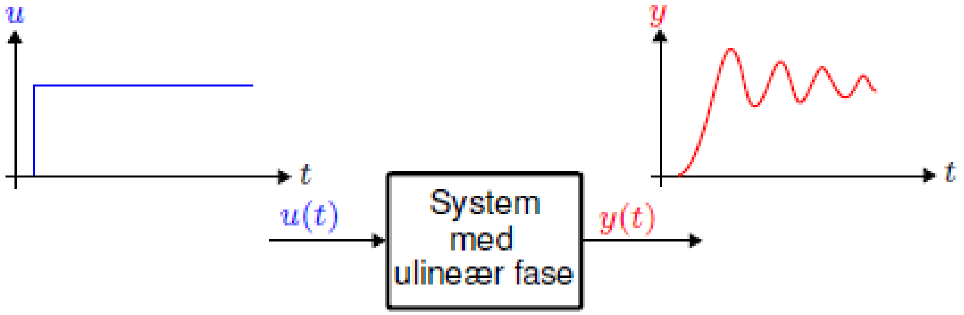


Group delay (non-linear phase)

For a system with non-linear phase



The step response of the system with nonlinear phase has oscillation.



Example – Group delay

```
% original signal
t = 0:.01:3;
% delay
tau_linear = 0;
tau_nonlinear = 0;

% different frequency component
comp1 = sin(2*pi*(t+tau_linear)+tau_nonlinear);
comp2 = sin(3*2*pi*(t+tau_linear)+tau_nonlinear)/3;
comp3 = sin(5*2*pi*(t+tau_linear)+tau_nonlinear)/5;
comp4 = sin(7*2*pi*(t+tau_linear)+tau_nonlinear)/7;
comp5 = sin(9*2*pi*(t+tau_linear)+tau_nonlinear)/9;
y = comp1 + comp2+ comp3+ comp4+comp5;
plot(t,y);
hold on
```

```
% linear group delay
tau_linear = 0.2;
tau_nonlinear = 0;

% different frequency component
comp1 = sin(2*pi*(t+tau_linear)+tau_nonlinear);
comp2 =
sin(3*2*pi*(t+tau_linear)+tau_nonlinear)/3;
comp3 =
sin(5*2*pi*(t+tau_linear)+tau_nonlinear)/5;
comp4 =
sin(7*2*pi*(t+tau_linear)+tau_nonlinear)/7;
comp5 =
sin(9*2*pi*(t+tau_linear)+tau_nonlinear)/9;
y = comp1 + comp2+ comp3+ comp4+comp5;
plot(t,y);
```

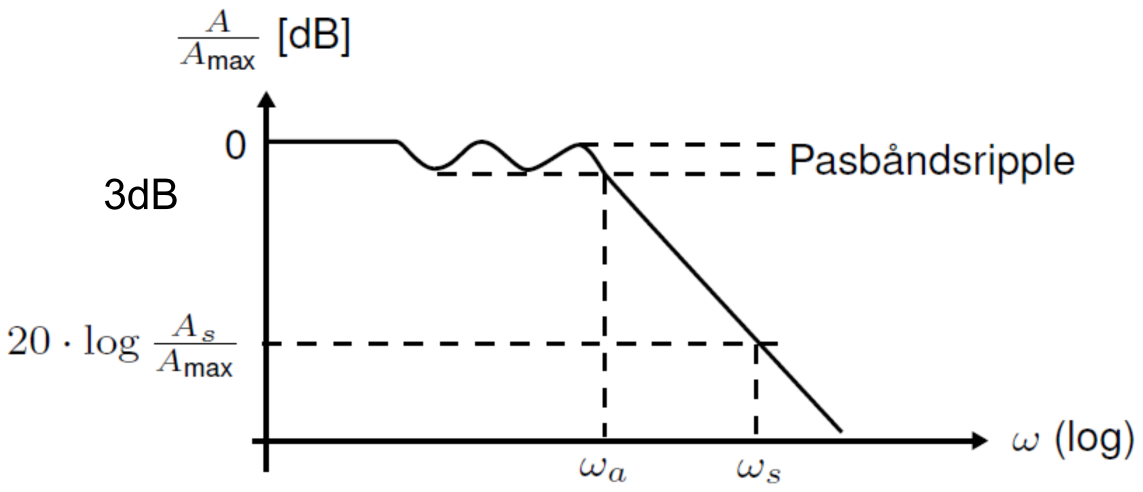
```
% non-linear group delay
tau_linear = 0;
tau_nonlinear = 0.2;

% different frequency component
comp1 =
sin(2*pi*(t+tau_linear)+tau_nonlinear);
comp2 =
sin(3*2*pi*(t+tau_linear)+tau_nonlinear)/3;
comp3 =
sin(5*2*pi*(t+tau_linear)+tau_nonlinear)/5;
comp4 =
sin(7*2*pi*(t+tau_linear)+tau_nonlinear)/7;
comp5 =
sin(9*2*pi*(t+tau_linear)+tau_nonlinear)/9;
y = comp1 + comp2+ comp3+
comp4+comp5;
plot(t,y);
```

Filter specification

A filter specification consists of minimum requirements for the amplitude characteristic of the filter (there may also be requirements for the phase).
For a low-pass filter, these requirements can be the followings:

1. A **cutoff frequency ω_a** which indicates the upper limit of the passband.
2. A **stopband frequency ω_s** at which a **stopband attenuation A_s** is specified.
3. Information on permissible gain variation in the passband.

$$\boxed{\begin{aligned} dB &= 10\log_{10}\frac{P_2}{P_1} \\ dB &= 20\log_{10}\frac{A_2}{A_1} \end{aligned}}$$



SDU

Filter transfer functions

In practice, the ideal filters cannot be realized, but can be approximated by using various filter function types.

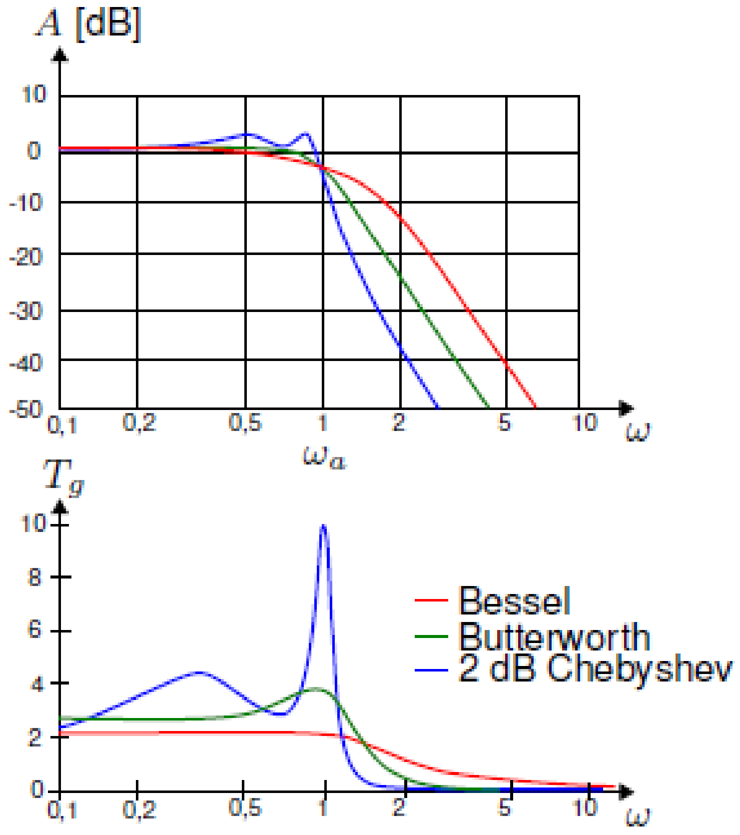
We look at three filter function types

1. Butterworth
2. Chebyshev
3. Bessel

Filter transfer functions

When choosing the filter function for a given application, we are interested in the following characteristics

1. Constant gain in the passband
2. High attenuation after the cutoff frequency
3. Linear phase



SDU

Butterworth Filter

The amplitude response of a N^{th} order Butterworth filter is

$$|H(j\omega)| = \frac{1}{\sqrt{1 + (\omega/\omega_a)^{2N}}}$$

where ω_a is the cutoff frequency [rad/s].

Butterworth Filter

A Butterworth filter has the following properties:

1. Optimal in terms of **constant gain in the passband**.
2. Has attenuation of 3 dB at the cutoff frequency and then the filter's gain drops rapidly by 20 dB/decade.
3. The **phase** of the filter is **not constant** in the passband, which causes ripple at step input.

All pole pairs of a Butterworth filter have a natural resonant frequency ω_n, which equals to the cutoff frequency ω_a.

Chebyshev filter

The squared amplitude response of an N^th order Chebyshev Filter is

$$|H(\omega)|^2 = \frac{1}{1+\epsilon^2 T_N^2(\omega/\omega_a)}$$

where $T_N(\omega)$ is the Chebyshev polynomial of degree N given by

$$T_N(\omega) = \begin{cases} \cos(N\cos^{-1}(\omega)) & |\omega| \le 1 \\ \cosh(N\cosh^{-1}(\omega)) & |\omega| > 1 \end{cases}$$

$$\epsilon = \sqrt{10^{\delta/10} - 1}$$

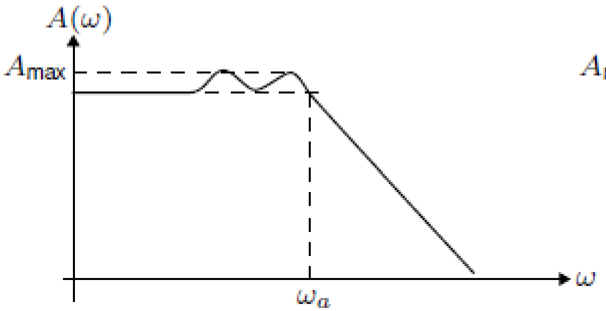
where δ is the size of the passband ripple in dB

Chebyshev filter

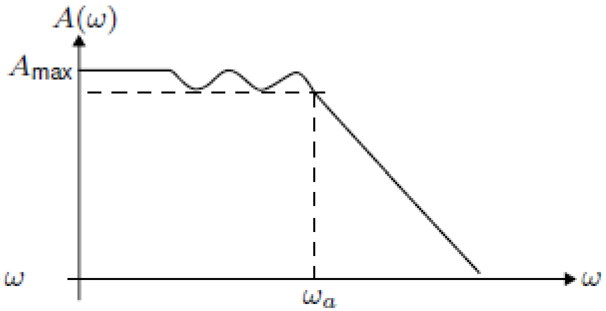
A Chebyshev filter has the following characteristics:

1. It has varying gain in the passband – **the size of the passband ripple can be chosen**.
2. The gain drops quickly around the cutoff frequency.
3. The **phase** of the filter is **not constant** in the passband, which leads to ripple with a step input.

The DC gain of a Chebyshev low-pass filter is not the maximum gain of the filter if the number of poles is even.



Even number of poles



Odd number of poles

Bessel filters

A Bessel filter has the following properties:

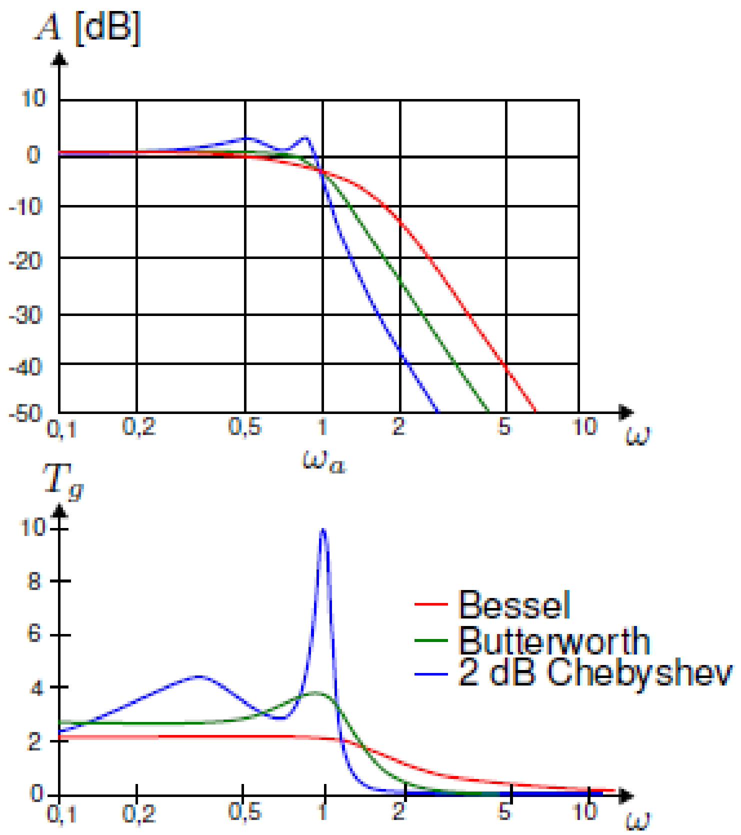
1. Has **no ripple** in the passband, but the amplitude is not as constant as with the Butterworth filter.
2. **Attenuation slowly**. The amplitude characteristic of the filter is as for the first order filter for the first 6 dB of attenuation regardless of the number of poles.
3. The **phase is almost linear** with the frequency within the passband.

The transfer function of a N^{th} order Bessel filter is

$$H_N(s) = \frac{b_0}{s^N + b_{N-1}s^{N-1} + \cdots + b_1 s + b_0}$$

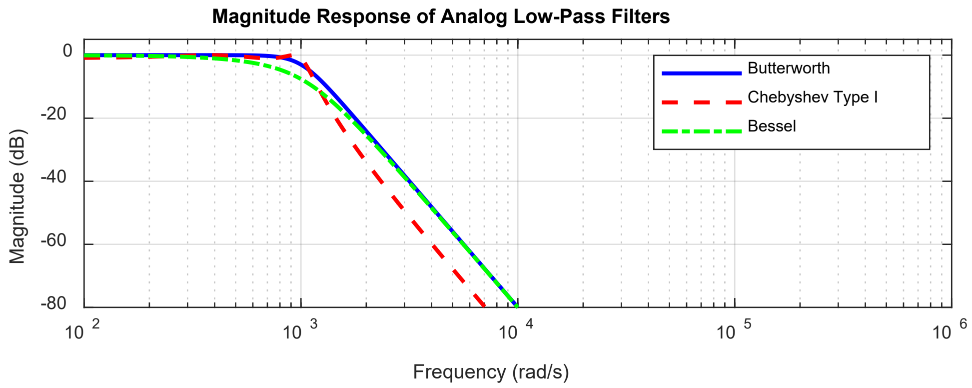
where

$$b_k = \frac{(2N-k)!}{2^{N-k}k!(N-k)!}$$



Matlab

```
% Define filter specifications
N = 4;                % Filter order
wa = 1000;            % Cutoff frequency in rad/s

% Butterworth Filter
[num_butter, den_butter] = butter(N, wa, 's');  % Butterworth filter

% Chebyshev Type I Filter
Rp = 1; % Ripple in the passband (dB)
[num_cheby, den_cheby] = cheby1(N, Rp, wa);  % Chebyshev Type I prototype filter

% Bessel Filter
[num_bessel, den_bessel] = besself(N, wa);  % Bessel filter
```



Matlab analog filter prototype

Please check the following functions, which construct normalized filter.

[z,p,k] = buttap(n)

[z,p,k] = cheb1ap(n,Rp)

[z,p,k] = besselap(n)

These functions return zeros and poles and gains.
You may need to use

```
[num,den] = zp2tf(z,p,k);
```

To convert (z,p,k) to transfer function's numerater and denominater.

Matlab

To see the frequency response of a filter:

```
% for analog filter, magnitude (not dB)
freqs()
% for digital filter, magnitude (not dB)
freqz()

% transfer function based bode plot(dB)
sys = tf(numerator, denominator);
bode(sys);
```

Simulating an input signal passing through this filter:

```
% Define time vector
t = 0:0.0001:0.01;
% Generate input signal 1000Hz
x = sin(2*pi*1000* t);
% output after passing through the filter
y = lsim(sys, x, t);

% plot both input & output
plot(t, x, 'b', t, y, 'r');
```

Selection of the filter's order

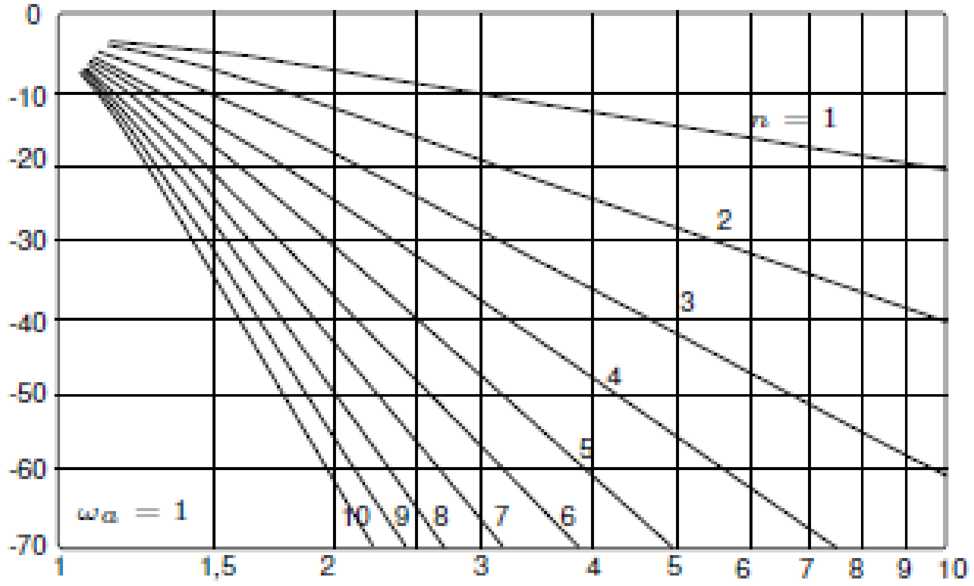
The filter’s order can be found by reading the amplitude characteristic for a given filter function, and comparing this with requirements for attenuation at the stopband frequency.

The gain is normalized

$$Y = 20 \log \frac{A}{A_{\text{max}}}$$

The frequency normalization

$$X = \frac{\omega}{\omega_a}$$



[num,den] = zp2tf(z,p,k);

Example

Find the number of poles for a low-pass filter with the requirements:

→ Cutoff frequency $f_a = 2$ kHz
→ Stopband frequency $f_s = 8$ kHz
→ Stopband attenuation compared to A_{max} of at least 60 dB

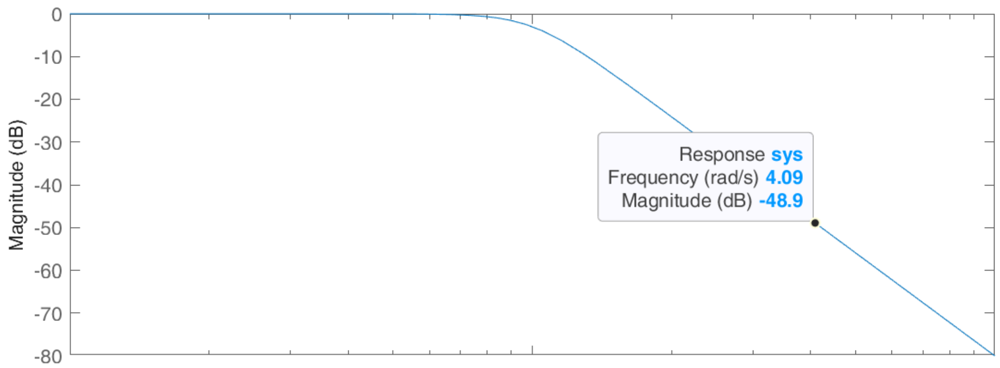
The normalized stopband frequency is $\frac{f_s}{f_a} = 4$

Then ask Matlab
For example, using Butterworth filter

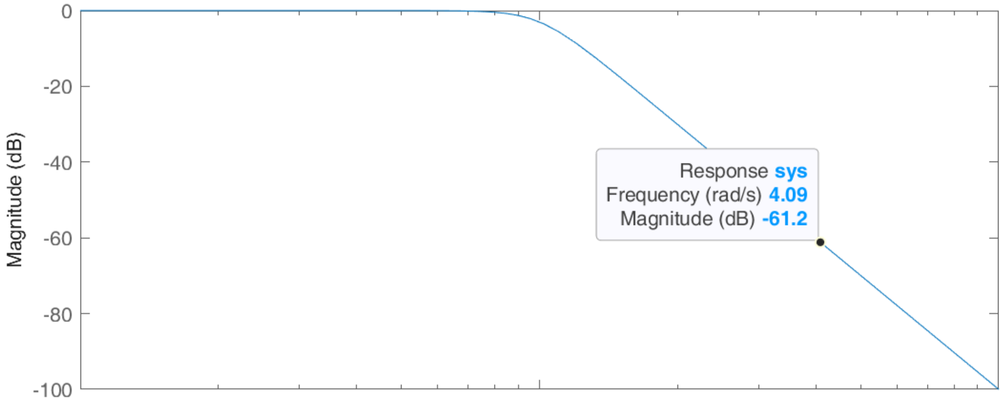
Notice: it is still an analog filter!

```
% Define filter specifications
N = 4;  % Filter order (try different value)
% Butterworth Filter
[z,p,k] = buttap(N);
[num,den] = zp2tf(z,p,k);
% Check frequency response
sys = tf(num,den);
bode(sys);
```

Order of 4



Order of 5



SDU

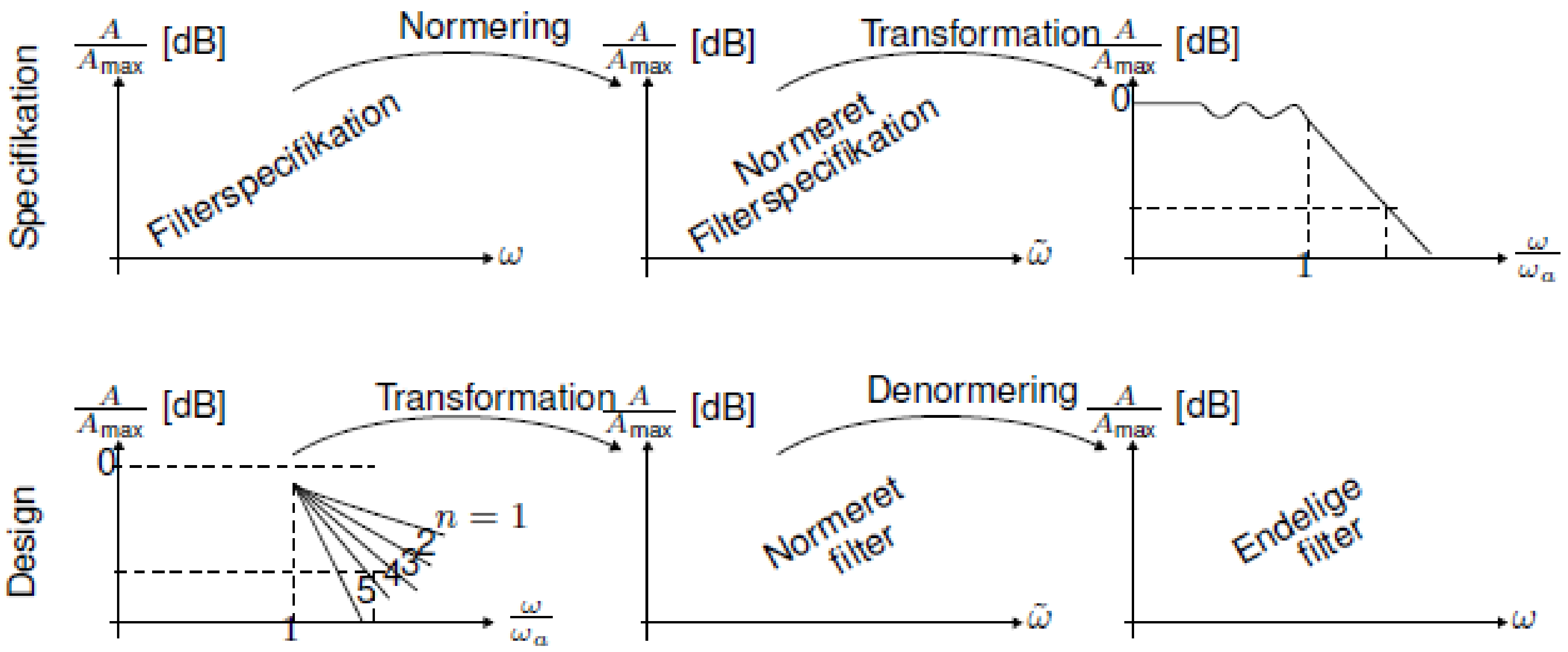
Construct a filter

For **digital filters**, we want it with less order:
→ Less complexity of realization
→ Less computation
→ Higher precision

Digital filters are implemented using numerical computations. The transfer function can be affected by the precision limits of the system (e.g., due to rounding errors).

$$H(s) = \frac{A_0}{s^n + B_{n-1}s^{n-1} + \cdots + B_1 s + B_0}$$

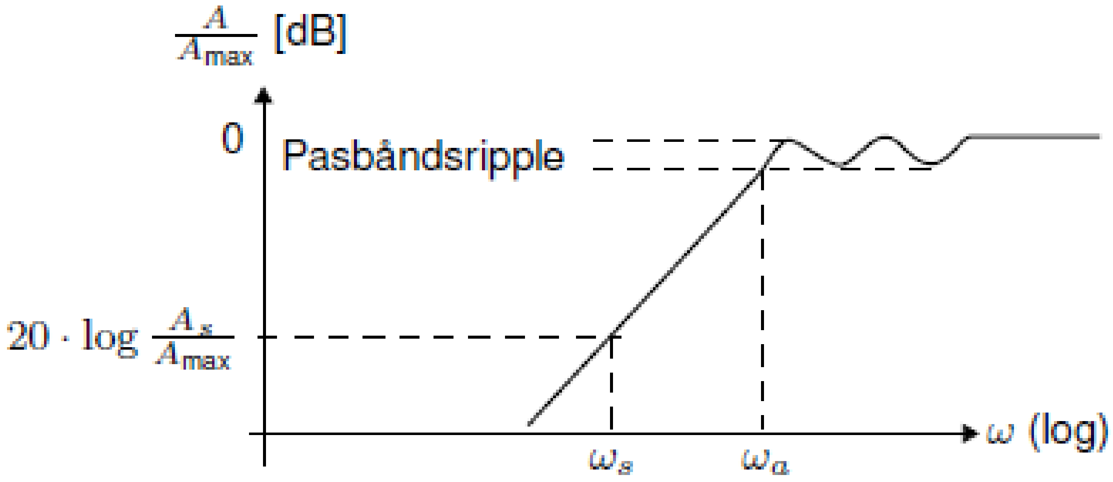
Filter transformation



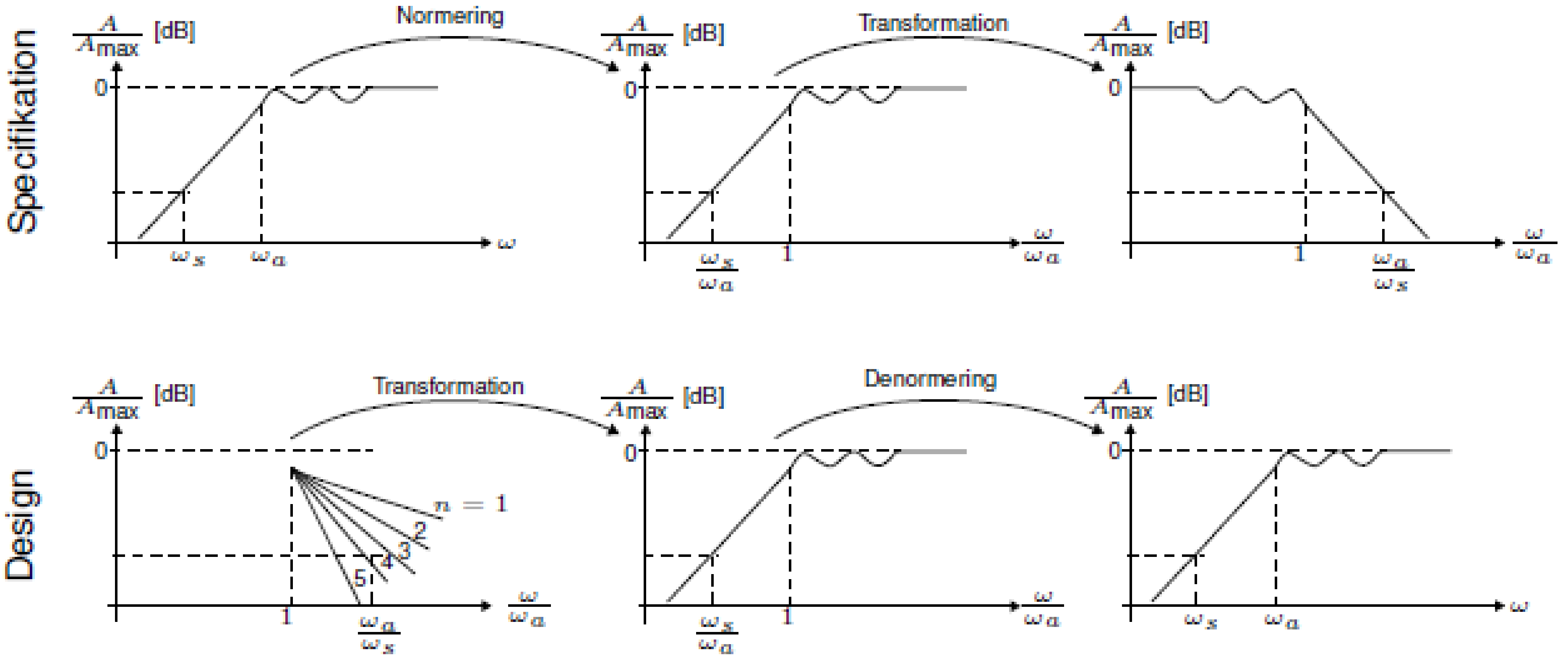
Low-pass to high-pass transformation

High-pass filters can be designed from normalized prototype low-pass filters based on the following specification:

→ Filter function (Bessel, Butterworth, Chebyshev)
→ Cutoff frequency of the filter ω_a
→ Filter's stopband frequency ω_s
→ Filter's stopband attenuation A_s at the stopband frequency ω_s



Overview of the design procedure



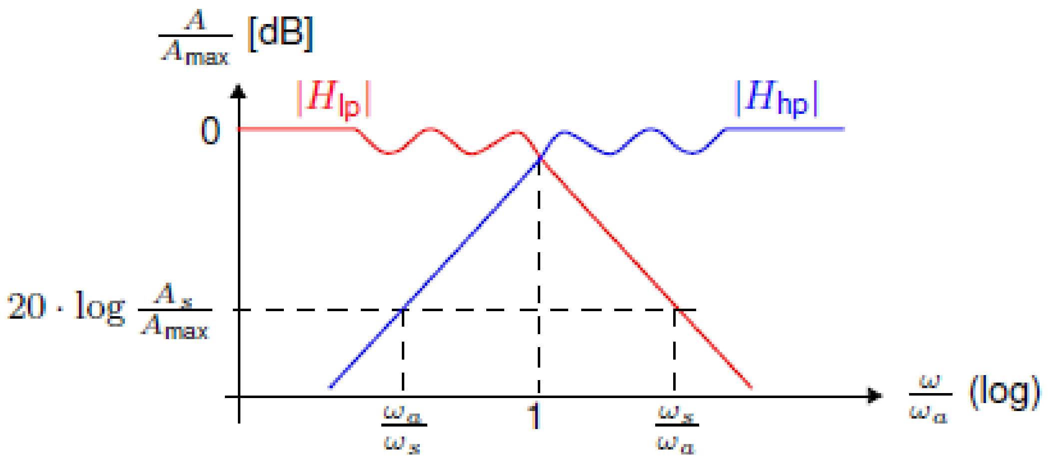
SDU

Filter transformation

A low-pass filter can be transformed into a high-pass filter by

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

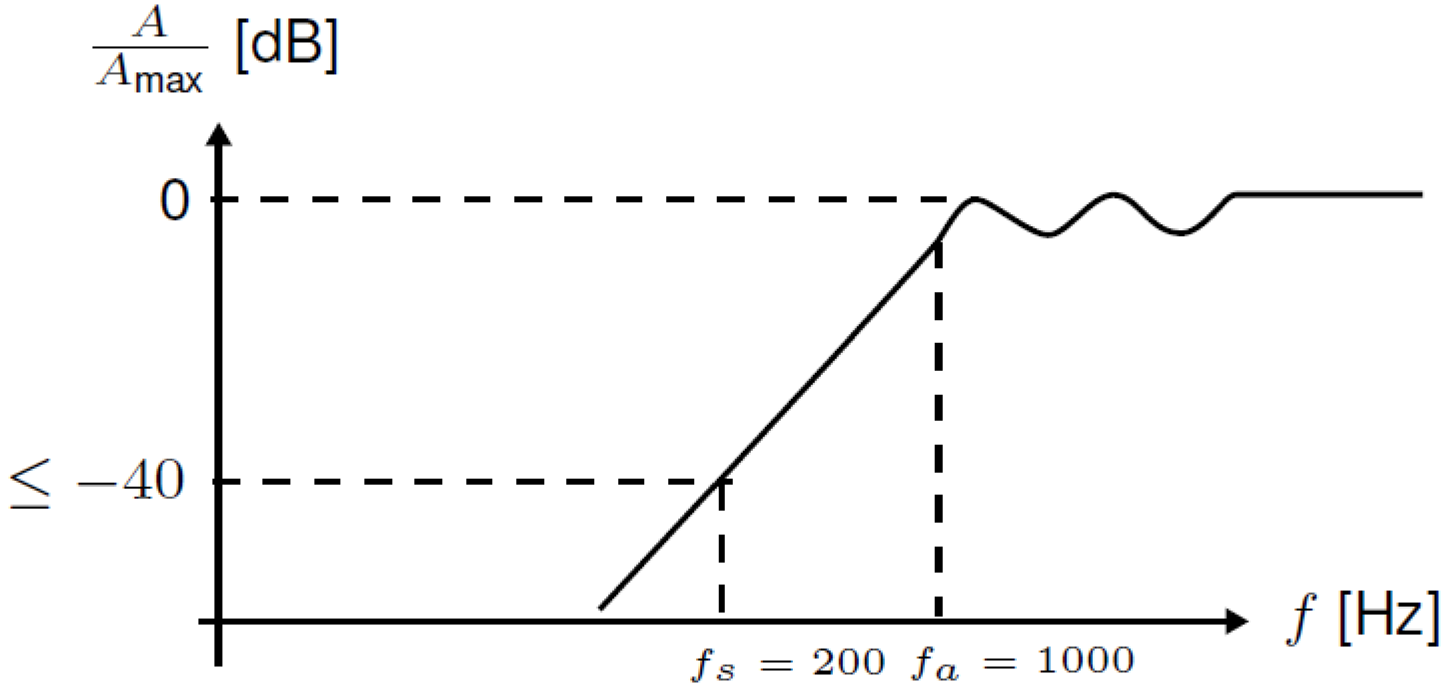
The transformation corresponds to mirroring the frequency axis by 1, as shown here



Example: Low-pass to high-pass transformation

Step 1: Filter specification

Design a 1 dB Chebyshev high-pass filter that satisfies the following amplitude characteristic.



Example: Low-pass to high-pass transformation

Step 2: Transformation of specification

To design the filter, then normalized high-pass filter, i.e. the standardized stopband frequency is calculated for

$$\frac{\omega_s}{\omega_a} = \frac{f_s}{f_a} = \frac{1}{5}$$

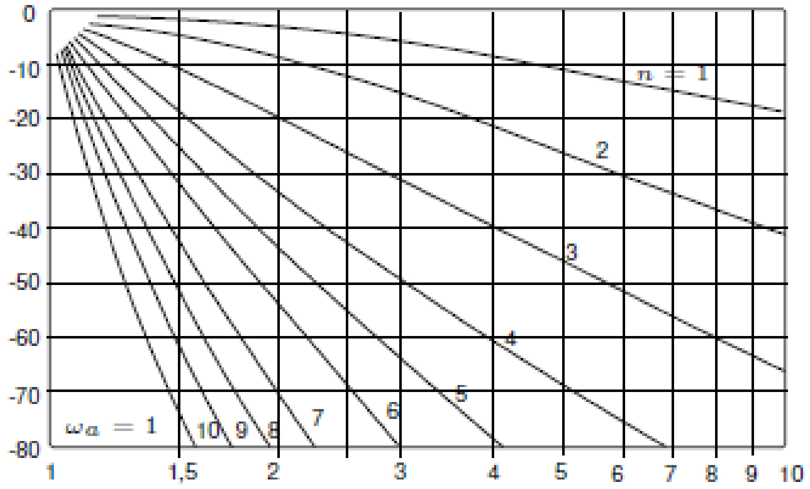
Finally, the stopband frequency for the normalized low-pass filter used for the design is calculated.

$$\frac{\omega_a}{\omega_s} = 5$$

Example: Low-pass to high-pass transformation

Step 3: Choose filter order

The filter is designed by choosing the filter's order. This is done based on the amplitude characteristics of 1 dB Chebyshev low-pass filters. It is a requirement that the gain is below -40 dB at a frequency of 5.



The minimum power for the filter that meets the requirement is 3 ($n = 3$).

SDU

Example: Low-pass to high-pass transformation

Step 4: Denormalization

The transfer function of the normalized **low-pass** filter can be calculated

$$H(s) = \frac{1}{(s^2 + 0,49417s + 0,99421)(s + 0,49417)}$$

The normalized **high-pass** filter thus becomes (replace s with $1/s$ in the above)

$$H(s) = \frac{s^3}{(1 + 0,49417s + 0,99421s^2)(1 + 0,49417s)}$$

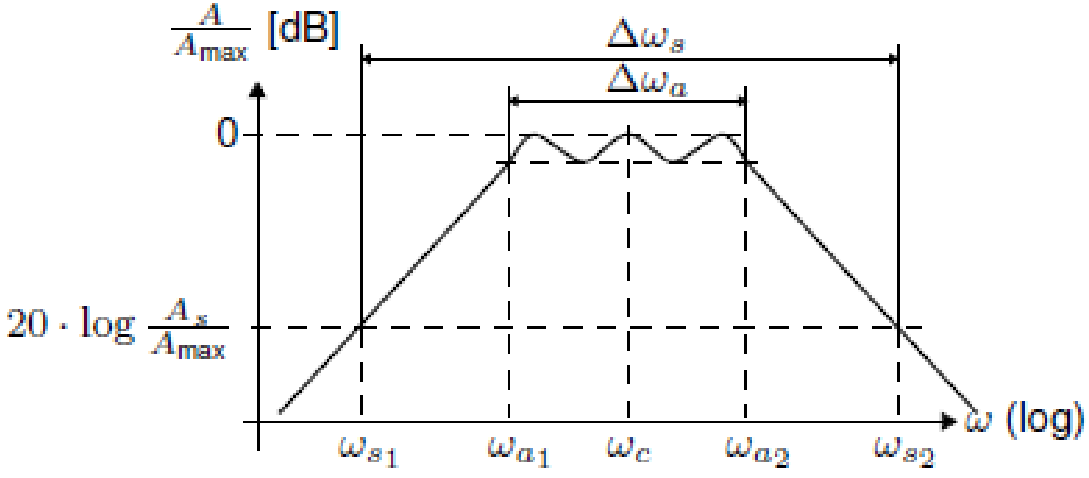
The transfer function of the denormalized **high-pass** filter is (replace s with s/ω_a in the above)

$$H(s) = \frac{s^3/\omega_a^3}{(1 + 0,49417/\omega_a s + 0,99421/\omega_a^2 s^2)(1 + 0,49417/\omega_a s)}$$

Low-pass to band-pass transformation

Bandpass filters can be designed from normalized prototype lowpass filters based on the following specification

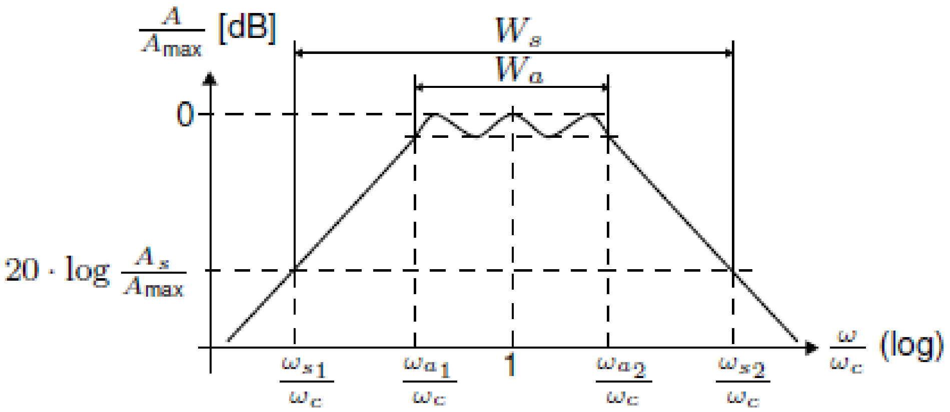
→ The filter function (Bessel, Butterworth, Chebyshev)
→ The center frequency of the filter ω_c
→ The Passband bandwidth $\Delta\omega_a$
→ The Stopband bandwidth $\Delta\omega_s$
→ The filter's stopband attenuation A_s



Low-pass to band-pass transformation

The bandpass filter is normalized to the center frequency ω_c, which is midway between ω_{a1} and ω_{a2} on a logarithmic axis, i.e.

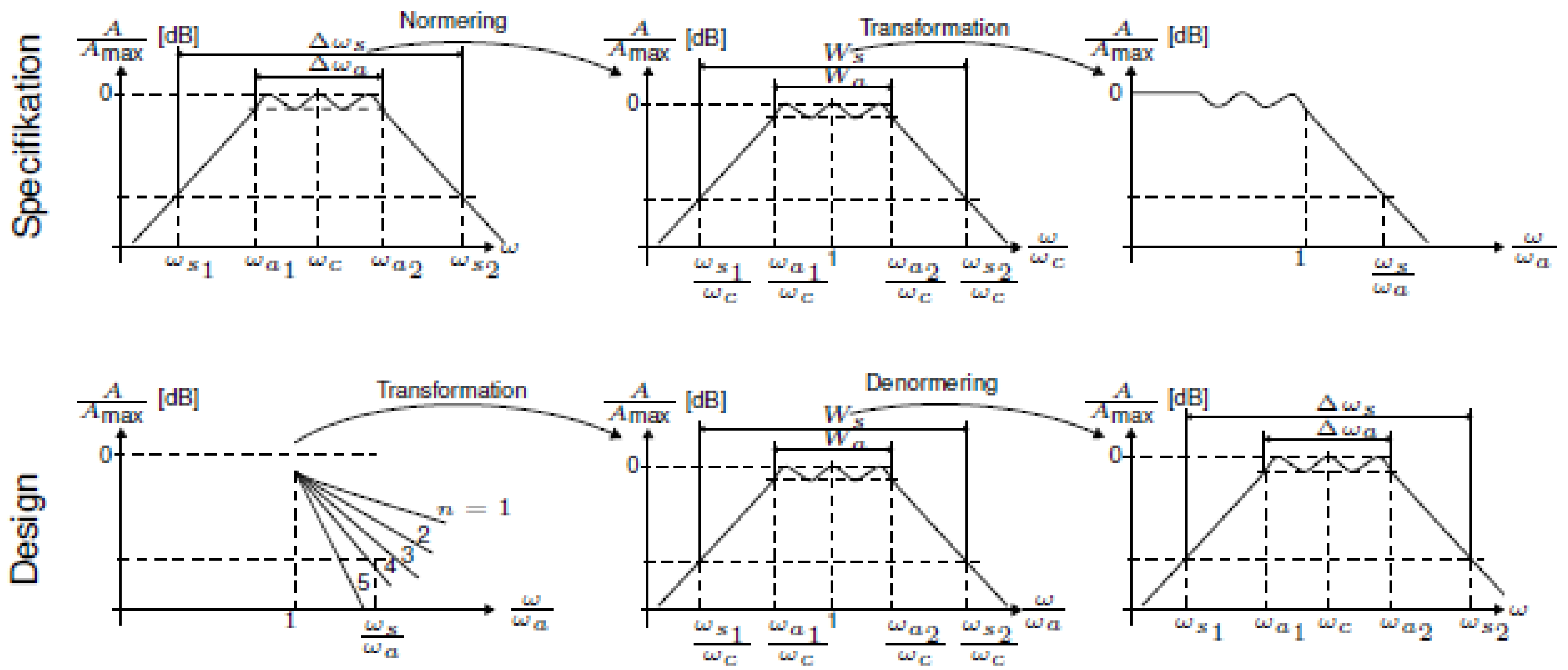
$$\log \omega_c = \frac{\log \omega_{a_1} + \log \omega_{a_2}}{2} = \log \sqrt{\omega_{a_1}\omega_{a_2}}$$
$$\omega_c = \sqrt{\omega_{a_1}\omega_{a_2}}$$



The form factor for a band-pass filter is defined as $F = \frac{W_s}{W_a}$

SDU

Overview of the design procedure

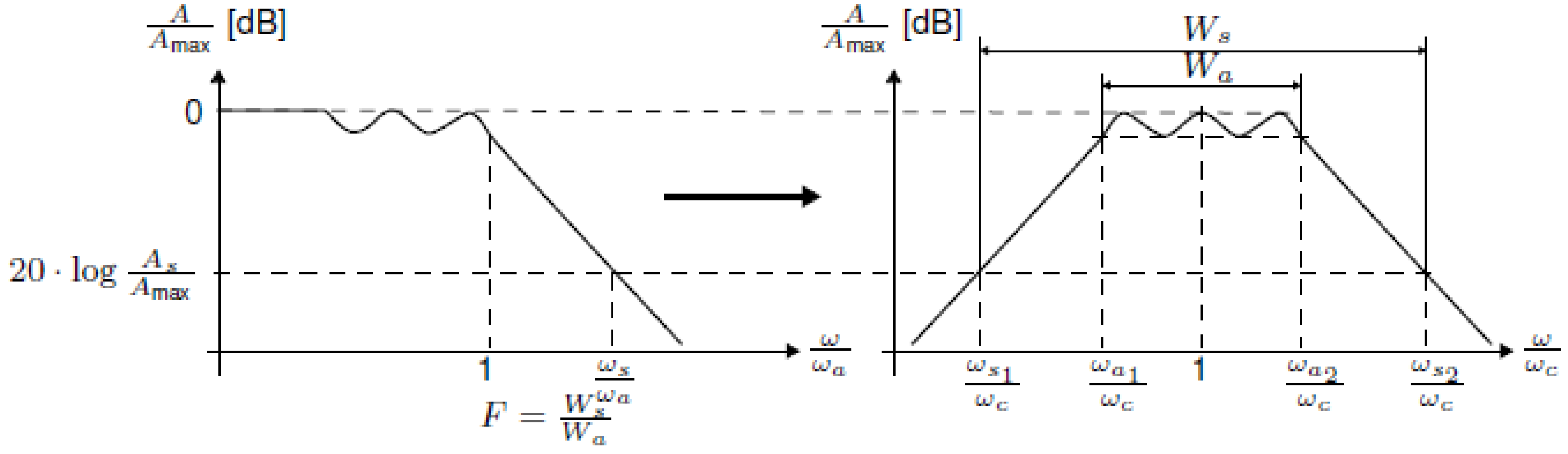


SDU

Filter transformation

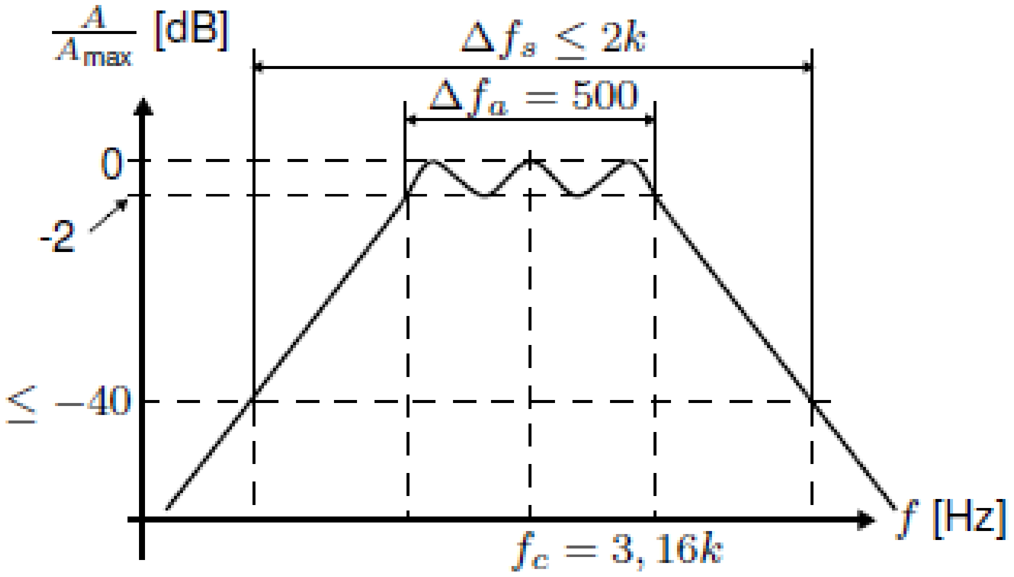
A low-pass filter can be transformed into a band-pass filter by

$$H_{\mathrm{bp}}(s) = H_{\mathrm{lp}}(\bar{s})\big|_{\bar{s}=\frac{1}{W_a}\left(s+\frac{1}{s}\right)}$$



Example: Filter specification

Design a 2 dB Chebyshev bandpass filter that satisfies the following amplitude characteristic.



Example: Low-pass to band-pass transformation

Step 1: Filter specification

To design the filter, the bandpass filter is normalized, i.e. the standardized stopband width and the standardized passband width are calculated.

$$W_a = \frac{\Delta f_a}{f_c} = 0,1582$$
$$W_s = \frac{\Delta f_s}{f_c} = 0,6329$$

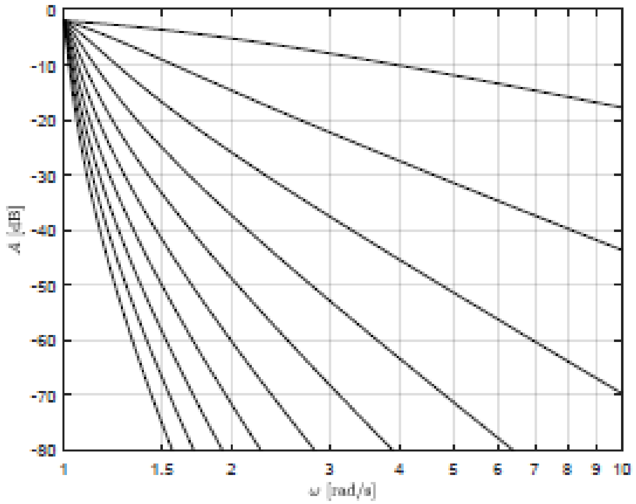
Finally, the form factor is calculated

$$F = \frac{\Delta f_s}{\Delta f_a} = 4$$

Example: Low-pass to band-pass transformation

Step 2: choose filter's order

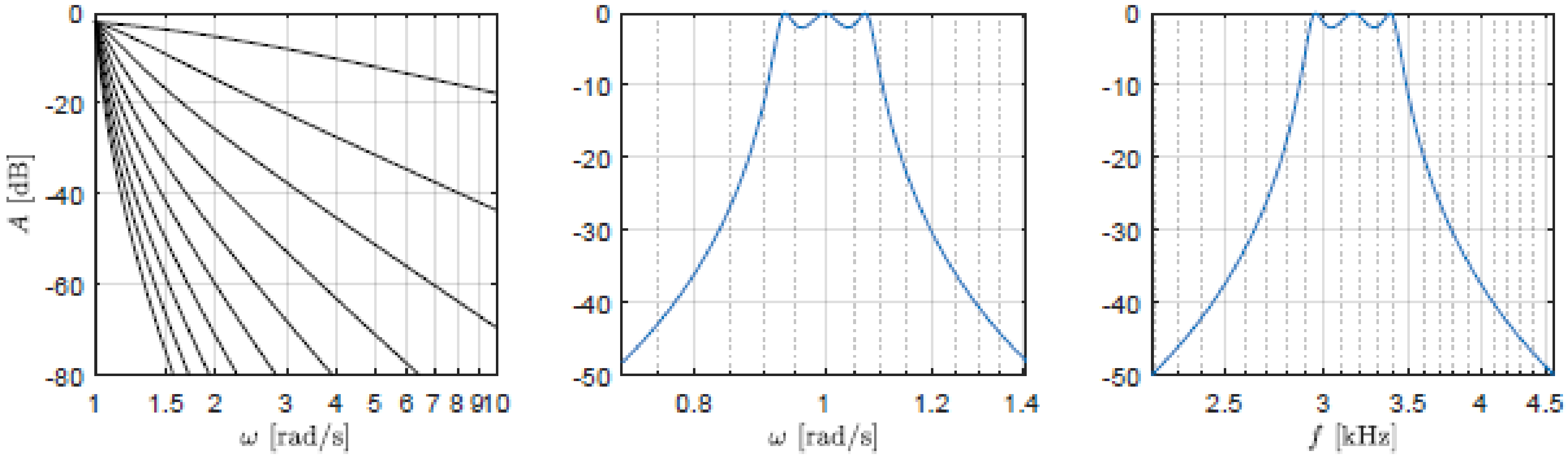
The filter's order is designed based on the amplitude characteristics of 2 dB Chebyshev low-pass filters. It is required that the gain is below -40 dB at a frequency of 4, since $F = 4$.



The minimum number for the filter's order that meets the requirement is 3 (n = 3).

Example: Low-pass to band-pass transformation

Step 3: Denormalization



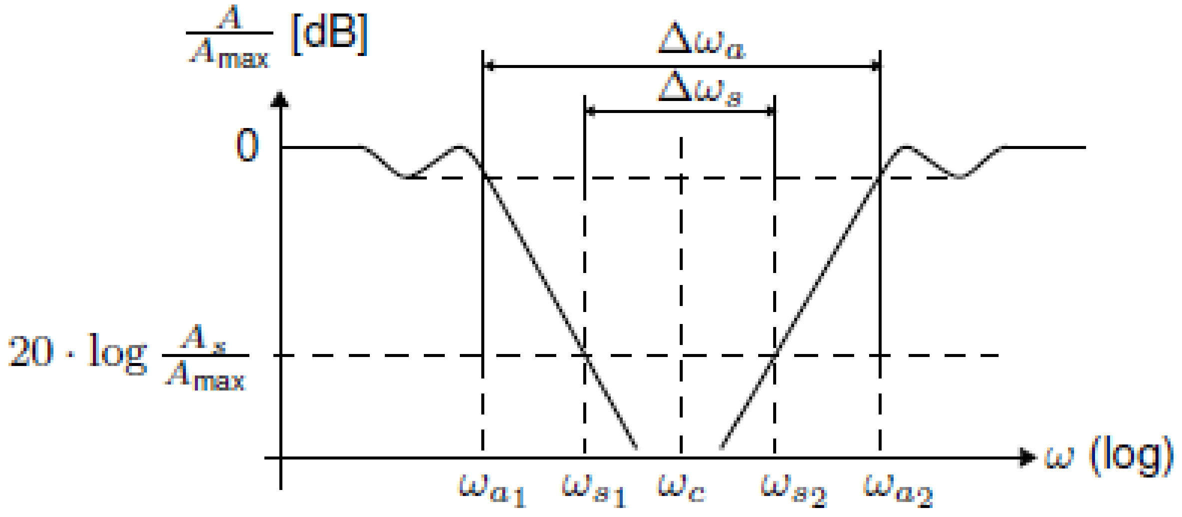
→ The normalized filter is made by replacing s with $\frac{1}{W_a}(s+\frac{1}{s})$ in the low-pass filter.

→ The denormalized bandpass filter is found by replacing s in the normalized bandpass filter with s/ω_c

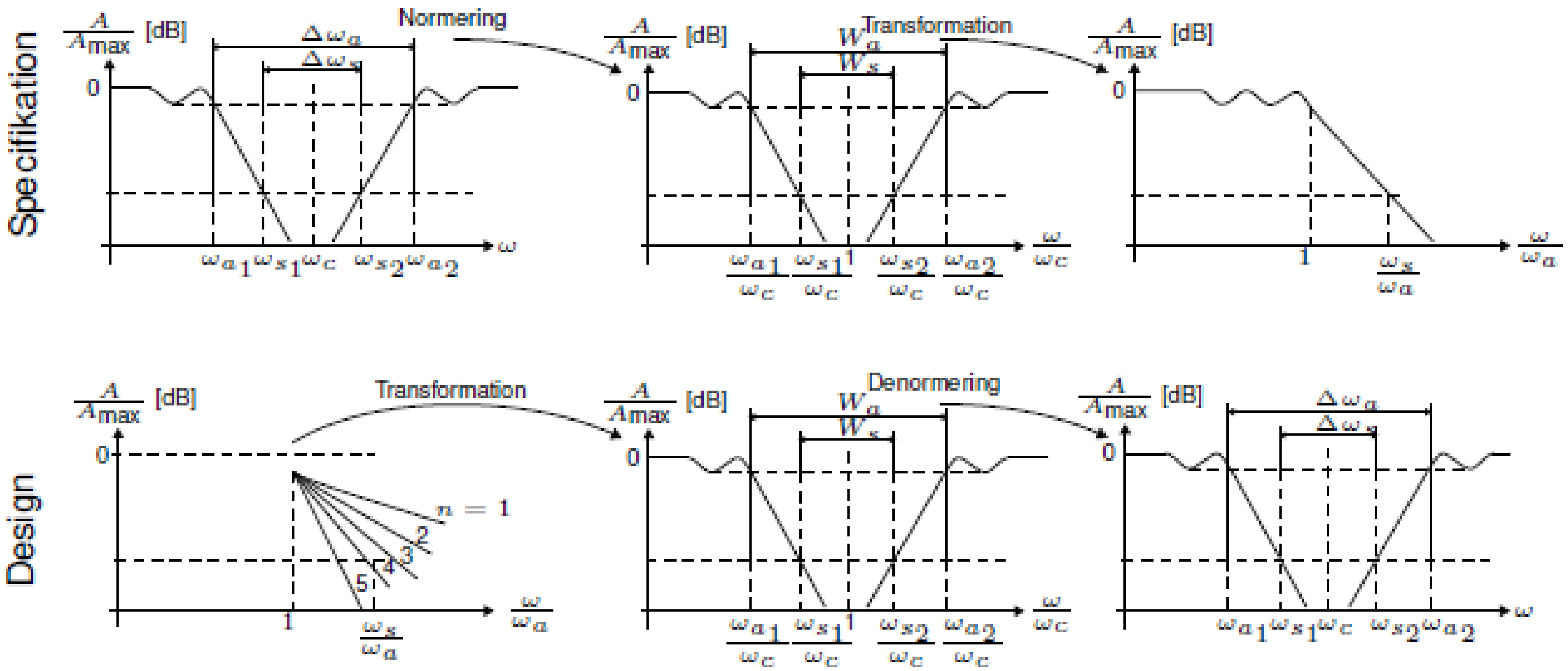
Low-pass to band-stop transformation

Bandstop filters can be designed from normalized prototype low-pass filters based on the following specification

→ The Filter function (Bessel, Butterworth, Chebyshev)
→ The center frequency of the filter ω_c
→ The Passband bandwidth $\Delta\omega_a$
→ The Stopband bandwidth $\Delta\omega_s$
→ The filter's stopband attenuation A_s



Overview of the design procedure



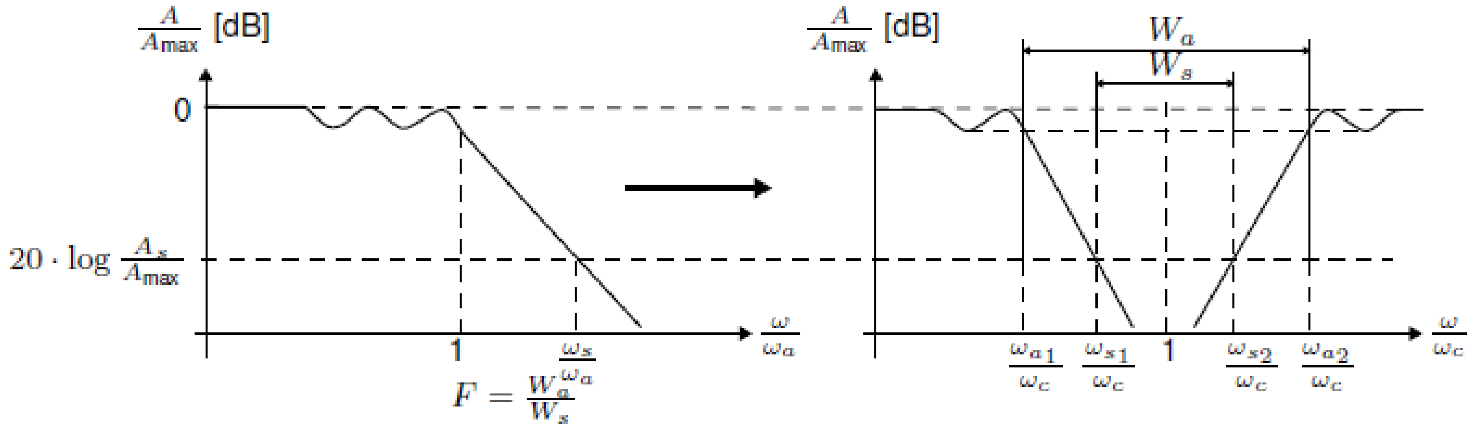
SDU

Filter transformation

A low-pass filter can be transformed into a band-stop filter by

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

The transformation is illustrated in the following figure.



The **form factor** of a bandstop filter is defined as $F = \frac{W_a}{W_s}$

Matlab example

Please check Matlab inherent functions for filter conversion:

```
lp2lp()  %Transform lowpass analog filters to lowpass
lp2hp()   %Transform lowpass analog filters to highpass
lp2bp()  %Transform lowpass analog filters to bandpass
lp2bs()  %Transform lowpass analog filters to bandstop
```

Exercise