FIR filter

Zhuoqi Cheng
zch@mmmi.sdu.dk
SDU Robotics

SDU

Recap: realization

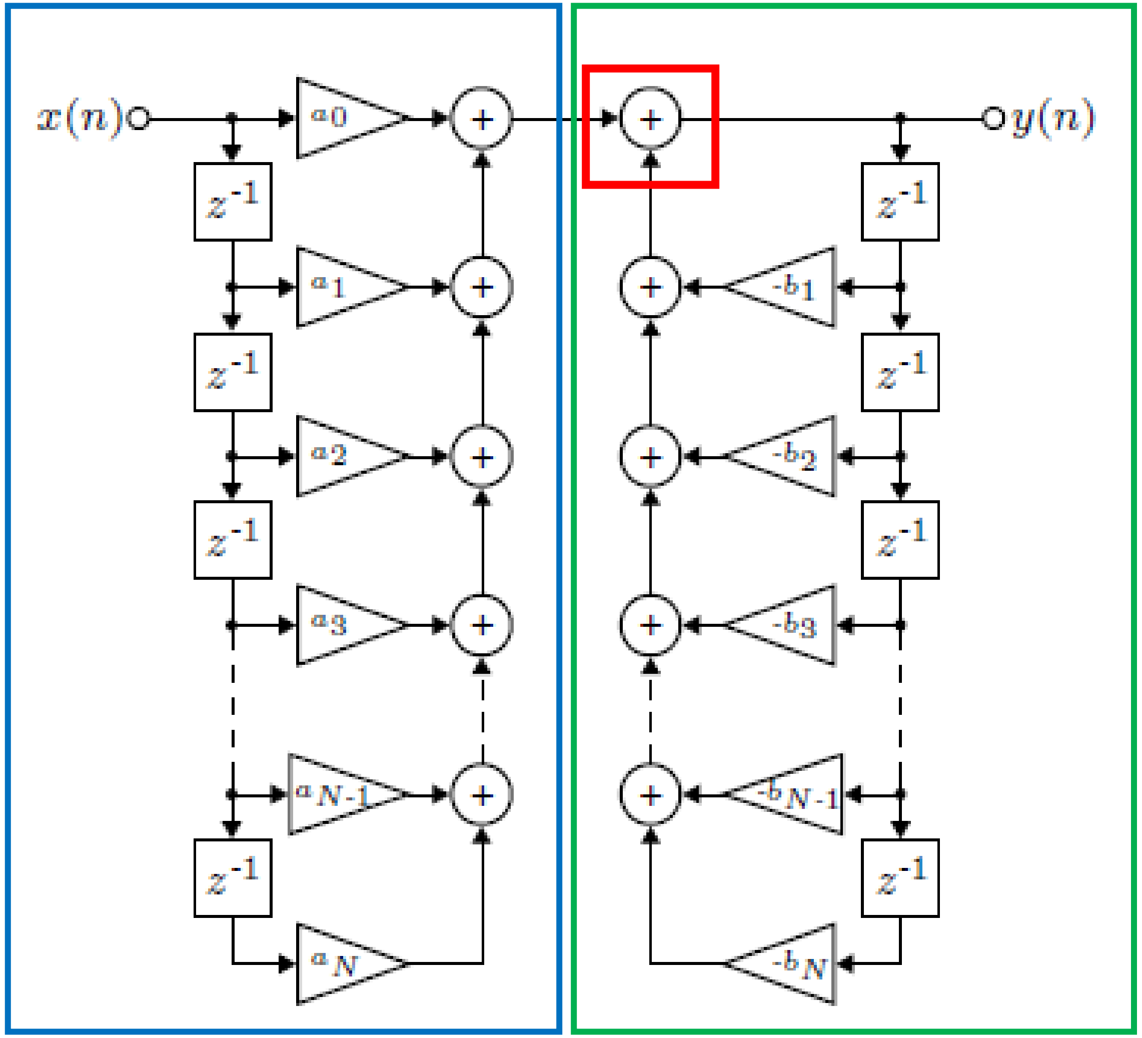
Transfer function

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{N} a_i\, z^{-i}}{1 + \sum_{i=1}^{N} b_i\, z^{-i}}$$

Difference equation

$$y(n) = \sum_{i=0}^{N} a_i x(n-i) - \sum_{i=1}^{N} b_i y(n-i)$$

In some cases, all $b_i = 0$.



SDU

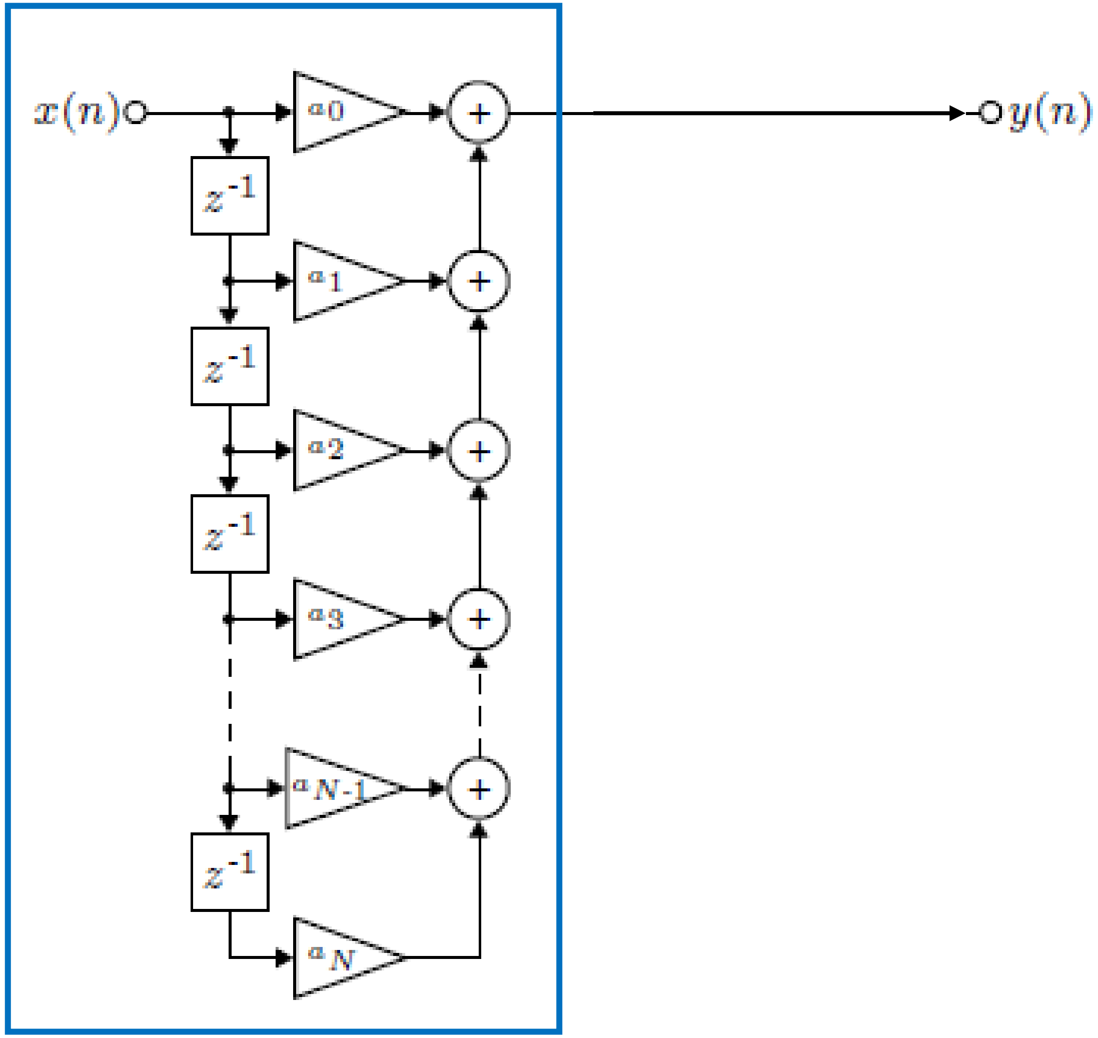
Recap: realization

Transfer function

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{N} a_i\, z^{-i}}{1}$$

Difference equation

$$y(n) = \sum_{i=0}^{N} a_i x(n-i)$$



Two categories of discrete-time filters

Digital filters are divided into two categories according to their impulse response:

Infinite Impulse Response filters (IIR filters)
- always have poles

$$y(n) = \sum_{i=0}^{N} a_i \, x(n-i) - \sum_{i=1}^{N} b_i \, y(n-i)$$

$$H(z) = \frac{\sum_{i=0}^{N} a_i \, z^{-i}}{1 + \sum_{i=1}^{N} b_i \, z^{-i}}$$

Finite Impulse Response filters (FIR filters)
- Has only zero points

$$y(n) = \sum_{i=0}^{N} a_i \, x(n-i)$$

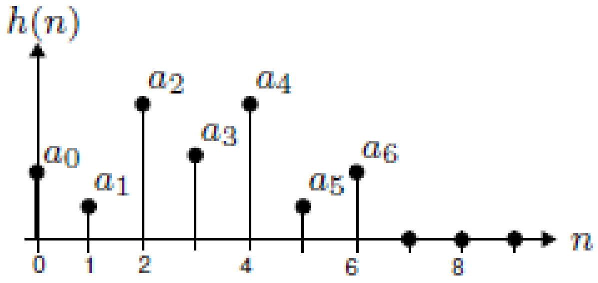
$$H(z) = \sum_{i=0}^{N} a_i \, z^{-i}$$

FIR definition

A ***finite impulse response filter*** **(FIR filter)** has a finite impulse response which defines the size of the filter. **An (N-1)th order discrete time FIR filter has N samples.** Its impulse response sequence is defined as follows

$$h(n) = \begin{cases} a_n, & 0 \leq n < N-1 \\ 0, & else \end{cases}$$

The filter thus has N coefficients a_n for $n = 0, 1, \ldots, N-1$



FIR filter Transfer function

The impulse response of the FIR filter can be written

$$h(n) = \sum_{i=0}^{N-1} a_i \delta(n-i)$$

z-transform of $h(n)$, we have

Recall:

$$\mathcal{Z}\{h(n)\} = \sum_{n=0}^{\infty} h(n) z^{-n}$$

$$H(z) = \mathcal{Z}\{h(n)\} = \sum_{i=0}^{N-1} a_i z^{-i}$$

Output:

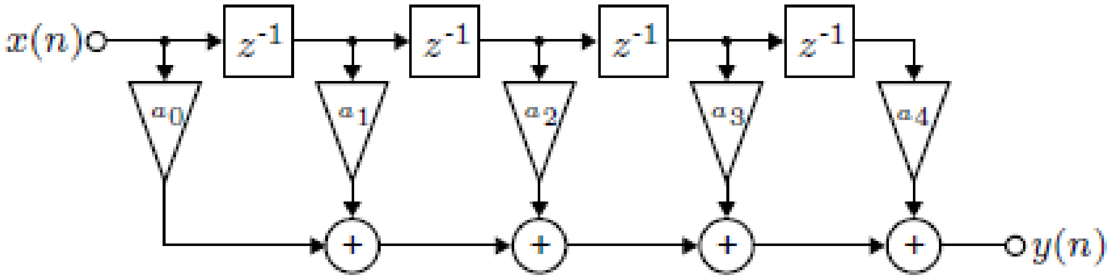
$$Y(z) = X(z)H(z) = X(z) \sum_{i=0}^{N-1} a_i z^{-i}$$

FIR filter realization structure

$$Y(z) = X(z) \sum_{i=0}^{N-1} a_i z^{-i}$$

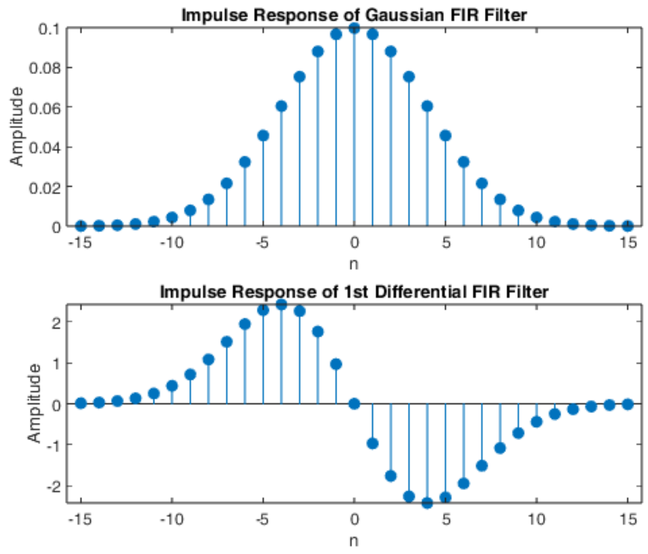
Then we do inverse z-transform

$$y(n) = \sum_{i=0}^{N-1} a_i x(n-i)$$



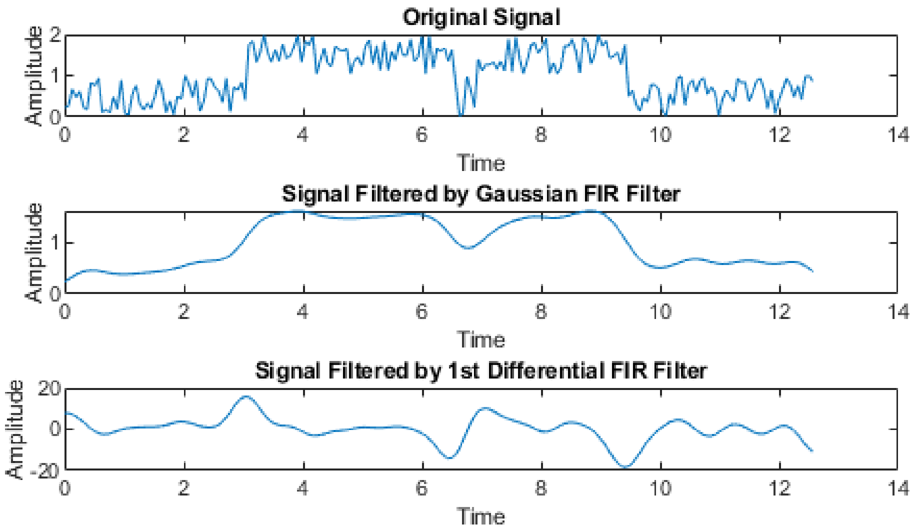
Matlab function:
conv(input_sig, fir_filter, ‘same’);

```
N = 31; % Length of the FIR filter (odd number for symmetry)
sigma = 4; % Standard deviation for Gaussian filter
n = -(N-1)/2 : (N-1)/2;

% 1. Gaussian Filter
h_gaussian = exp(-(n.^2) / (2 * sigma^2)); % Gaussian function
h_gaussian = h_gaussian / sum(h_gaussian); % Normalize

% 2. First Differential Filter
h_diff = -n .* exp(-(n.^2) / (2 * sigma^2)); % Derivative of Gaussian
h_diff = h_diff - mean(h_diff); % Normalize
```




```
% Input signal
t = linspace(0, 4*pi, 200);
signal = rand(1,200);
signal(50:150) = signal(50:150) + 1;
signal(105:110) = signal(105:110) - 1;
```

```
% Filtered Signals
signal_gaussian = conv(signal, h_gaussian, 'same');
signal_diff = conv(signal, h_diff, 'same’);

subplot(3,1,1);
plot(t, signal);
title('Original Signal’); xlabel('Time’); ylabel('Amplitude');

subplot(3,1,2);
plot(t, signal_gaussian);
title('Signal Filtered by Gaussian FIR Filter’); xlabel('Time’); ylabel('Amplitude');

subplot(3,1,3);
plot(t, signal_diff);
title('Signal Filtered by 1st Differential FIR Filter’); xlabel('Time’); ylabel('Amplitude');
```



SDU

Poles and zeros

For FIR filter

$$H(z) = \sum_{i=0}^{N-1} a_i z^{-i}$$

The transfer function can be re-written in positive powers as

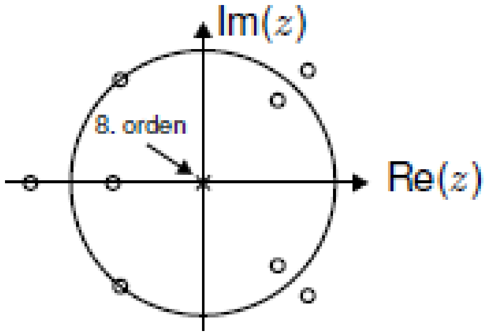
$$H(z) = \frac{\sum_{i=0}^{N-1} a_i z^{N-1-i}}{z^{N-1}}$$

from which you can see that

$H(z)$ has $\boldsymbol{N-1}$ **poles at the origin** of the z-plane and $\boldsymbol{N-1}$ **zeros**.

Poles and zeros

The following is an example of pole-zero diagram for an 8^{th}-order FIR filter

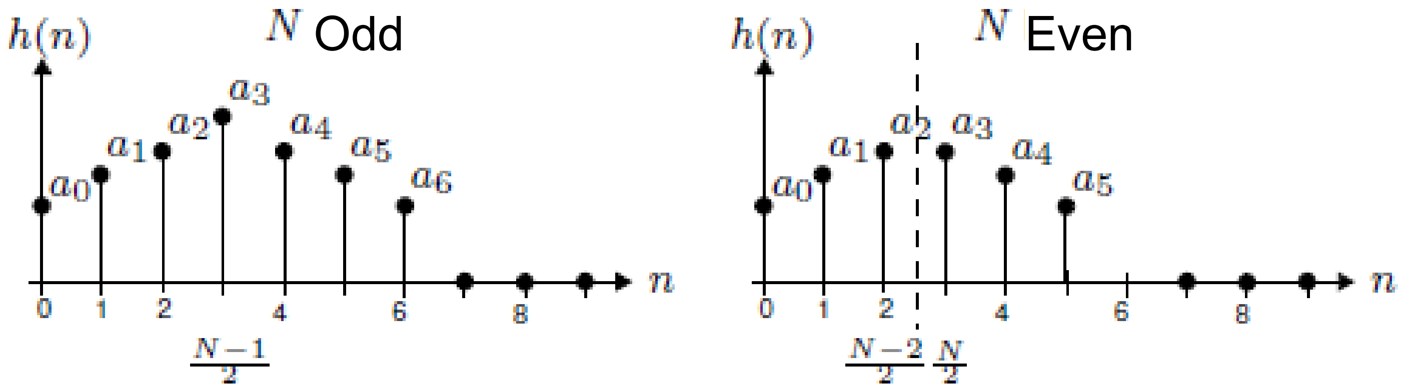


Q: stable?

Linear phase

An FIR filter with $N \geq 2$ samples has ***linear phase*** if it is symmetrical around the midpoint, i.e.

$$h(i) = h(N-1-i) \quad \text{for } i = 0, 1, \ldots, [N/2]-1$$

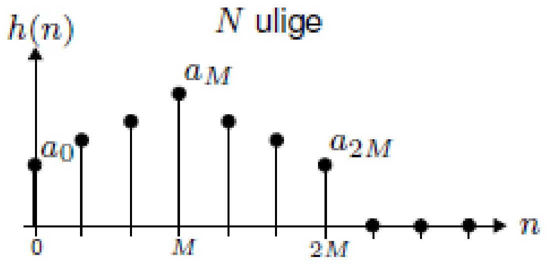


Q: can you write their transfer functions?

Prove

In the following, it is assumed that N is an odd number and the middle sample gets index

$$M = \frac{N-1}{2}$$



This means that

$$h(n) = h(2M - n)$$

And

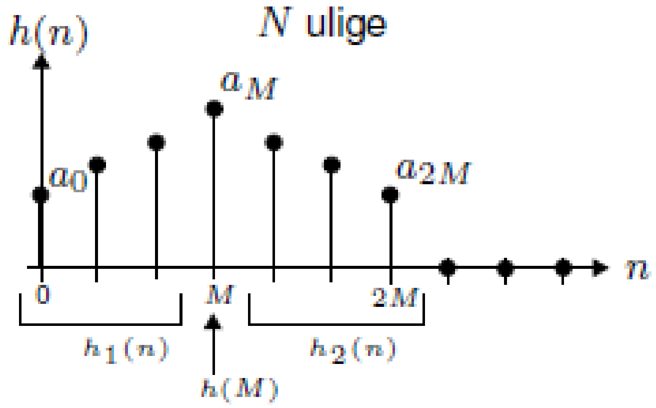
$$a_i = a_{2M-i}$$

SDU

Impulse response reformulation

The expression for an FIR filter's impulse response can be broken down as follows

$$h(n) = h_1(n) + h(M) + h_2(n)$$



This can be written as

$$h(n) = \sum_{i=0}^{M-1} a_i \delta(n-i) + a_M \delta(n-M) + \sum_{i=M+1}^{2M} a_i \delta(n-i)$$

because $h(n)$ is symmetric:

$$h(n) = \sum_{i=0}^{M-1} a_i \delta(n-i) + a_M \delta(n-M) + \sum_{i=0}^{M-1} a_i \delta\big(n-(2M-i)\big)$$

$$= a_M \delta(n-M) + \sum_{i=0}^{M-1} a_i \Big(\delta(n-i) + \delta\big(n-(2M-i)\big)\Big)$$

SDU

Impulse response reformulation

$$h(n) = a_M\delta(n-M) + \sum_{i=0}^{M-1} a_i\left(\delta(n-i) + \delta\big(n-(2M-i)\big)\right)$$

The transfer function can be found by z-transform as

$$H(z) = a_M z^{-M} + \sum_{i=0}^{M-1} a_i\left(z^{-i} + z^{-(2M-i)}\right)$$

$$= a_M z^{-M} + \sum_{i=0}^{M-1} a_i z^{-M}\left(z^{M-i} + z^{-(M-i)}\right)$$

To find the frequency response, z is replaced by $e^{j\omega T} = e^{j2\pi f/2f_0}$ and we have

f_0 Nyquist frequency

$$H\left(e^{j\pi f/f_0}\right) = a_M e^{-jM\pi f/f_0} + \sum_{i=0}^{M-1} a_i e^{-jM\pi f/f_0}\left(e^{j(M-i)\pi f/f_0} + e^{-j(M-i)\pi f/f_0}\right)$$

This can be written ($\gamma \coloneqq f/f_o$)

$$\boxed{H(\gamma) = e^{-jM\pi\gamma}\left(a_M + \sum_{i=0}^{M-1} 2a_i\cos((M-i)\gamma\pi)\right.}$$

Euler's formula:

$$\cos(jx) = \frac{e^{-x}+e^{x}}{2}$$

Amplitude and phase

The transfer function

$$H(\gamma) = e^{-jM\pi\gamma}(a_M + \sum_{i=0}^{M-1} 2a_i \cos((M-i)\gamma\pi)$$

Its amplitude is

$$\boxed{|H(\gamma)| = a_M + \sum_{i=0}^{M-1} 2a_i \cos((M-i)\gamma\pi)}$$

And phase

$$\boxed{\angle H(\gamma) = -M\pi\gamma}$$

Linear phase!

Its group delay

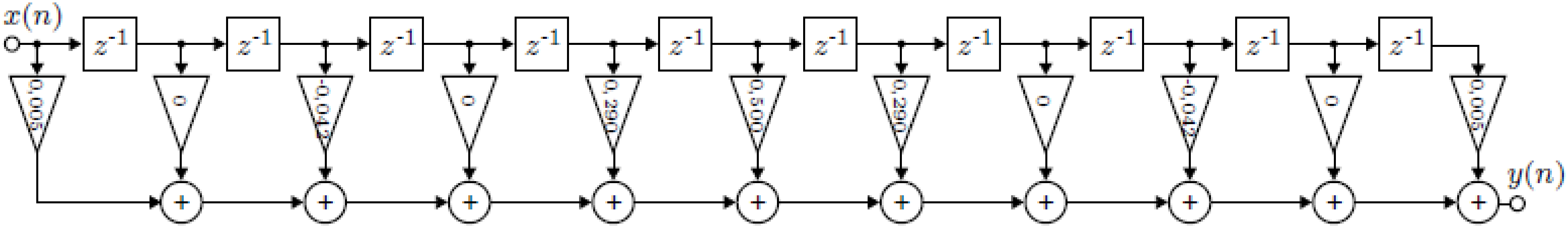
$$T_g = -\frac{d\phi(\omega)}{d\omega} = MT$$

$\phi(\omega)$ is the phase of $H(\omega)$

where T is the sampling interval [s]

Example

Let's consider an FIR filter with the following realization structure and sampling frequency $f_s = 40$ kHz



What is the filter's length?

$$N = 11$$

Q: What is the number of order?

What is the M value of the filter?

$$M = \frac{N-1}{2} = \frac{11-1}{2} = 5$$

What are the coefficients of the filter?

$$[0.005, 0, -0.042, 0, 0.290, 0.500, 0.290, 0, -0.042, 0, 0.005]$$

SDU

Example (amplitude and phase)

When the input signal x(n) is a sinusoidal at $f = 10$ kHz

$$\gamma = \frac{f}{f_0} = \frac{10000}{40000/2} = 0.5$$

$$|H(\gamma)| = a_M + \sum_{i=0}^{M-1} 2a_i \cos((M-i)\gamma\pi)$$

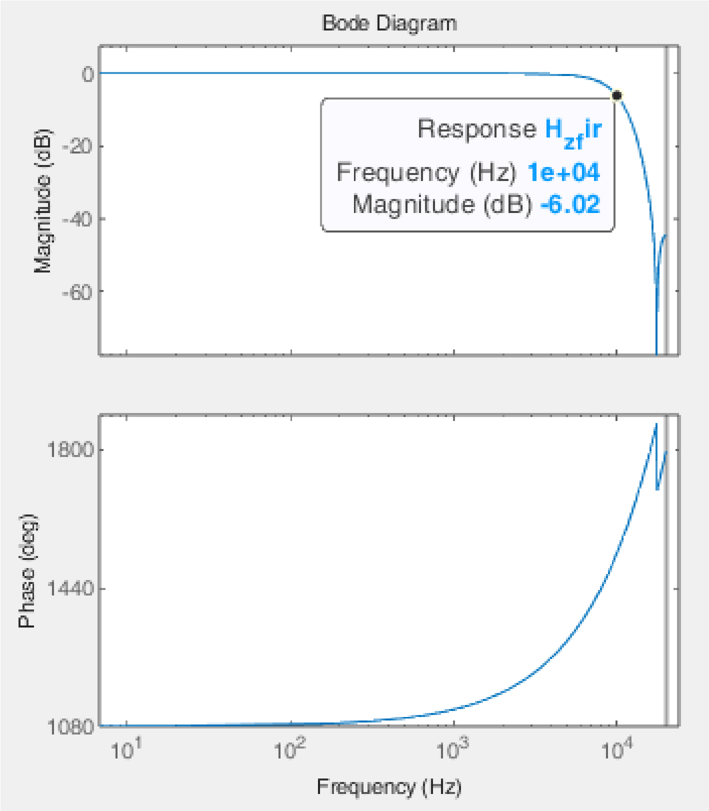
$$\angle H(\gamma) = -M\pi\gamma$$

$$|H(10k)| = 20\log\left[0.5 + \sum_{i=0}^{4} 2a_i \cos\left((5-i)\frac{\pi}{2}\right)\right] = -6.02 \text{ dB}$$

The phase is

$$\phi(10k) = -M\pi\gamma = -\frac{5\pi}{2}$$

Example (Bode plot)




```
% Matlab code

H_z_fir = tf([0.005, 0, -0.042, 0, 0.290, 0.500, 0.290, 0, -0.042, 0, 0.005],1, 1/40000)
options = bodeoptions;
options.FreqUnits = 'Hz';
bode(H_z_fir, options)
```

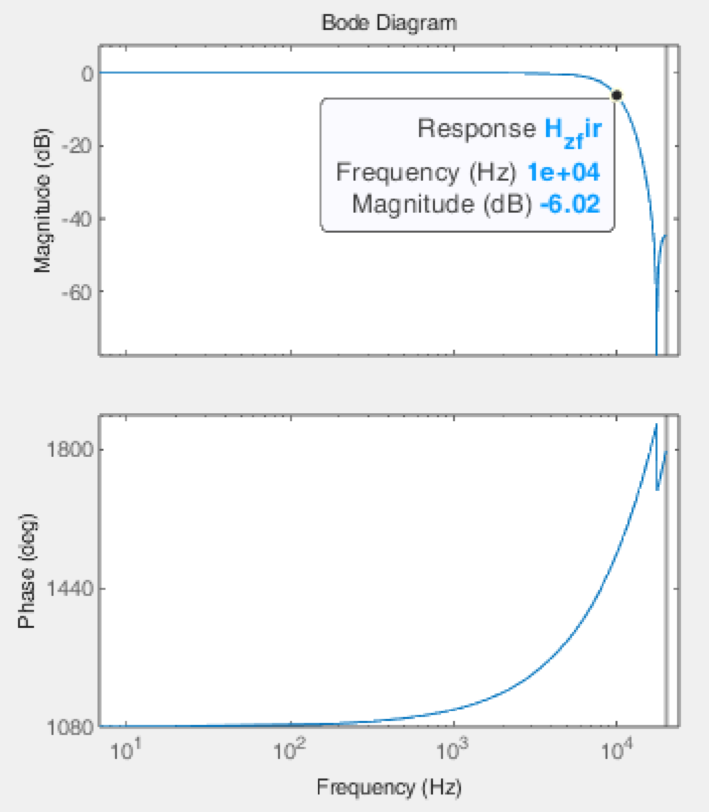
SDU

Design of FIR filter

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



Recall: Fourier series of periodic signal

Let $f(t)$ be a periodic signal with period time T. Then it can be expressed as

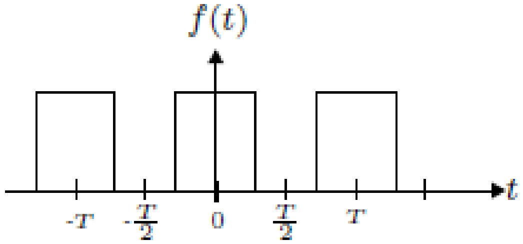
$$f(t) = \sum_{m=-\infty}^{\infty} c_m \, e^{jm(2\pi/T)t}$$

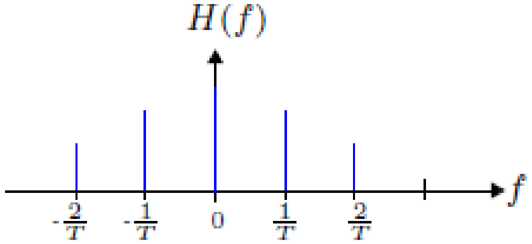
And the Fourier coefficients c_m can be expressed

$$c_m = \frac{1}{T} \int_{-T/2}^{T/2} f(t) \, e^{-jm(2\pi/T)t} dt$$

Recall: Fourier series of a periodic signal

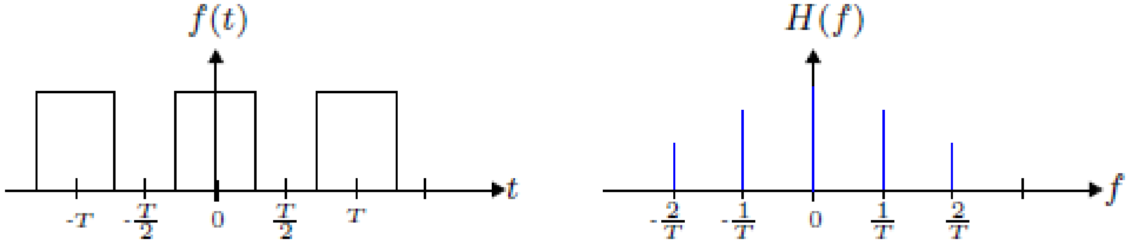
By Fourier transform, a periodic signal $f(t)$ with period T can be expressed into an infinite number of discrete frequency components $H(f)$.





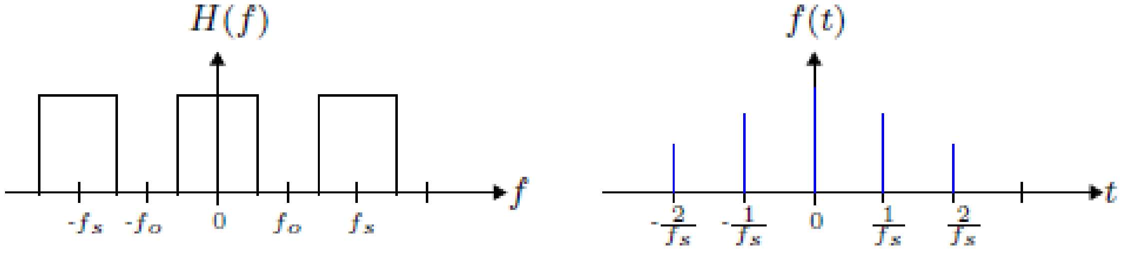
Recall: Fourier series of a periodic signal

By Fourier transform, a periodic signal $f(t)$ with period T can be resolved into an infinite number of discrete frequency components at frequencies that are integer multiples of the fundamental frequency $f_1 = 1/T$.



Vice versa…

If the filter's amplitude $|H(f)|$ is periodic with f_s, we can apply an inverse Fourier transform. Then a discrete-time signal is obtained by integer multiples of $1/f_s$



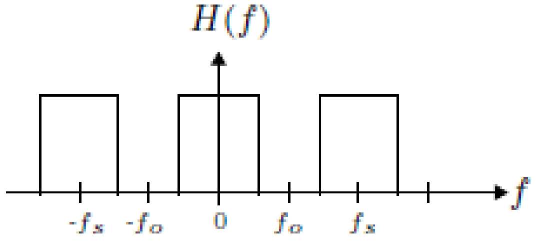
Fourier transform of $|H(f)|$

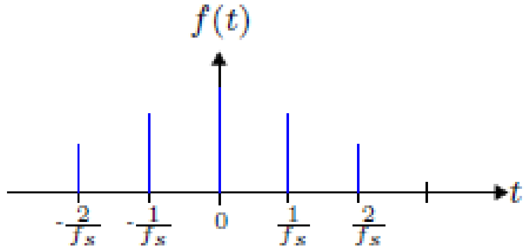
Fourier transform can be used

$$|H(f)| = \sum_{m=-\infty}^{\infty} c_m e^{jm(2\pi/f_s)f} = \sum_{m=-\infty}^{\infty} c_m e^{j2\pi mTf}$$

and the Fourier coefficients are

$$c_m = \frac{1}{f_s}\int_{-f_s/2}^{f_s/2} |H(f)| e^{-j2\pi mTf}\, df$$





SDU

Fourier transform of a FIR filter

Let's find the coefficients of FIR filters by Fourier transform of the frequency response function $|H(f)|$

$$|H(f)| = \sum_{m=-\infty}^{\infty} c_m e^{j2\pi mTf}$$

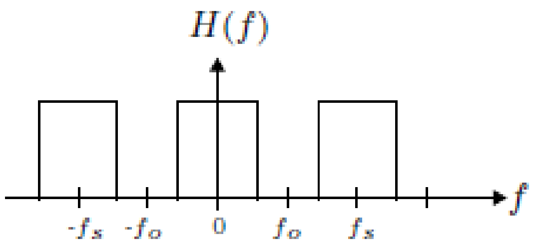
$\boxed{\text{Is it necessary?}}$

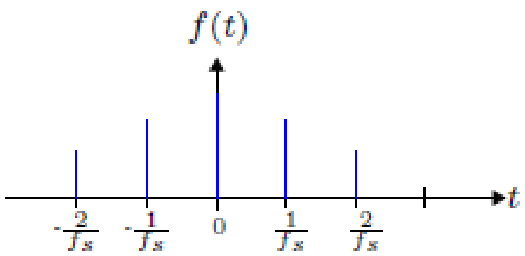
where the Fourier coefficients can be expressed

$$c_m = \frac{1}{f_s}\int_{-f_s/2}^{f_s/2} |H(f)| e^{-j2\pi mTf}\, df$$

For a FIR (finite impulse response) filter, the Fourier series for $|H(f)|$ is truncated so that it has $2M+1$ terms, i.e.

$$|H(f)| = \sum_{m=-M}^{M} c_m e^{j2\pi mTf}$$





Fourier transform of a FIR filter

The filter becomes

$$|H(f)| = \sum_{m=-M}^{M} c_m e^{j2\pi mTf}$$

Given $z = e^{j2\pi fT}$, we can have

$$H(z) = \sum_{m=-M}^{M} c_m z^m$$

Here, we investigate frequency response, so we set

$$z = e^{j\omega T}$$

Since this filter is **not causal, a delay of M samples** is introduced

$$H(z) = z^{-M} \sum_{m=-M}^{M} c_m z^m = \sum_{m=-M}^{M} c_m z^{m-M} = \sum_{i=0}^{2M} c_{M-i} z^{-i}$$

From the above formula, the filter coefficients can be determined by

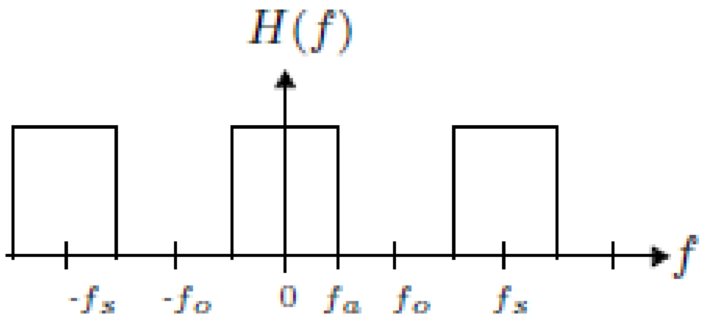
$$\boxed{a_i = c_{M-i}}$$

SDU

Example: low pass filter

An ideal low-pass filter has the following amplitude characteristic

$$|H(f)| = \begin{cases} 1, & \text{for } 0 < f < f_a \\ 0, & \text{for } f_a < f < f_0 \end{cases}$$



The coefficients of the low-pass filter can therefore be calculated as (for $m \neq 0$)

$$c_m = c_{-m} = \frac{1}{f_s}\int_{-f_s/2}^{f_s/2} |H(f)| e^{-j2\pi mTf}\, df$$

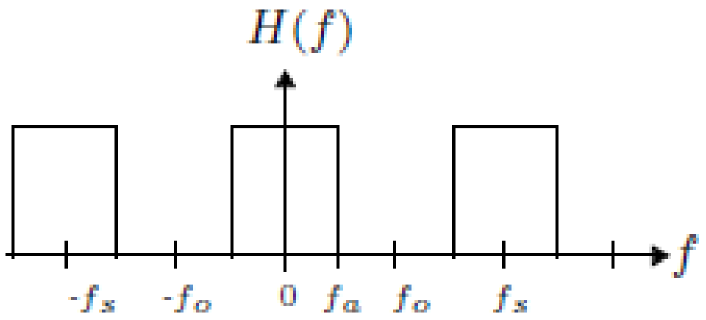
Euler's formula is used here

$$c_m = \frac{2}{f_s}\int_{0}^{f_s/2} |H(f)| \cos(2\pi mTf)\, df = 2T\int_0^{f_0} 1\cdot\cos(2\pi mTf)\, df$$

$$= \frac{2T}{2\pi mT}\int_0^{f_0} \cos(2\pi mTf)\, d(2\pi mTf) = \frac{1}{m\pi}[\sin(2\pi mTf)]_{f=0}^{f_a} = \frac{1}{m\pi}\sin(2\pi mTf_a)$$

SDU

Example: low pass filter



An ideal low-pass filter has the following amplitude characteristic

$$|H(f)| = \begin{cases} 1, & \text{for } 0 < f < f_a \\ 0, & \text{for } f_a < f < f_0 \end{cases}$$

The coefficients of the low-pass filter can therefore be calculated as (for $m \neq 0$)

$$\boxed{c_m = \frac{1}{m\pi} \sin(2\pi m T f_a)}$$

For $m = 0$, we can use **L'Hospitals rule**

$$c_0 = \frac{\sin(2\pi m T f_a)}{m\pi} = \frac{(\sin(2\pi m T f_a))'}{(m\pi)'} = 2Tf_a$$

L'Hospitals rule: If $\lim_{x\to c} f(x) = \lim_{x\to c} g(x) = 0$ or $\pm\infty$, and $g'(x) \neq 0$, and $\lim_{x\to c} \frac{f'(x)}{g'(x)}$ exists, then

$$\lim_{x\to c} \frac{f(x)}{g(x)} = \lim_{x\to c} \frac{f'(x)}{g'(x)}$$

Example: low pass filter

If a low-pass filter with cut-off frequency $f_a = 2$ kHz is desired via a FIR filter with 23 samples.

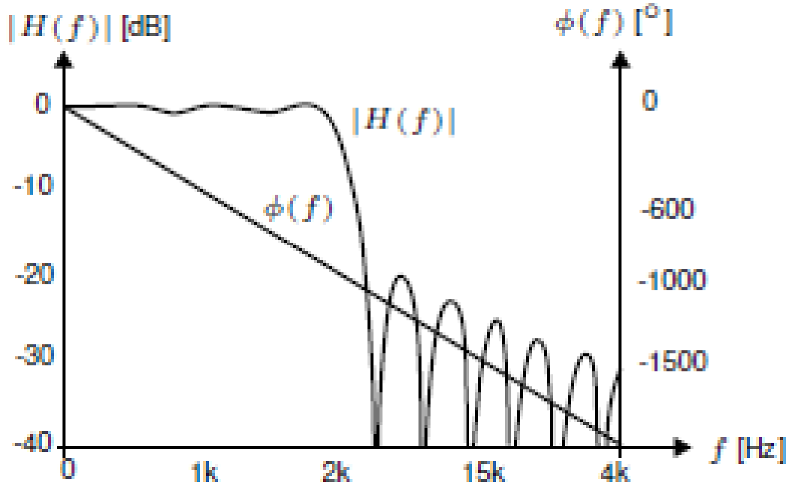
$$M = \frac{23-1}{2} = 11$$

The coefficients can thus be found from the Fourier coefficients as

$$a_i = c_{11-i}$$

The coefficients can thus be calculated from previous formulas:

$$\boxed{\begin{gathered} c_m = \frac{1}{m\pi}\sin(2\pi m T f_a), \text{ for } m \neq 0 \\ c_0 = 2Tf_a \end{gathered}}$$

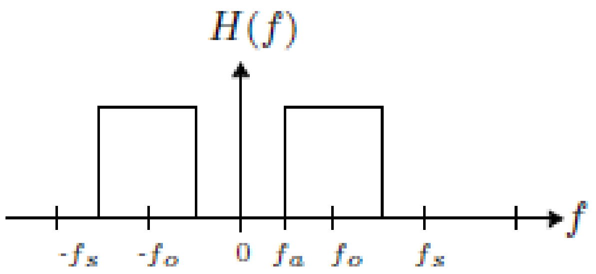


SDU

Example: High pass filter

An ideal high-pass filter has the following amplitude characteristic

$$|H(f)| = \begin{cases} 0, & \text{for } 0 < f < f_a \\ 1, & \text{for } f_a < f < f_0 \end{cases}$$



The coefficients of the high-pass filter can therefore be calculated as (for $m \neq 0$)

$$c_m = c_{-m} = \frac{2}{f_s}\int_{f_a}^{f_0} |H(f)| \cos(2\pi mTf)\, df = 2T\int_{f_a}^{f_0} 1 \cdot \cos(2\pi mTf)\, df$$

$$= \frac{2T}{2\pi mT}\int_{f_a}^{f_0} \cos(2\pi mTf)\, d(2\pi mTf) = \frac{1}{m\pi}[\sin(2\pi mTf)]_{f=f_a}^{f_0} = \frac{1}{m\pi}(\sin(\pi m) - \sin(2\pi mTf_a))$$

Last coefficients (again using L'Hospitals rule)

$$c_0 = 1 - 2Tf_a$$

Example: Bandpass filter

An ideal bandpass filter has the following amplitude characteristic

$$|H(f)| = \begin{cases} 1, & \text{for } f_{a1} < f < f_{a2} \\ 0, & \text{else} \end{cases}$$

The coefficients of the bandpass filter can therefore be calculated as (for $m \neq 0$)

$$c_m = c_{-m} = \frac{1}{m\pi}\left[\sin(2\pi mTf)\right]_{f=f_{a_1}}^{f_{a_2}} = \frac{1}{m\pi}\left(\sin\left(2\pi mTf_{a_2}\right) - \sin\left(2\pi mTf_{a_1}\right)\right)$$

Last coefficients (using L'Hospitals rule)

$$c_0 = 2T(f_{a_2} - f_{a_1})$$

This gives a center frequency

$$f_c = \frac{f_{a_2} - f_{a_1}}{2}$$

SDU

Example: Bandstop filter

An ideal bandstop filter has the following amplitude characteristic

$$|H(f)| = \begin{cases} 0, & \text{for } f_{a1} < f < f_{a2} \\ 1, & \text{else} \end{cases}$$

The coefficients of the bandstop filter can therefore be calculated as (for $m \neq 0$)

$$c_m = c_{-m} = \frac{1}{m\pi}\left[\sin(2\pi mTf)\right]_{f=0}^{f_{a_1}} + \frac{1}{m\pi}\left[\sin(2\pi mTf)\right]_{f=f_{a_2}}^{f_0}$$
$$= \frac{1}{m\pi}\left(\sin(m\pi) + \sin\left(2\pi mTf_{a_1}\right) - \sin\left(2\pi mTf_{a_2}\right)\right)$$

Last coefficients (using L'Hospitals rule)

$$c_0 = 1 - 2T(f_{a_2} - f_{a_1})$$

This gives a center frequency

$$f_c = \frac{f_{a_2} - f_{a_1}}{2}$$

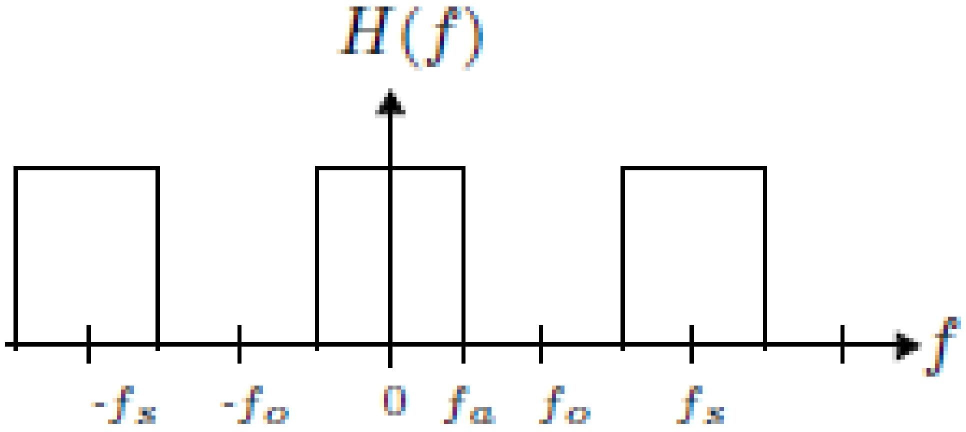
SDU

FIR design summary

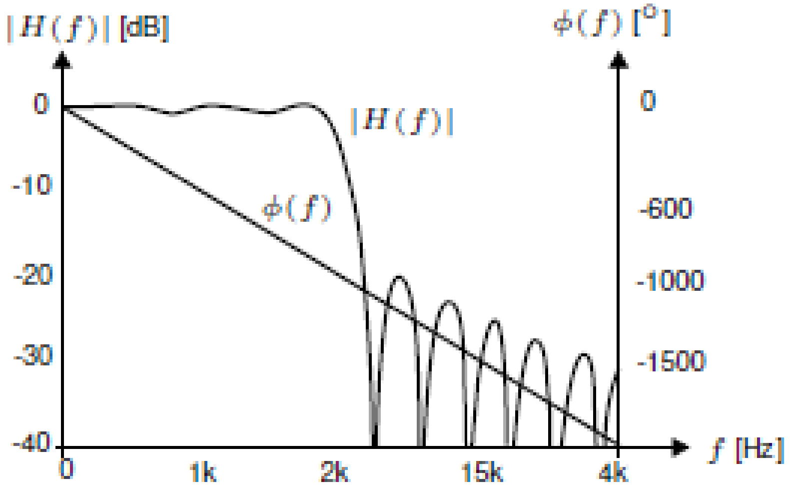
The following shows an overview of Fourier coefficients for the four filter types.

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}

Problems of the current design



Desire



Result

Big side lopes due to truncation!

Solution: Multiply impulse response by a window function!

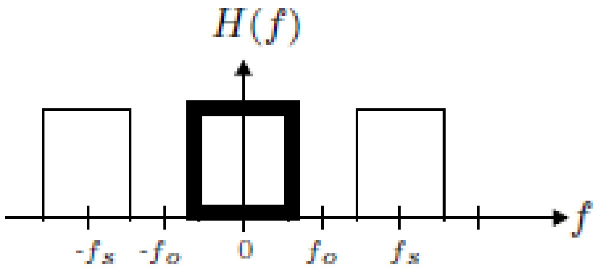
SDU

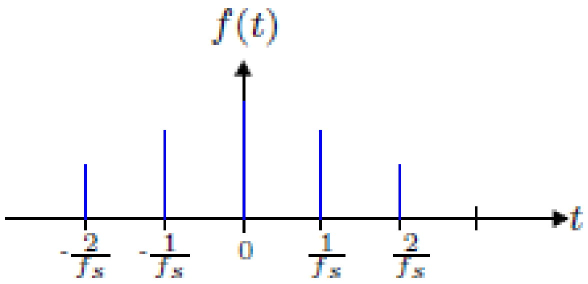
Recall: slide in ‘Sampling & reconstruction’

Relationship btw time domain and freq domain

Discrete signal in time domain -> periodic frequency spectra

Periodic signal in time domain -> discrete frequency spectra





A signal that is time-limited -> it must have infinite spectral content.

A signal that is band-limited -> it must extend infinitely in time.

Multiply in frequency domain -> Convolution in time domain
Convolution in frequency domain -> Multiply in time domain

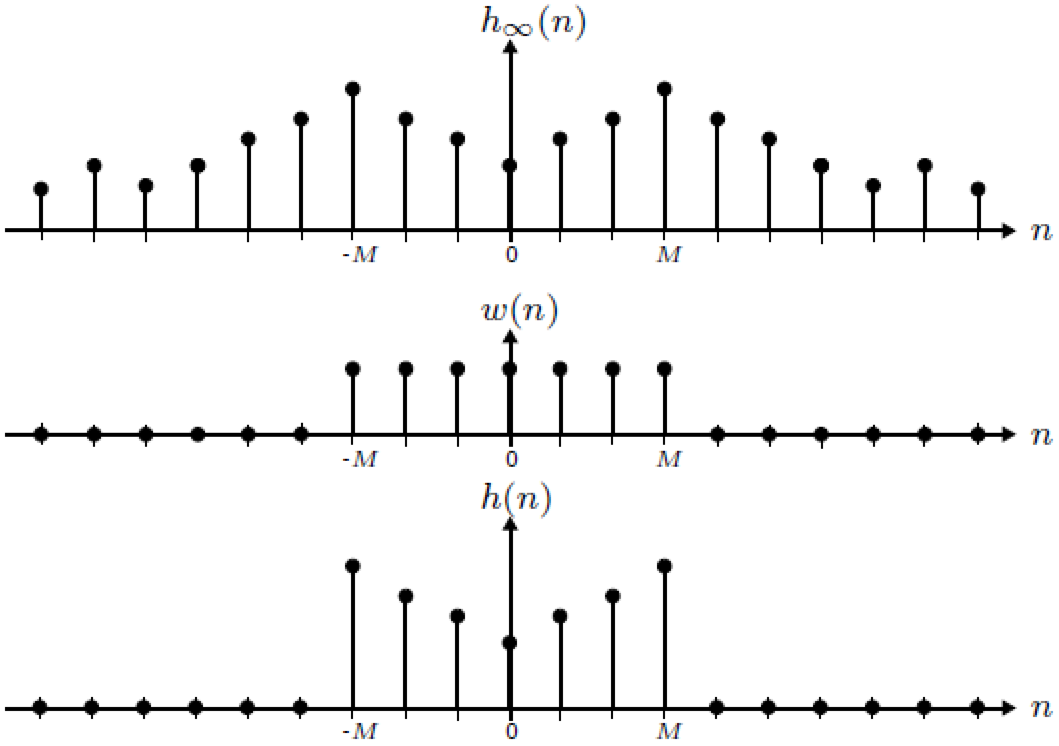
Window functions in FIR

If the previous impulse response had not been truncated, then it would be infinite, $h_\infty(n)$.
To obtain the impulse response of the FIR filter, $h_\infty(n)$ could be multiplied by the **rectangular window function**, i.e.

$$h(n) = h_\infty(n)\omega(n)$$

where

$$\omega(n) = \begin{cases} 1, & -M \leq x \leq M \\ 0, & else \end{cases}$$

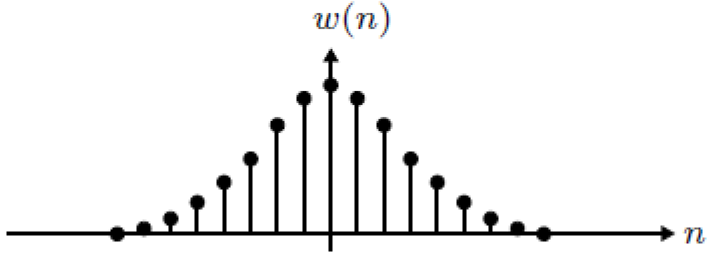


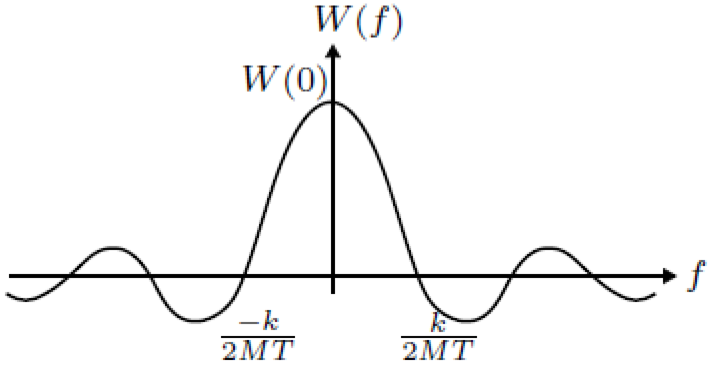
SDU

Frequency response

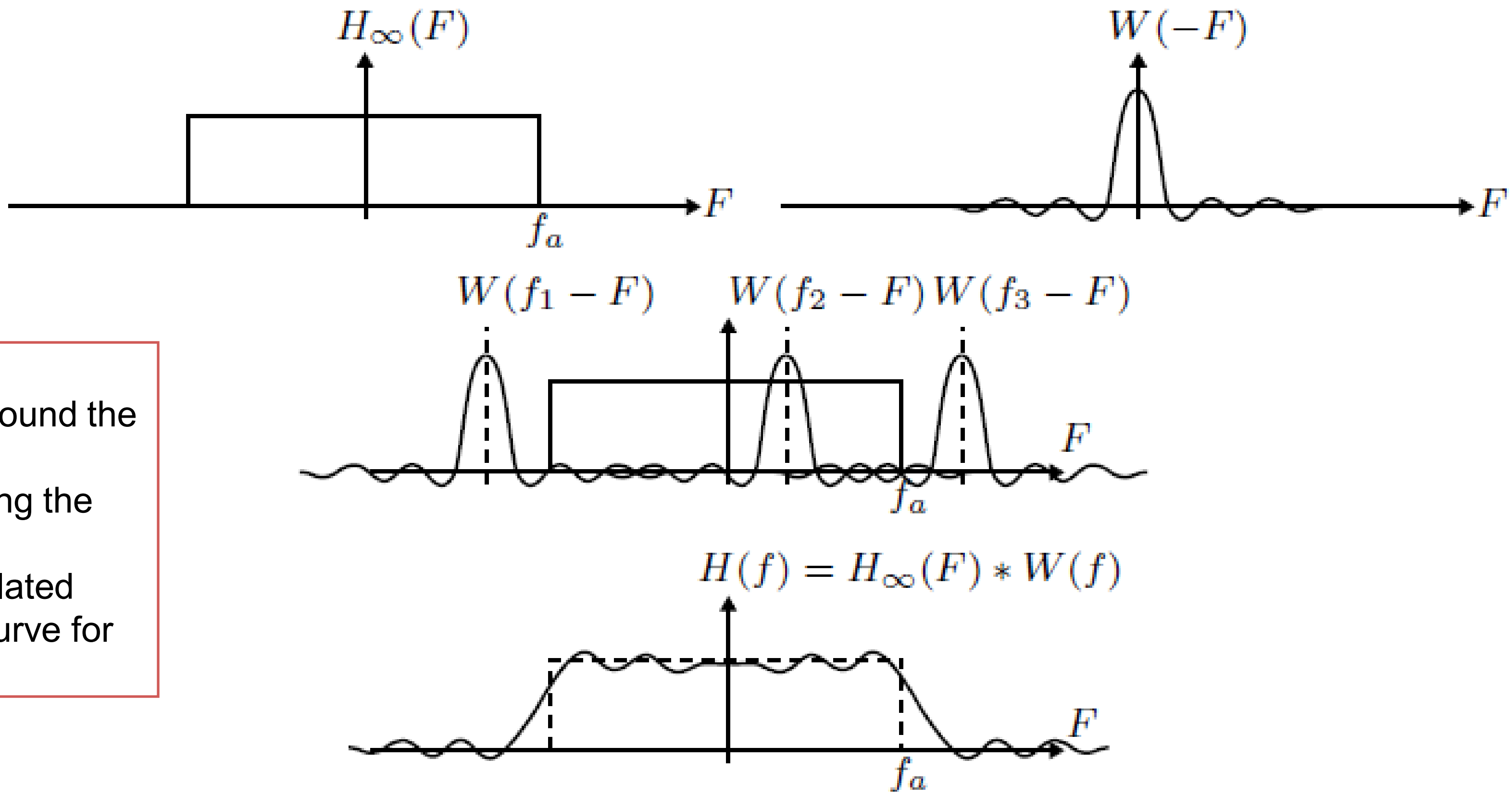
The frequency response of the impulse response sequence $h(n)$ is obtained by

$$H(f) = H_\infty(f) * W(f) = \int_{-\infty}^{\infty} H_\infty(f) W(f - F) dF$$





Some definitions:
The spectrum between crossings around 0 Hz is called the **main lobe**.
Remaining spectrum is **side lobes**.



Procedure:

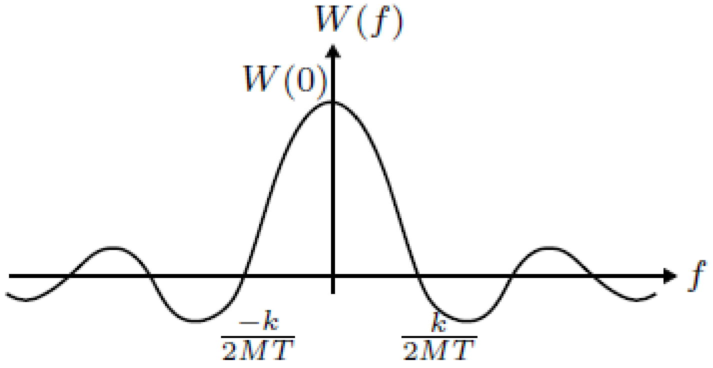
1. $W(-F)$ is found by mirroring $W(f)$ around the y-axis
2. $W(-F)$ is shifted from left to right along the frequency axis.
3. The product $H_\infty(F)W(f - F)$ is calculated
4. $H(f)$ is equal to the area under the curve for $H_\infty(F)W(f - F)$

The oscillations in $H(f)$ are called Gibbs oscillations and are referred to as the Gibbs phenomenon.

A window function is desired to have the following characteristics

→The width of the **main lobe** must be small. This provides a **quick transition** between passband and stopband.

→The maximum amplitude of the **sidelobes** must be small. This gives small passband and stopband ripple.



Typical Window functions

→ Rectangular window
→ Bartlett window
→ Hamming window
→ Hanning window
→ Kaiser window

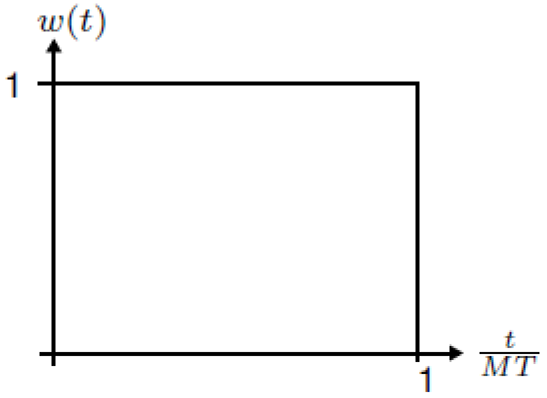
Rectangle window

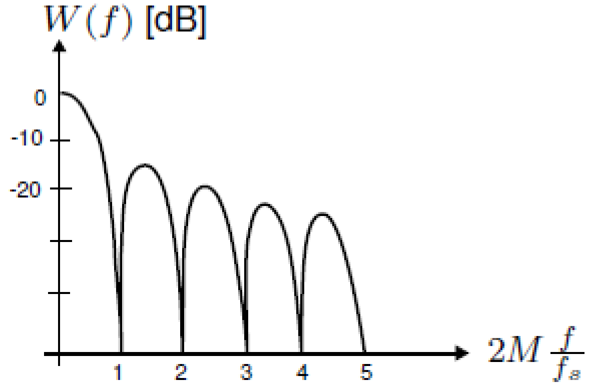
A rectangular window is defined as

$$w(n) = \begin{cases} 1 & \text{hvis } -M \leq n \leq M \\ 0 & \text{ellers} \end{cases}$$

The spectrum function of the rectangular filter is

$$W(f) = \frac{\sin(2\pi f M t)}{\pi f}$$



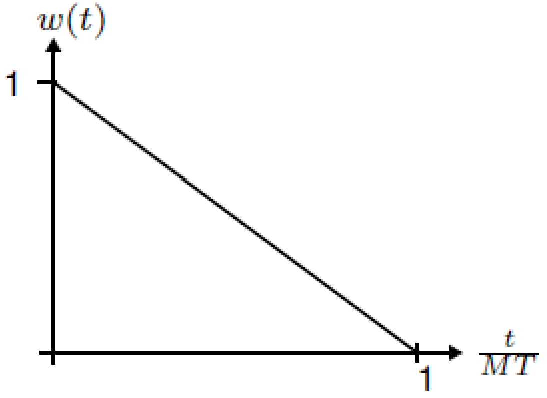


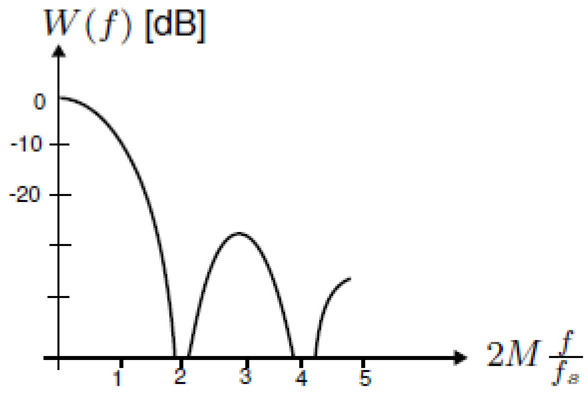
SDU

Bartlett window

A Bartlett window is defined as

$$w(n) = \begin{cases} 1 - \frac{|n|}{M} & \text{hvis } -M \leq n \leq M \\ 0 & \text{ellers} \end{cases}$$





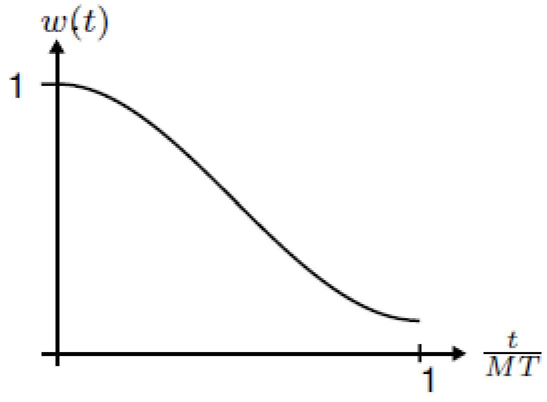
Hamming and Hanning window

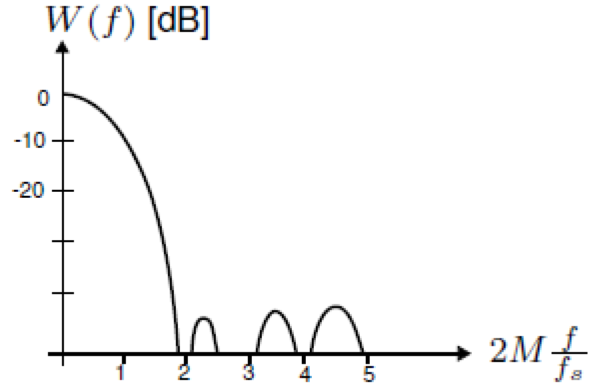
A Hamming or Hanning window is defined as

$$w(n) = \begin{cases} \alpha + (1-\alpha)\cos(\frac{n\pi}{M}) & \text{hvis } -M \leq n \leq M \\ 0 & \text{ellers} \end{cases}$$

Where $\alpha = 0.5$ for Hanning Window, and $\alpha = 0.54$ for Hamming Window

Hamming window





SDU

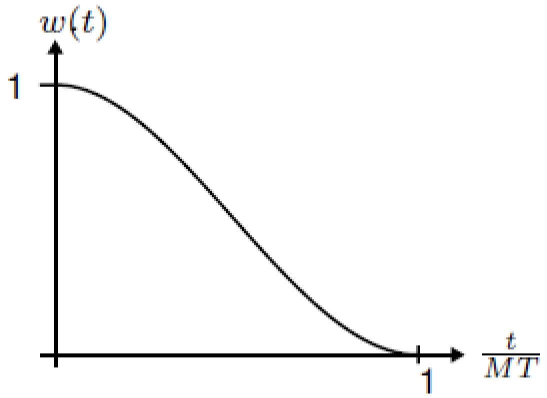
Hamming and Hanning window

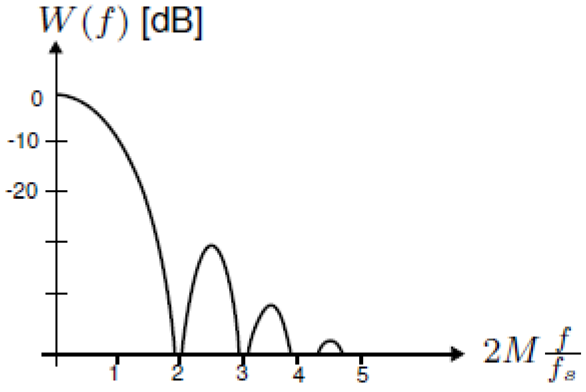
A Hamming or Hanning window is defined as

$$w(n) = \begin{cases} \alpha + (1-\alpha)\cos(\frac{n\pi}{M}) & \text{hvis } -M \le n \le M \\ 0 & \text{ellers} \end{cases}$$

Where $\alpha = 0.5$ for Hanning Window, and $\alpha = 0.54$ for Hamming Window

Hanning window





SDU

Kaiser window

The Kaiser window is introduced, as the other windows only have M as a parameter, which makes it impossible to adjust the main lobe and side lobes independently of each other.

Kaiser windows are defined as

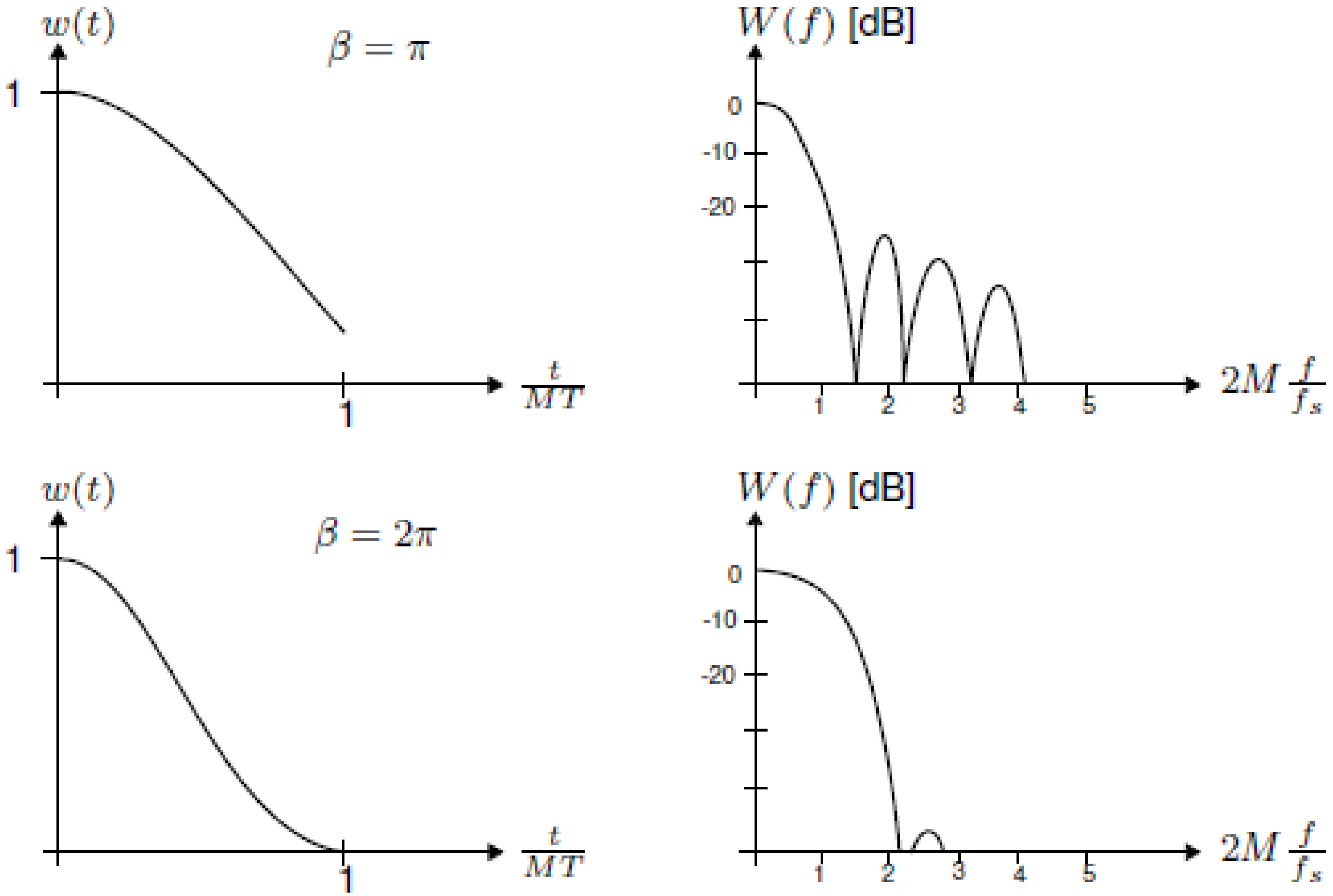
$$w(n) = \begin{cases} \frac{I_0\left(\beta\sqrt{1-\left(\frac{n}{M}\right)^2}\right)}{I_0(\beta)} & \text{hvis } -M \leq n \leq M \\ 0 & \text{ellers} \end{cases}$$

where $I_0(x)$ is a zero-order Bessel function, **β is a parameter that adjusts primarily the side lobe amplitude (usually β is between 1 and 10)**.

The Bessel function is defined as

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

Kaiser window



Matlab window functions

Check list: mathworks.com/help/signal/ug/windows.html

→ bartlett(L)
→ hamming(L)
→ hann(L)
→ kaiser(L, beta)

Design procedure

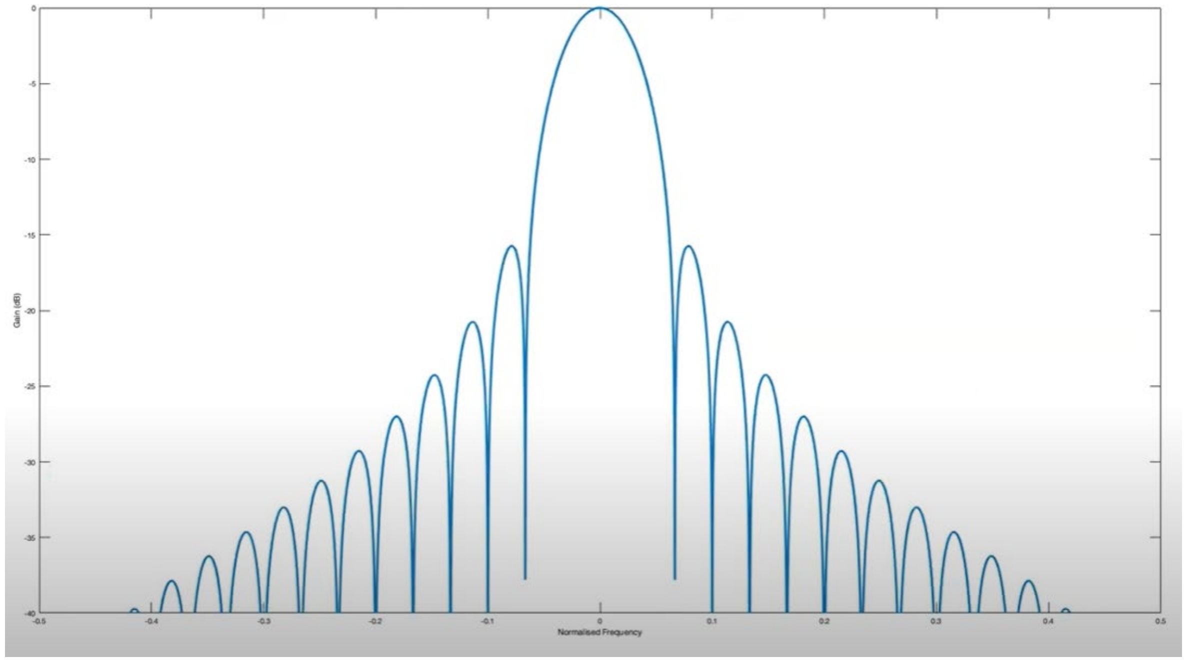
The construction of an FIR filter can proceed according to the following procedure

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_a
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).

Specs of a FIR filter

When designing an FIR filter, the following must be specified

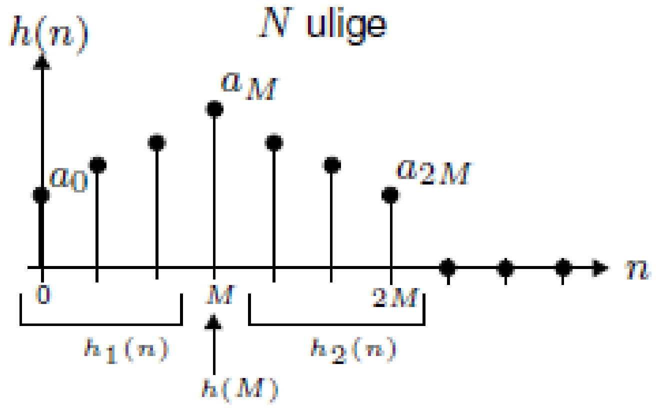
1. The cut-off frequency f_a
2. Maximum permissible width of transition area Δf_a
3. Maximum permissible stopband gain H_s
4. Maximum permissible pass band ripple H_r



Determination of filter order

When designing an FIR filter, its length cannot be determined exactly in advance.

Normally, we try different values and see whether the filter can achieve our desired filter specifications.

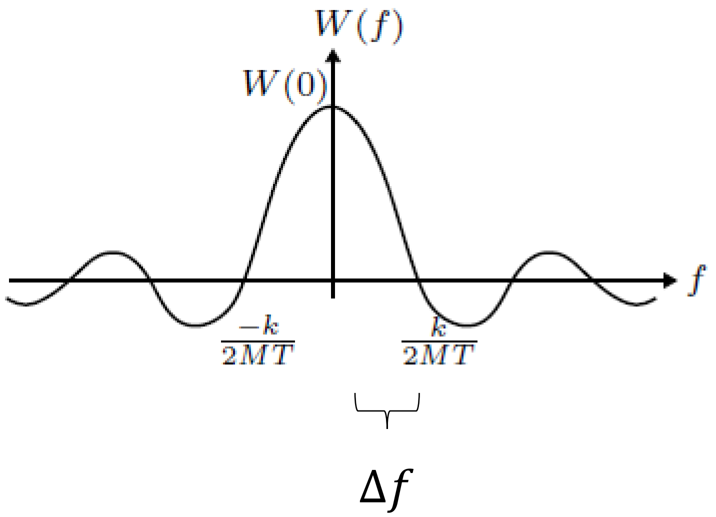


Determination of filter order

It can be seen from the previous slide that transition area $\Delta f \approx B$, where B is the main lobe width.

The frequency-normalized main lobe width is

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



Thus, the M value becomes

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

Find Bn for different window functions

The following table can also be used to select the window function.

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

Calculation of filter coefficients (with window)

When a window function is added, this corresponds to a change in the coefficients of the FIR filter.

The FIR filter's new Fourier coefficients c'_m stays for $-M \leq m \leq M$

$$c'_m = c_m \omega_m$$

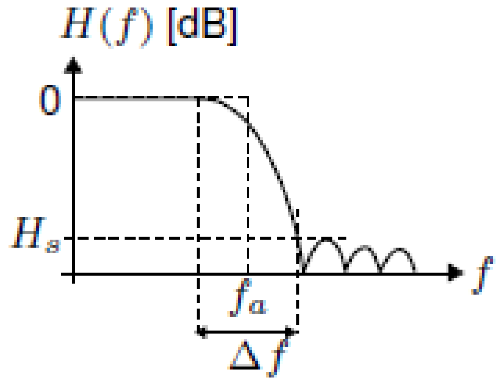
where ω_m is the mth coefficient of the window function and c_m is the mth Fourier coefficient of the ideal filter.

Now the filter coefficients can be calculated as

$$a_i = c'_{M-i}$$

Example: Design of lowpass filter

The following shows the filter specification.



a) $f_a = 1,5$ kHz
b) $\Delta f = 1$ kHz
c) $H_s \leq -50$ dB
d) $f_s = 8$ kHz

Due to the large stopband attenuation, the window function must be either Hamming or Kaiser.

We choose Hamming for example.

Example: Design of lowpass filter

The ordinal number is determined based on

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

where $B_n = 4$ for a Hamming filter (find this value from the table)

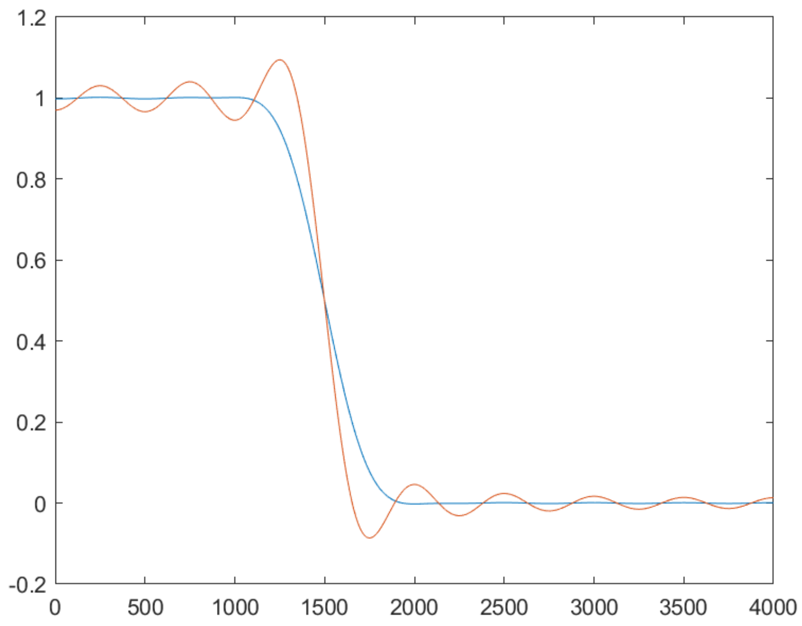
Thus, we have

$$M \geq \frac{4 * 8000}{2 * 1000} = 16$$

The filter order should be $N = 2M = 32$ or 33

Matlab

```
% Given parameters
f_a = 1500; % Cutoff frequency in Hz
f_s = 8000; % Sampling frequency in Hz
delta_f = 1000; % Transition band in Hz
T = 1/f_s;

% Estimate the filter order
M=4*8000/(2*1000);
N = 2*M+1;

% filter
c = zeros(1,M);
for m = 1:M
    c(m) = 1/(m*pi) * sin(2*pi*m*T*f_a);
end
c0 = 2*T*f_a;
cm = [fliplr(c) c0 c]';

% window
w = hamming(N);

% filter coefficients
a = cm .* w;

% input frequency
f = linspace(0,4000,1000);
f_0 = f_s/2;
gamma = f/f_0;

H_gamma_amp = zeros(1000,1);
for idx = 1:1000
    for i=1:M
        H_gamma_amp(idx) = H_gamma_amp(idx) + 2*a(i)*cos((M-
i+1)*gamma(idx)*pi);
    end
    H_gamma_amp(idx) = H_gamma_amp(idx) + a(M+1);
end

plot(f,H_gamma_amp)
```



Matlab function fir1()

fir1(n,Wn,ftype, window)
→ n: filter order
→ Wn: 1 value for low pass filter, 2 values for bandpass filter (normalized cut off frequency)
→ Ftype: ‘low’, ‘high’, ‘bandpass’, ‘stop’
→ Window: window function vector

```
% low pass filter with Hanning window
Fs = 1000;   % Sampling frequency (Hz)
Fc = 150;     % Cutoff frequency (Hz)
N  = 51;        % Filter order + 1

% Normalized cutoff
Wn = Fc / (Fs/2);

% FIR low-pass filter
b = fir1(N-1, Wn, 'low', hanning(N));
Hz = tf(b, 1, 1/Fs)
```

```
% bandstop pass filter with hamming window
Fs = 1000;   % Sampling frequency (Hz)
Fstop1 = 100;       % Lower stopband edge (Hz)
Fstop2 = 200;       % Upper stopband edge (Hz)
N  = 51;        % Filter order + 1

% Normalized cutoff
Wn = [Fstop1 Fstop2] / (Fs/2);

% FIR low-pass filter
b = fir1(N-1, Wn, ‘stop’);  % by default, it is a Hamming window
Hz = tf(b, 1, 1/Fs)
```

SDU