frontiers in
PSYCHOLOGY

GENERAL COMMENTARY
published: 09 July 2012
doi: 10.3389/fpsyg.2012.00233

# Filter effects and filter artifacts in the analysis of electrophysiological data

*Andreas Widmann* and Erich Schröger*

# Digital filter design for electrophysiological data – a practical approach

Andreas Widmann[1*], Erich Schröger[1], and Burkhard Maess[2]

[1]*Cognitive and Biological Psychology, University of Leipzig, Germany*
[2]*Max Planck Institute for Human Cognitive and Brain Sciences, Leipzig, Germany*

# Signal filtering

- Filtering is an **almost ubiquitous** step in the preprocessing of biosignals.
- Filters **improve** the signal-to-noise ratio but also **introduce** signal distortions

# Signal filtering

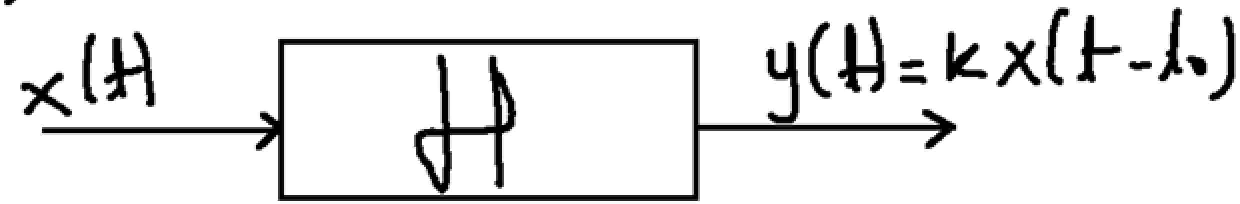
- **Temporal filtering or frequency filtering** (in contrast to spatial and other types of filtering) refers to the attenuation of signal components of a particular frequency (band).
- The common rationale behind filtering in general is to **attenuate noise** in the recordings, while preserving the signal (of interest).
- In some applications neither noise nor signal are clearly defined.

# Signal filtering

- Temporal filters cannot separate signal from noise in the **same band**; they will simply attenuate everything in the targeted band.

# Signal filtering

Sea:

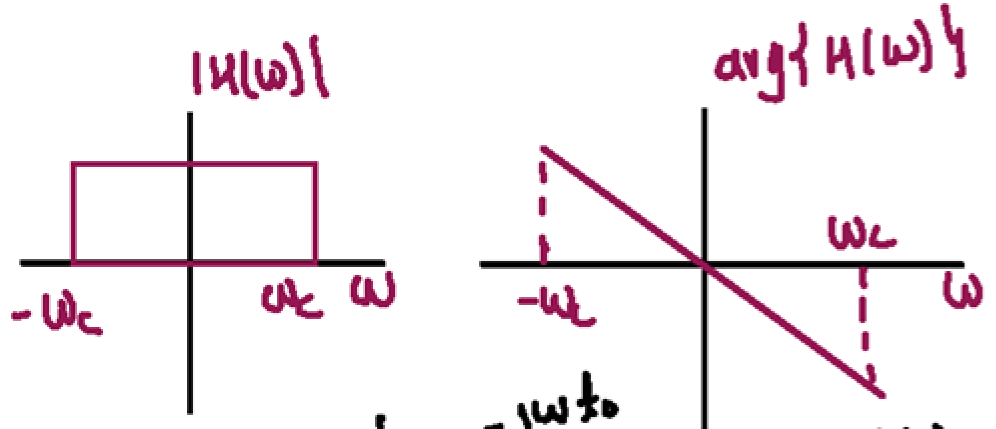



Con $x(t) \xleftrightarrow{\mathcal{F}} X(\omega)$

$$y(t) \xleftrightarrow{\mathcal{F}} Y(\omega) = k e^{-j\omega t_0} X(\omega)$$

$$\therefore \; H(\omega) = \frac{Y(\omega)}{X(\omega)} = k e^{-j\omega t_0}$$


# Signal filtering




$$H(\omega) = \begin{cases} k e^{-j\omega t_0} & -\omega_c \leq \omega \leq \omega_c \\ 0 & \text{e.o.c} \end{cases}$$

¿Cómo será $h(t)$? (Respuesta al impulso)

$$h(t) = \frac{k}{2\pi} \int_{-\omega_c}^{\omega_c} e^{j\omega t} e^{-j\omega t_0} \, d\omega$$

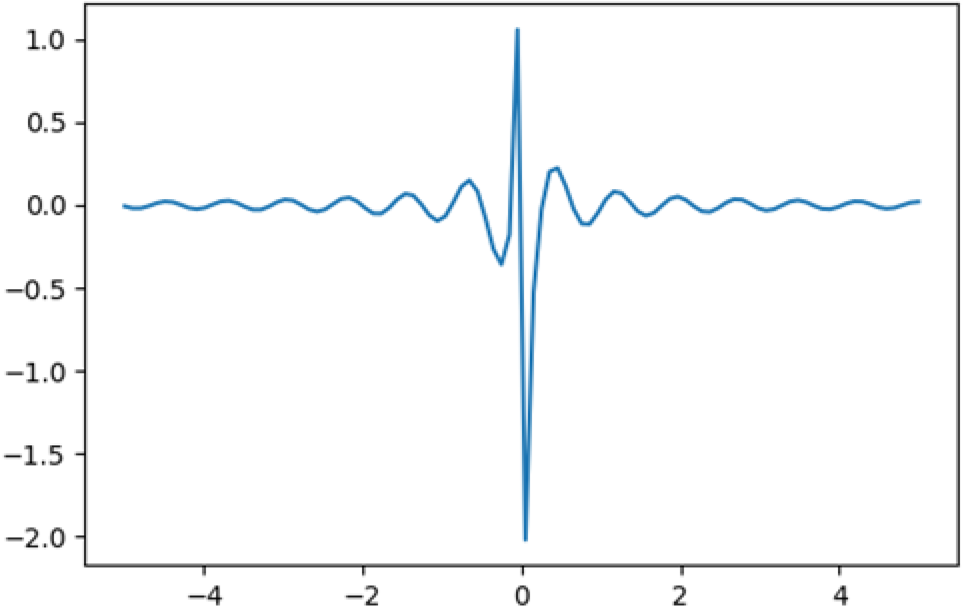
# Signal filtering

- What happens with h(t)?

```
import numpy as np
import matplotlib.pyplot as plt

t = np.linspace(-5,5,100)
wc = 2*np.pi*50
t0 = 0.001
ft = (wc/np.pi)*np.sinc(wc*(t - t0))

plt.plot(t,ft)
plt.show()
```

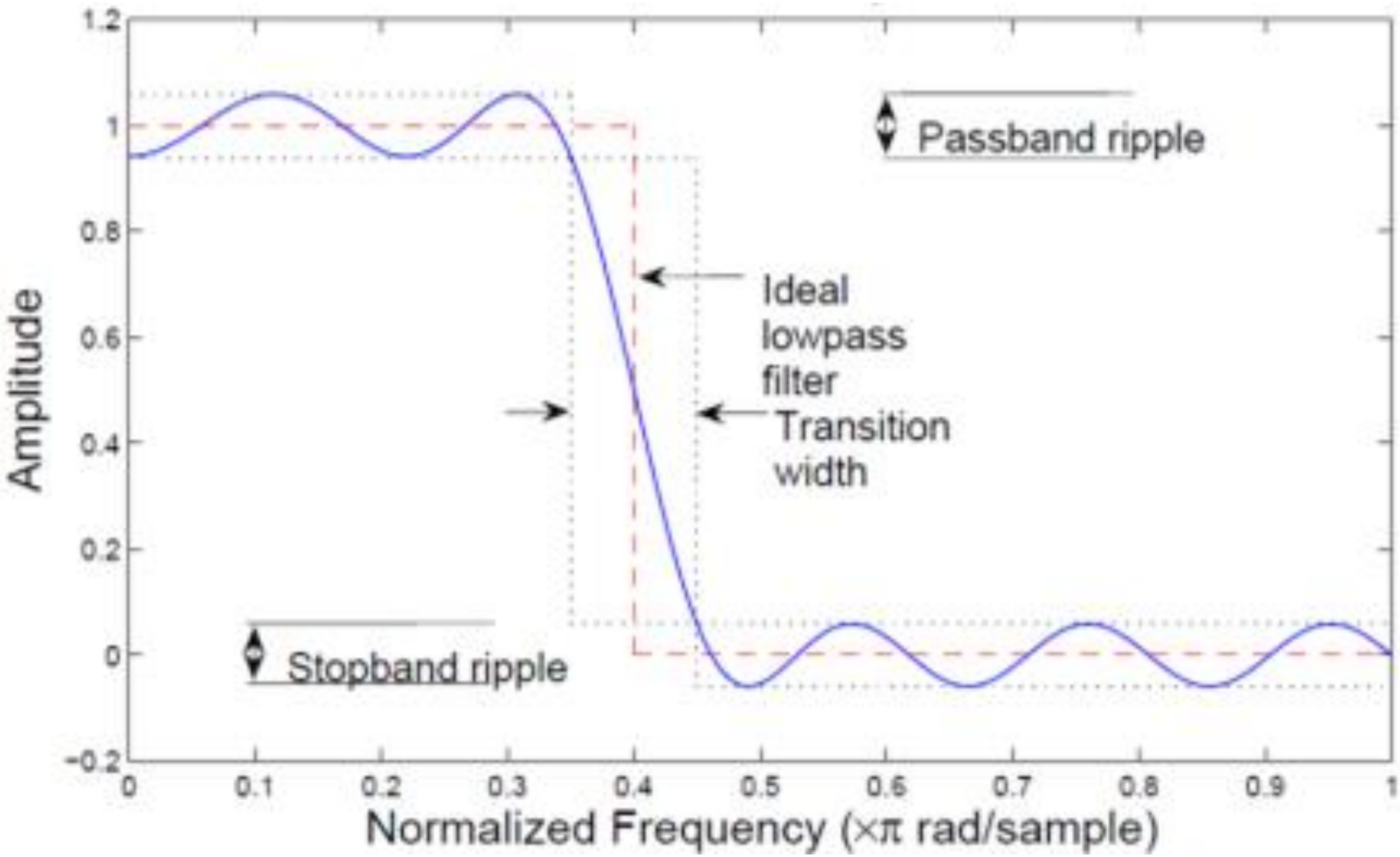


# Signal filtering

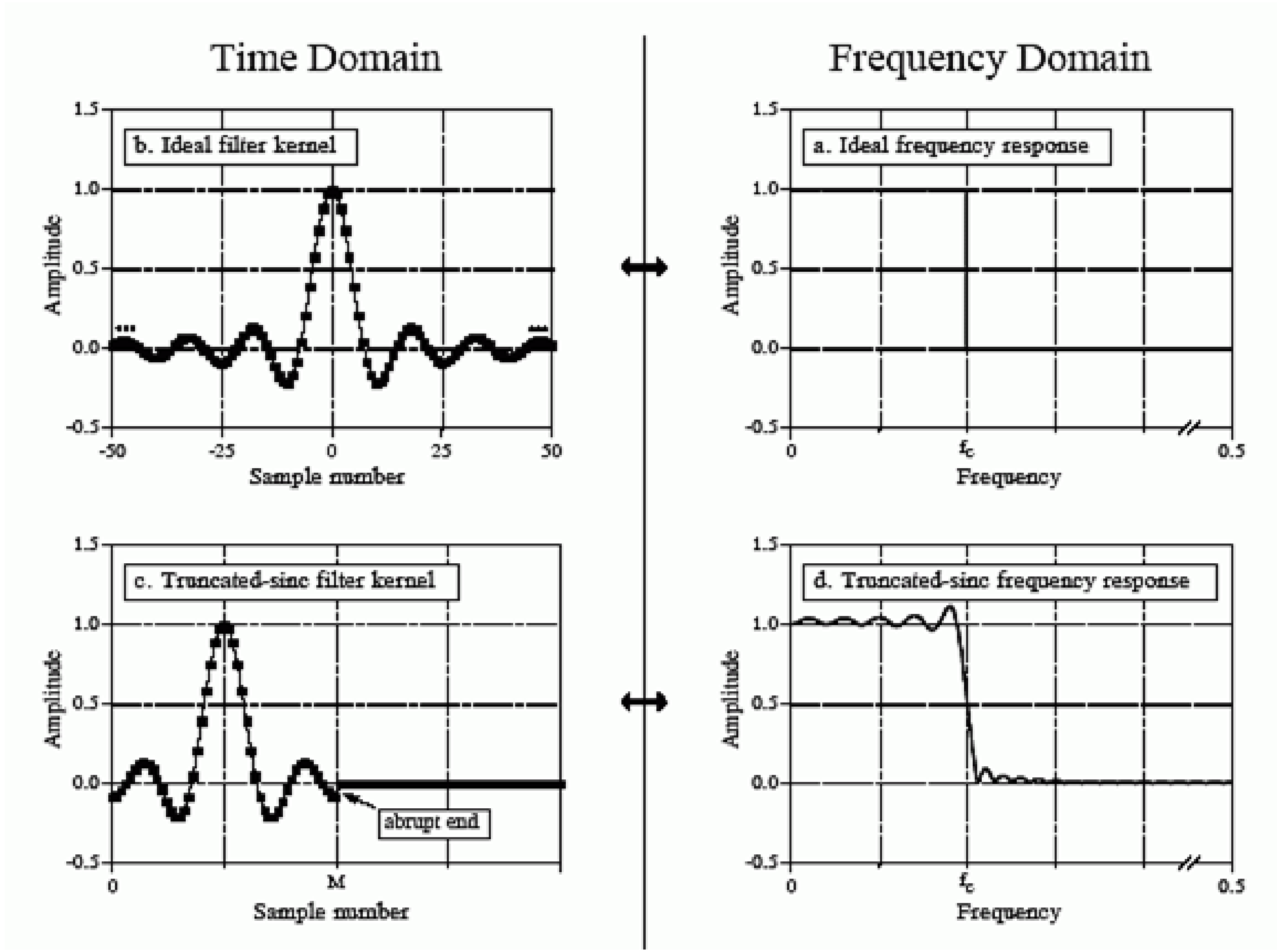
Causal filters

- $|H(w)|$ can be zero in some frequencies but can’t be zero over any frequency band
- The amplitude of the band pass frequency can’t be constant
- Magnitude and phase can’t be independently manipulated

# Signal filtering



# Signal filtering



# Signal filtering

Digital filters

- The digital filter bandwidth is limited by the sampling frequency. For the analog filter is limited by the amplifiers
- There is not saturation (although there are limitations by the number of bits used)
- Can be easily copied and transferred among systems
- Can be implemented using software or hardware
- There are not problems with tolerances and impedances
- The computers evolve faster than conventional electronics, so, better designs in digital filters can be reached

# Signal filtering

## Digital filters

FIR

$$H[z] = \frac{Y[z]}{X[z]} = B_0 + B_1 z^{-1} + B_2 z^{-2} + \dots + B_M z^{-M}$$

$\{B_k\}$: coeficientes del filtro

$M$: orden del filtro

IIR

$$H[z] = \frac{B_0 + B_1 z^{-1} + B_2 z^{-2} + \dots + B_M z^{-M}}{1 + A_1 z^{-1} + A_2 z^{-2} + \dots + A_N z^{-N}}$$

# Signal filtering

## Digital filters

- FIR (Finite Impulse Response): Always stable

- IIR (Infinite Impulse Response): Need a lower order to reach the same specification of the FIR filter.

  Have lower ripple than the FIR filter
  Have non-linear phase.

  Can be inestable

  Accumulated rounding errors result in deviating filter responses.

# Signal filtering

- Butterworth filters (IIR) have no passband and stopband ripple and have the shallowest roll-off near the cutoff frequency (the most relevant roll-off region) compared to the other commonly used (Chebyshev and elliptic IIR filters).

# Signal filtering

- Despite IIR filters often being considered as computationally more efficient, they are recommended only when high throughput and sharp cutoffs are required.

- **For offline data analysis, however, throughput and computational time do not matter on modern computer hardware**.

# Signal filtering

- The impulse response is also finite for IIR filters due to numerical precision, thus, **all relevant properties can also be implemented with FIR filters**.

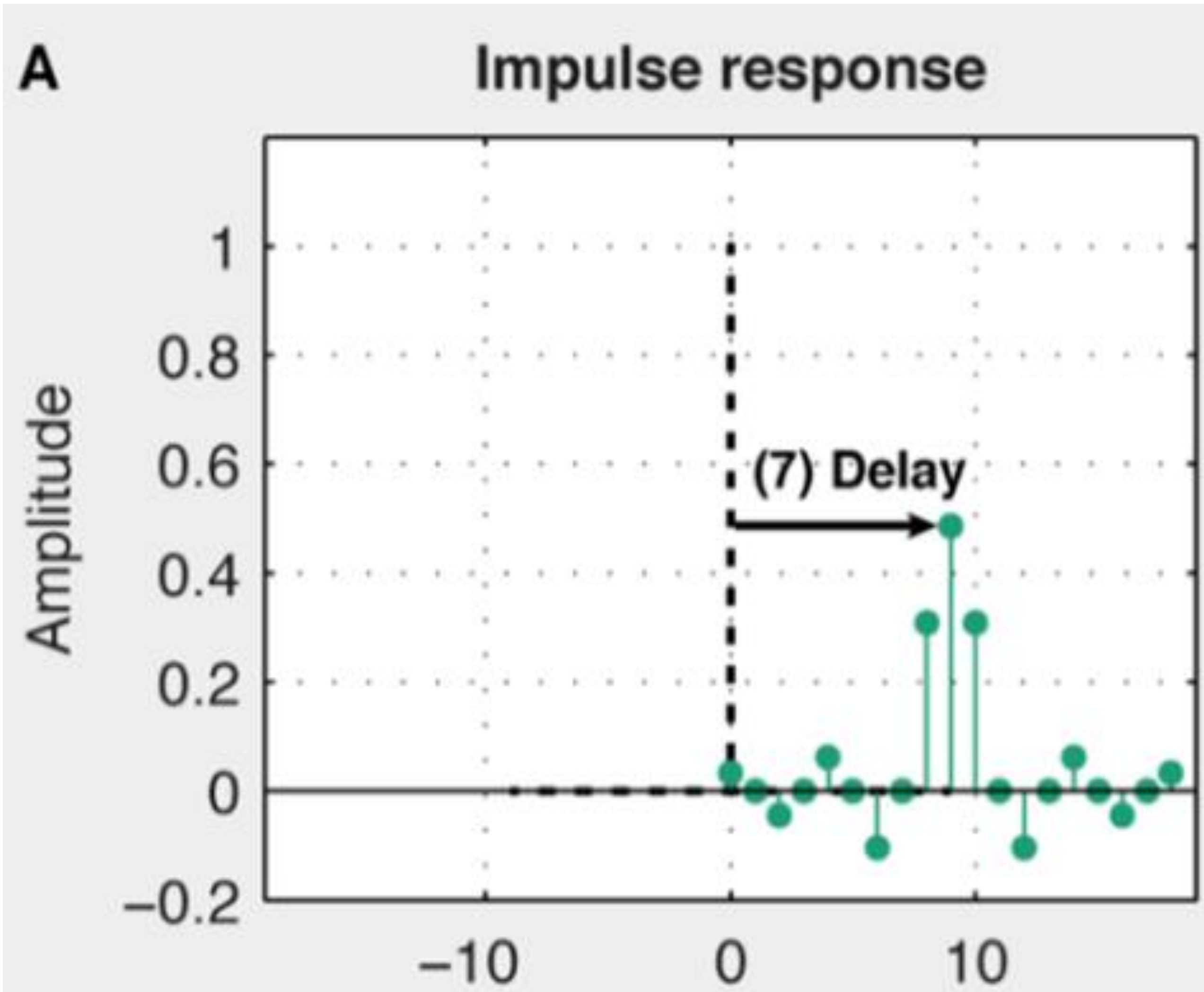
- FIR filters are easier to control, are always stable, have a well-defined passband, can be corrected to zero-phase without additional computations, and can be converted to minimum-phase.

# Filter design – Filter responses

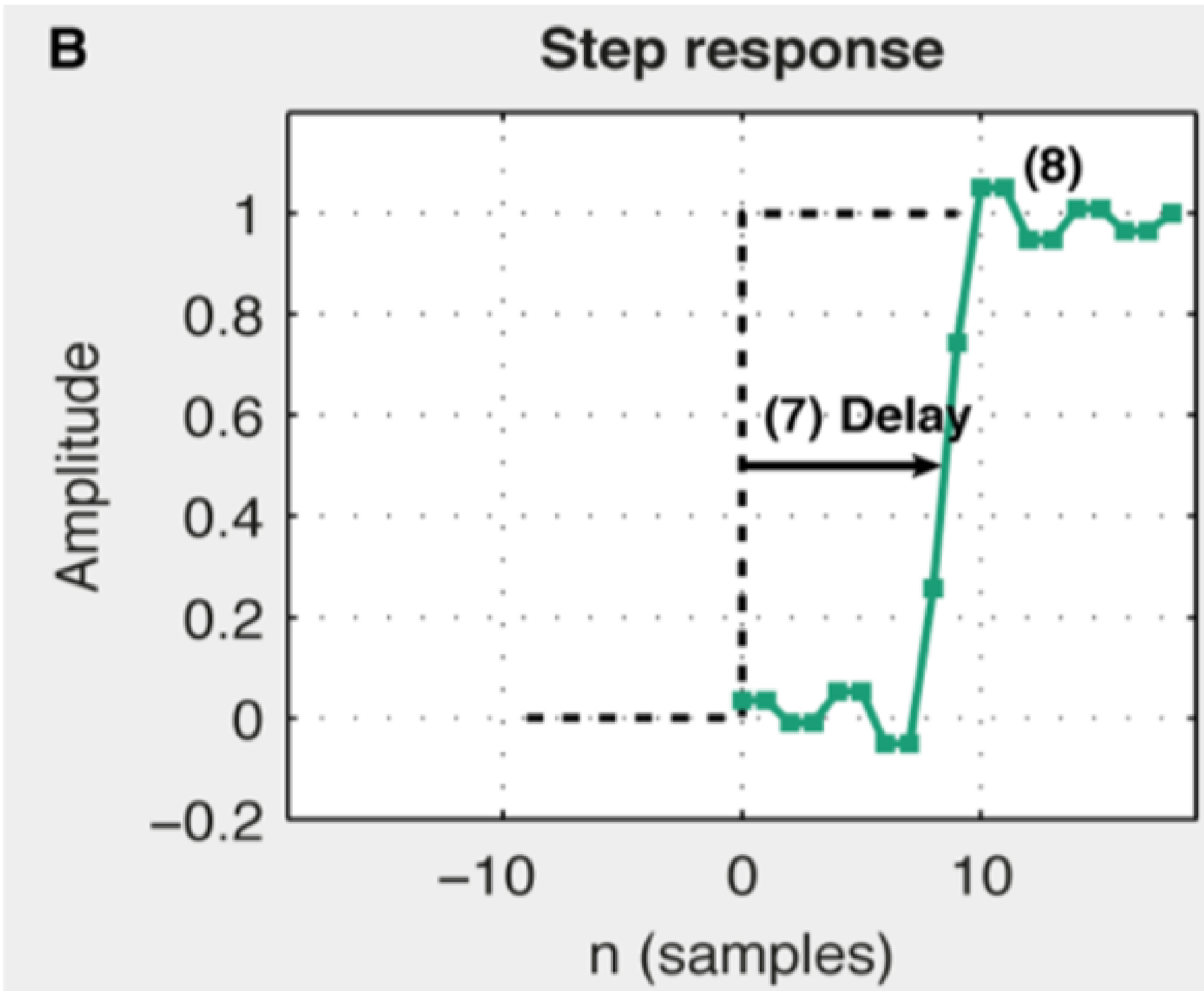
# Filter design – Filter responses

- **As impulse and step signals have energy across the whole spectrum** they are excellent tools to evaluate possible filter distortions when filtering broadband complex signals.

# Filter design – Filter responses

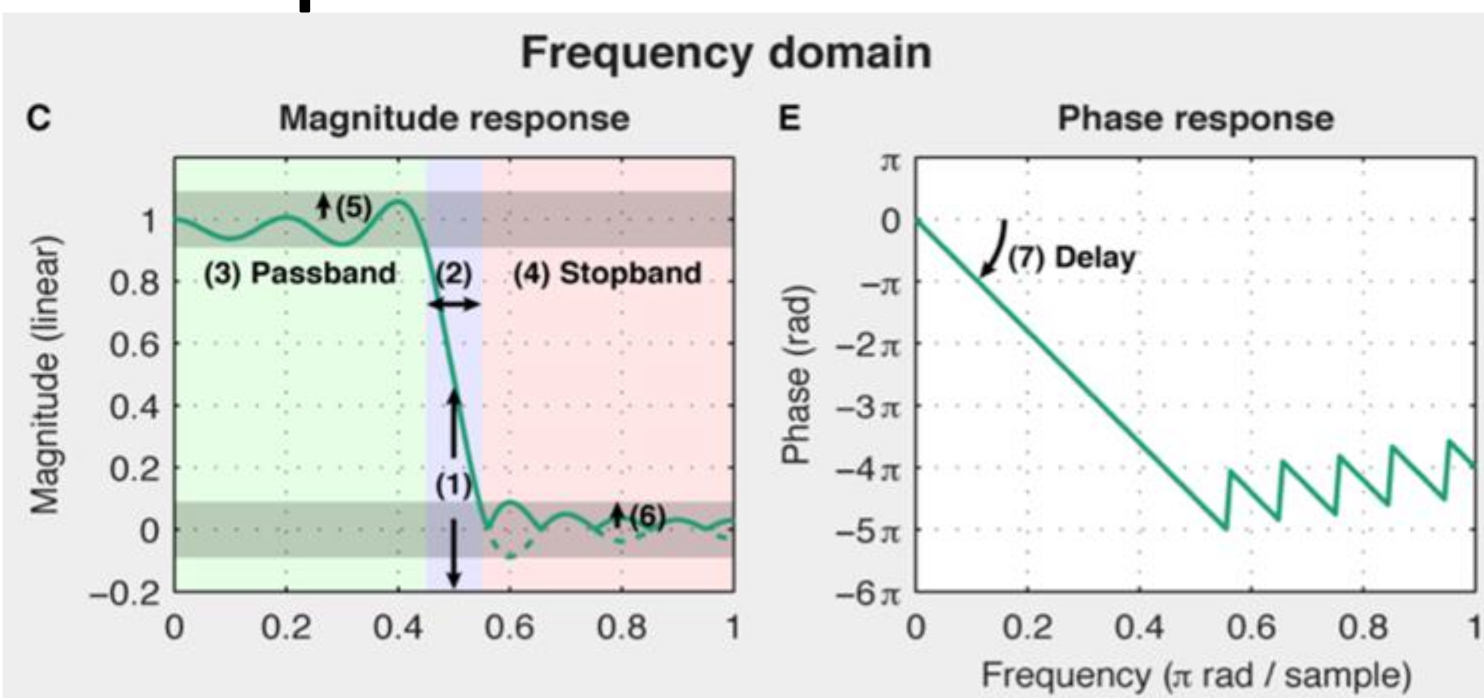


# Filter design – Filter responses



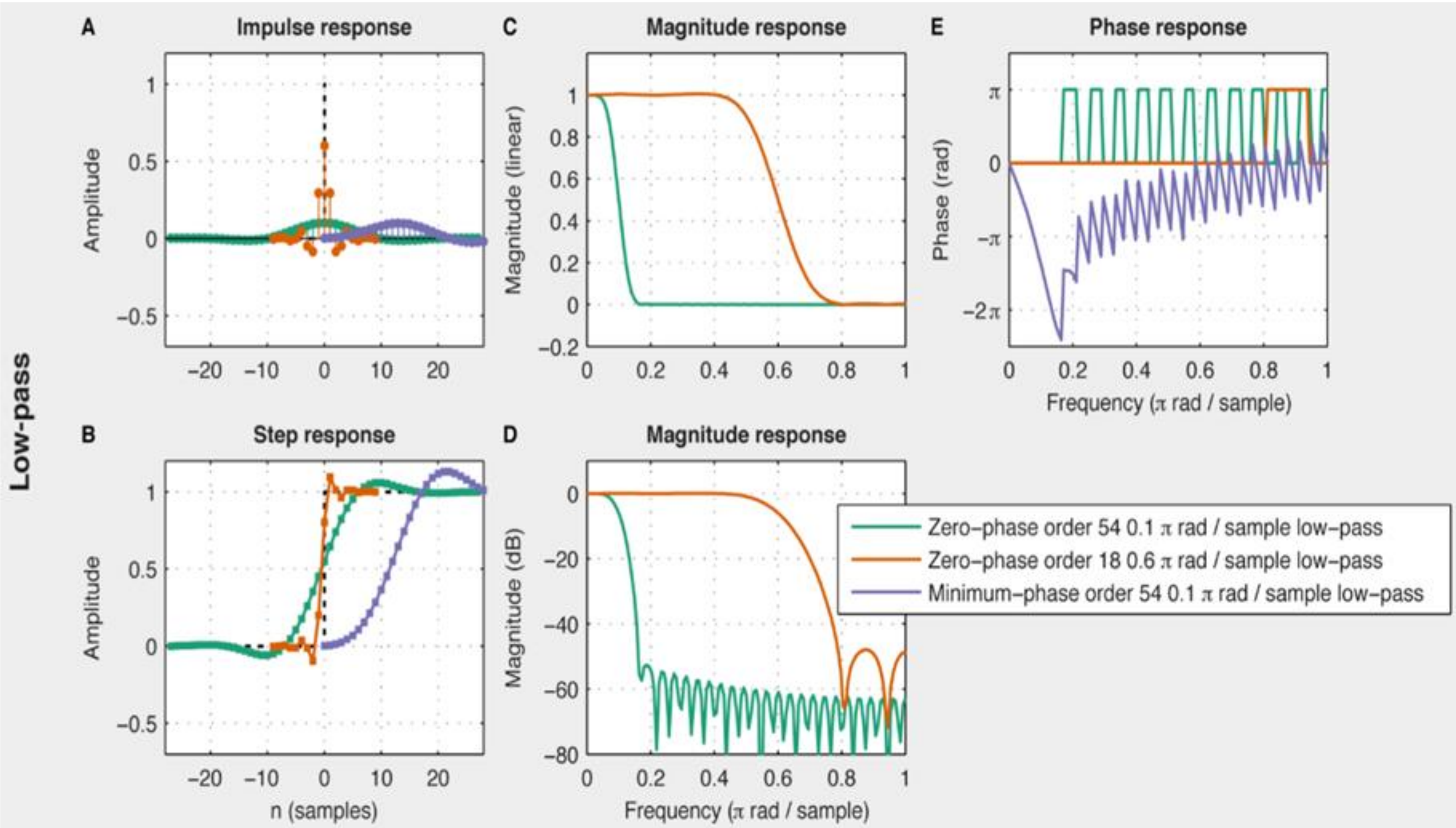
# Filter design – Filter responses

- The frequency response is the Fourier transform of the impulse response and consists of two parts: **magnitude and the phase response**.

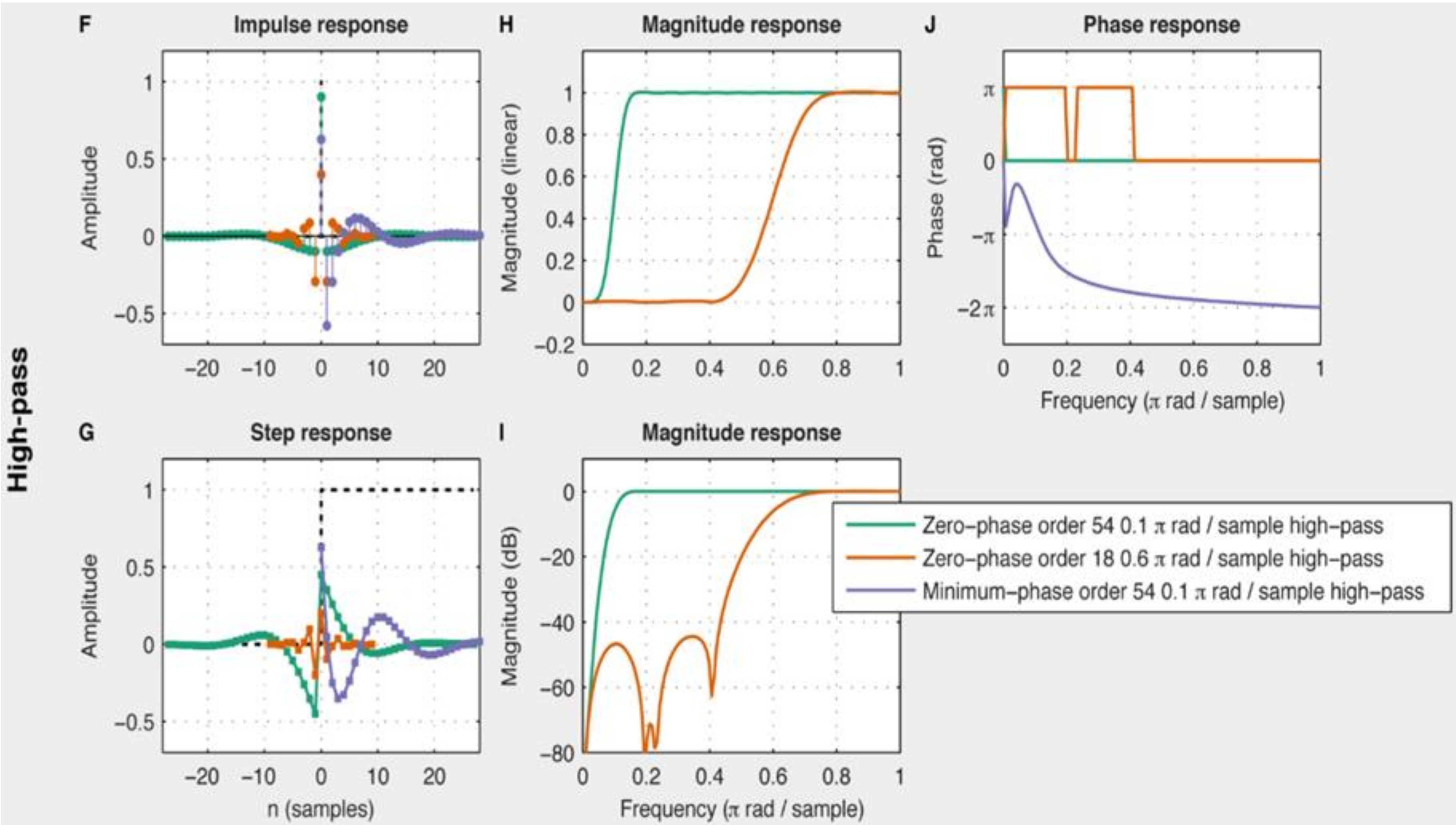


# Filter design – Filter type

# Filter design – Filter type



# Filter design – Filter type



# Filter design – Filter type

- A separate successive application of a steep **high-pass** and a shallow **low-pass** filter is often preferred over a band-pass filter.

- The use of digital **band-stop** filters is not recommended in biosignal research as they likely produce strong artifacts.

# Filter design – Filter type

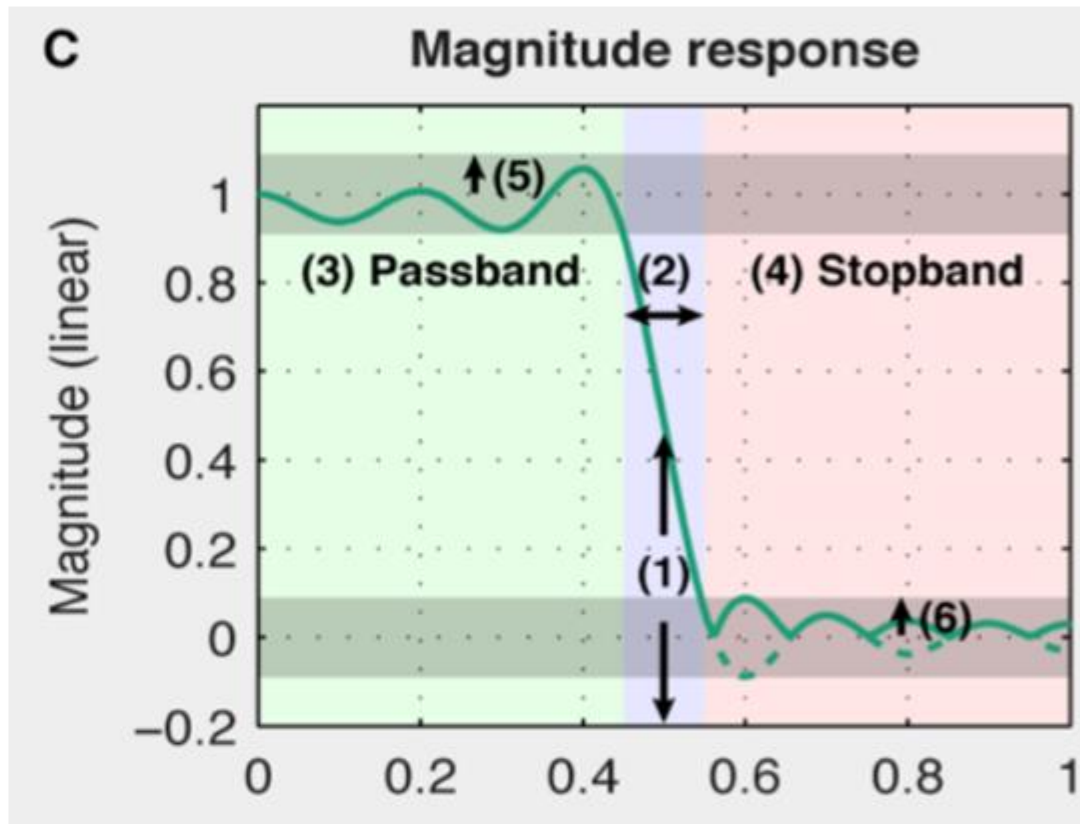
- Band-stop filters are almost exclusively used to suppress line (50/60 Hz) and should be replaced **by time domain regression-based approaches.**

- These approaches are superior due to the very high phase stability of line noise.

# Filter design – Cutoff Frequency

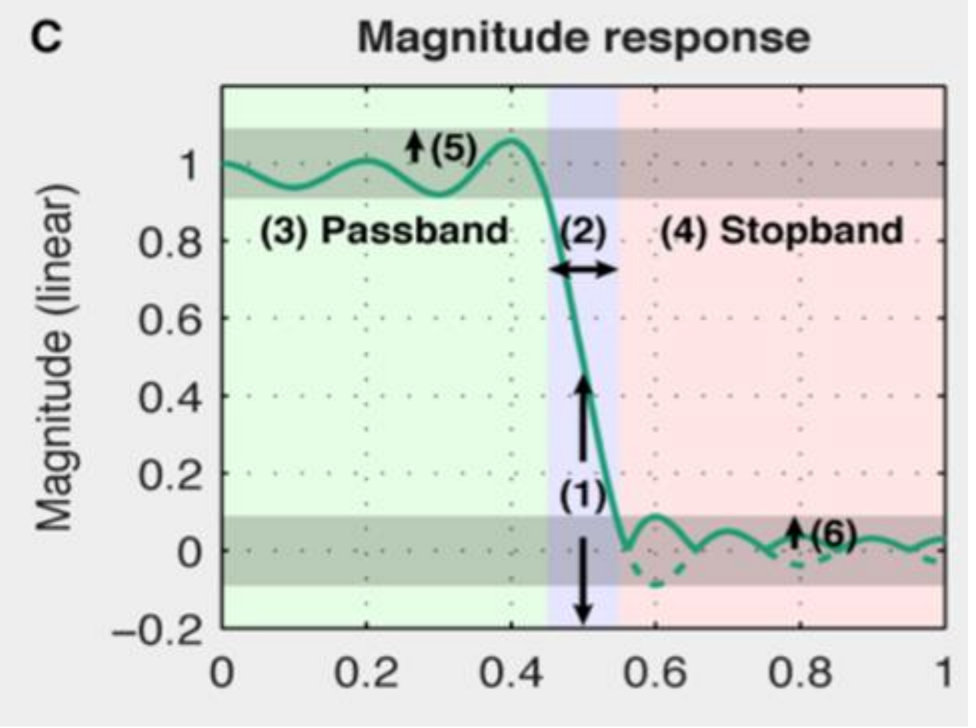
# Filter design – Cutoff Frequency

- The **cutoff frequency** separates passband and stopband of the filter and always lies in the transition band



# Filter design – Cutoff Frequency

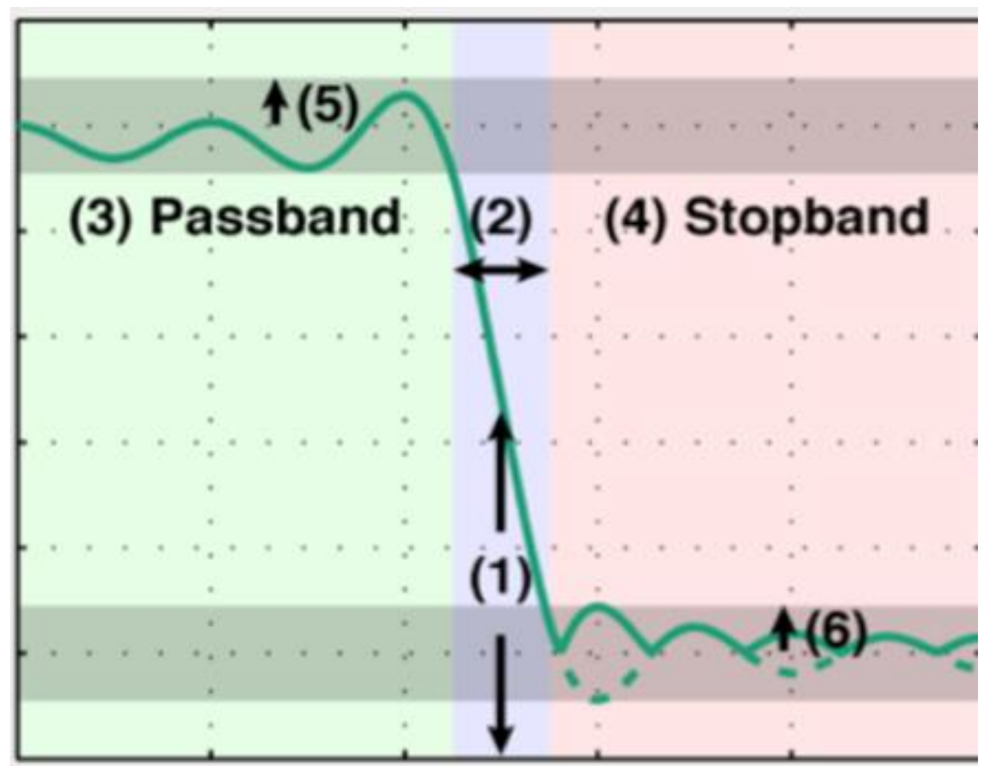
- Different definitions of cutoff frequency are used: **–3 dB (half-energy) cutoff (common for IIR filters) and –6 dB (half amplitude) cutoff (common for FIR).**



# Filter design – Roll-off, transition bandwidth, and filter order

# Filter design – Roll-off, transition bandwidth, and filter order

- The transition region between passband and stopband enclosing the cutoff frequency is defined as the transition band. **For most FIR filters the –6 dB cutoff frequency is at the center of the transition band**.



# Filter design – Roll-off, transition bandwidth, and filter order

- The slope of the magnitude response in the transition band is termed roll-off.

- **The filter roll-off is a function of the filter order.**

# Filter design – Roll-off, transition bandwidth, and filter order

- Longer filters produce stronger signal distortions.

- Thus, shorter filters with wider transition bands are preferable where possible.

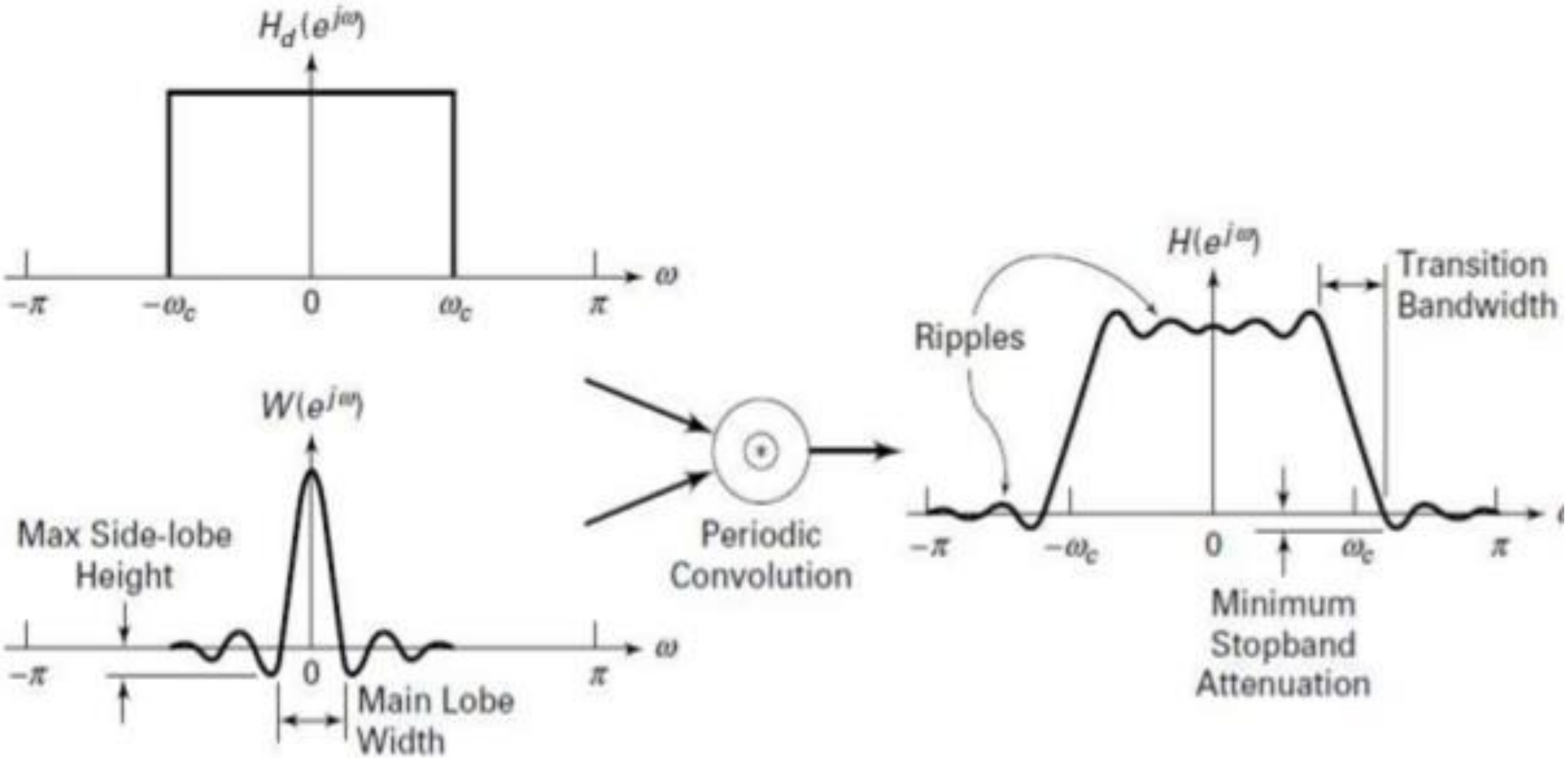
# Filter design – Passband ripple/stopband attenuation

# Filter design – Passband ripple/stopband attenuation

- The practically achieved magnitude response usually **deviates** from the requested magnitude response (**one in the passband and zero in the stopband**).

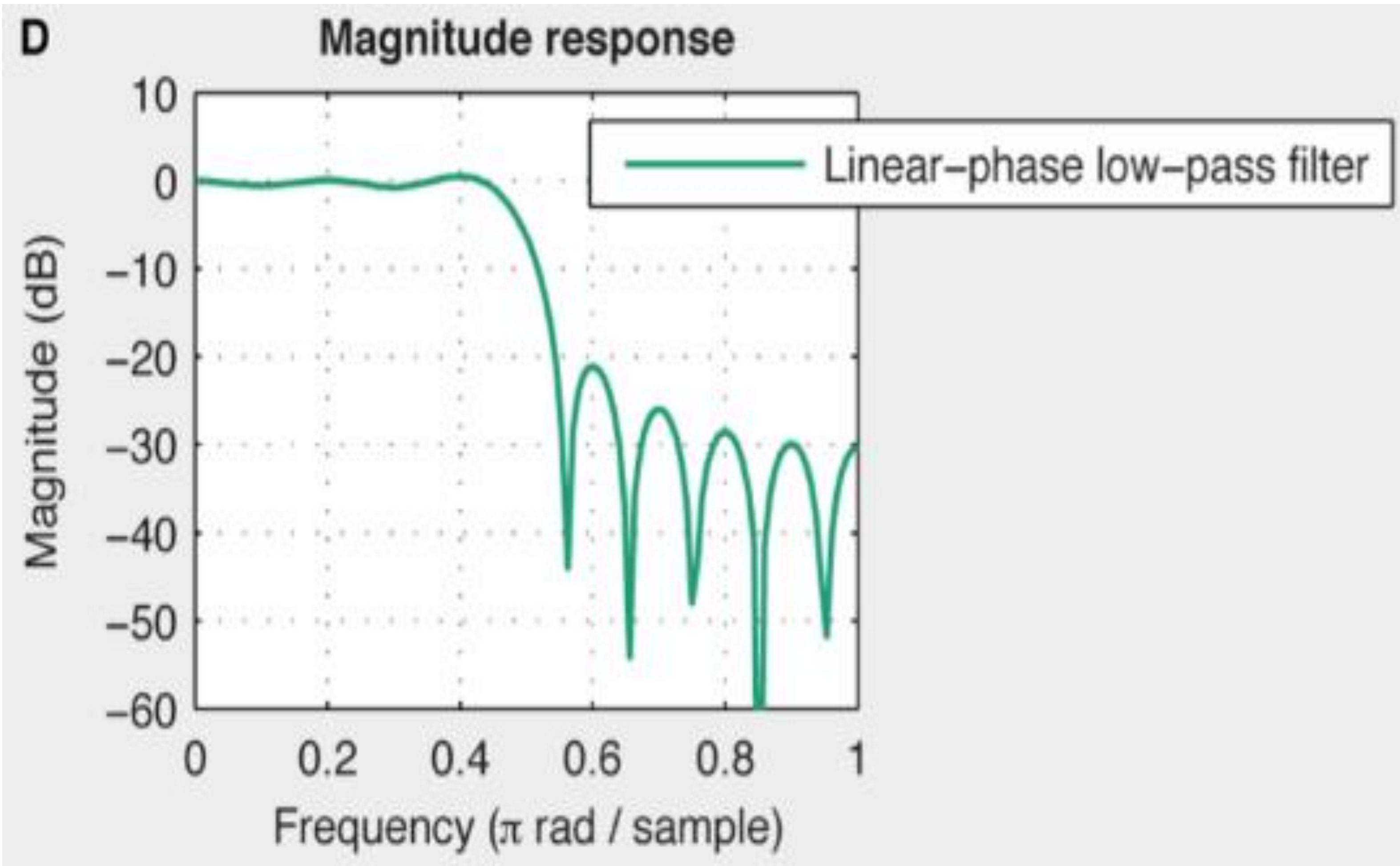
- This deviation is commonly termed **passband ripple** in the passband and **stopband attenuation** in the stopband

# Filter design – Passband ripple/stopband attenuation




*Windowing operation in the frequency domain*

# Filter design – Passband ripple/stopband attenuation



# Filter design – Passband ripple/stopband attenuation

- Passband ripple is reported as maximal passband **deviation** in linear or logarithmic units.

- With a passband deviation of, for example, 0.01, the filter output does not amplify or attenuate the signal by more than 1% in the passband (0.086 dB).

# Filter design – Passband ripple/stopband attenuation

- Stopband attenuation is reported most commonly in logarithmic units.

- With a stopband attenuation of –60 dB (or 0.001), the signal is **attenuated** by a factor of 1000 in the stopband.

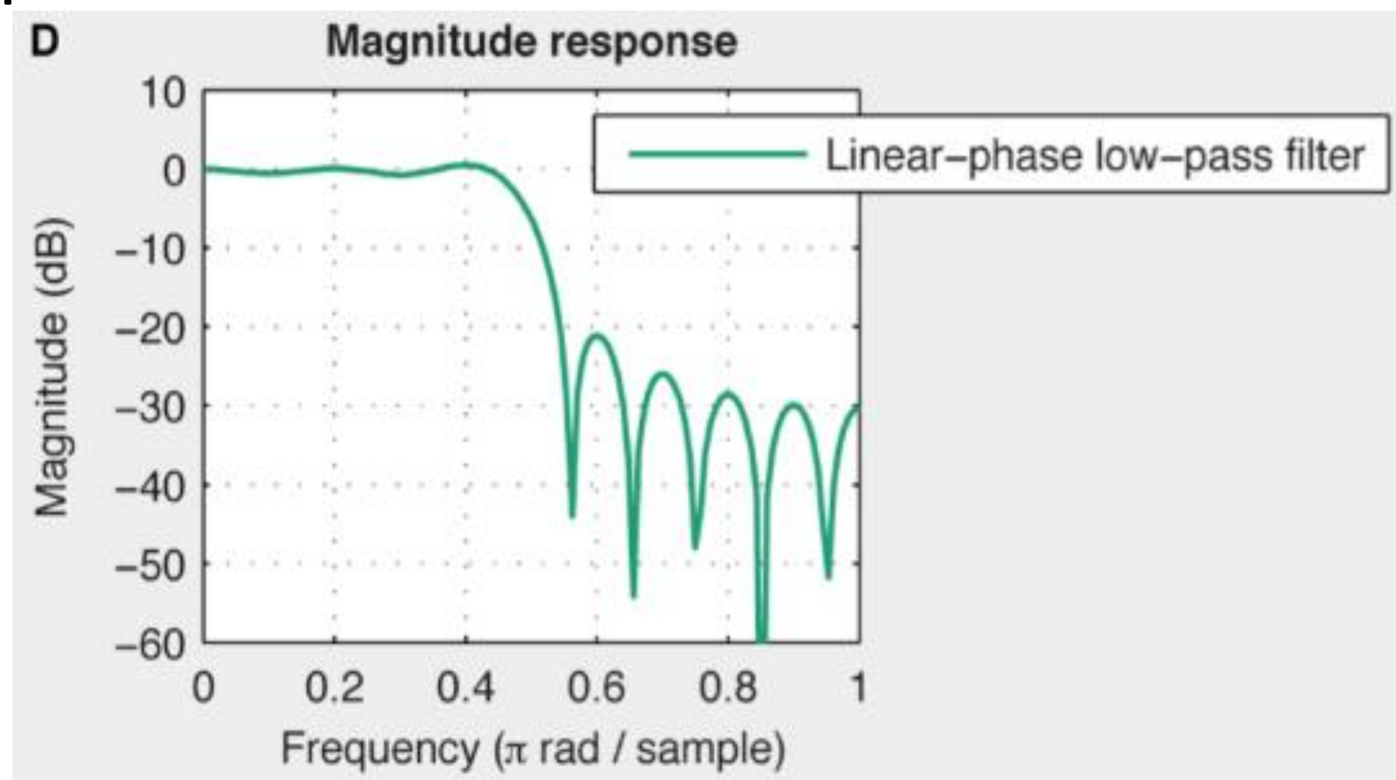
# Filter design – Passband ripple/stopband attenuation

- For instance, passband ripple of 0.002–0.001 (0.2%–0.1%) and −54 to −60 dB stopband attenuation are reasonable values for biosignal applications.

- For high amplitude low-frequency noise (near DC), a stopband attenuation of −100 dB or stronger might be required.

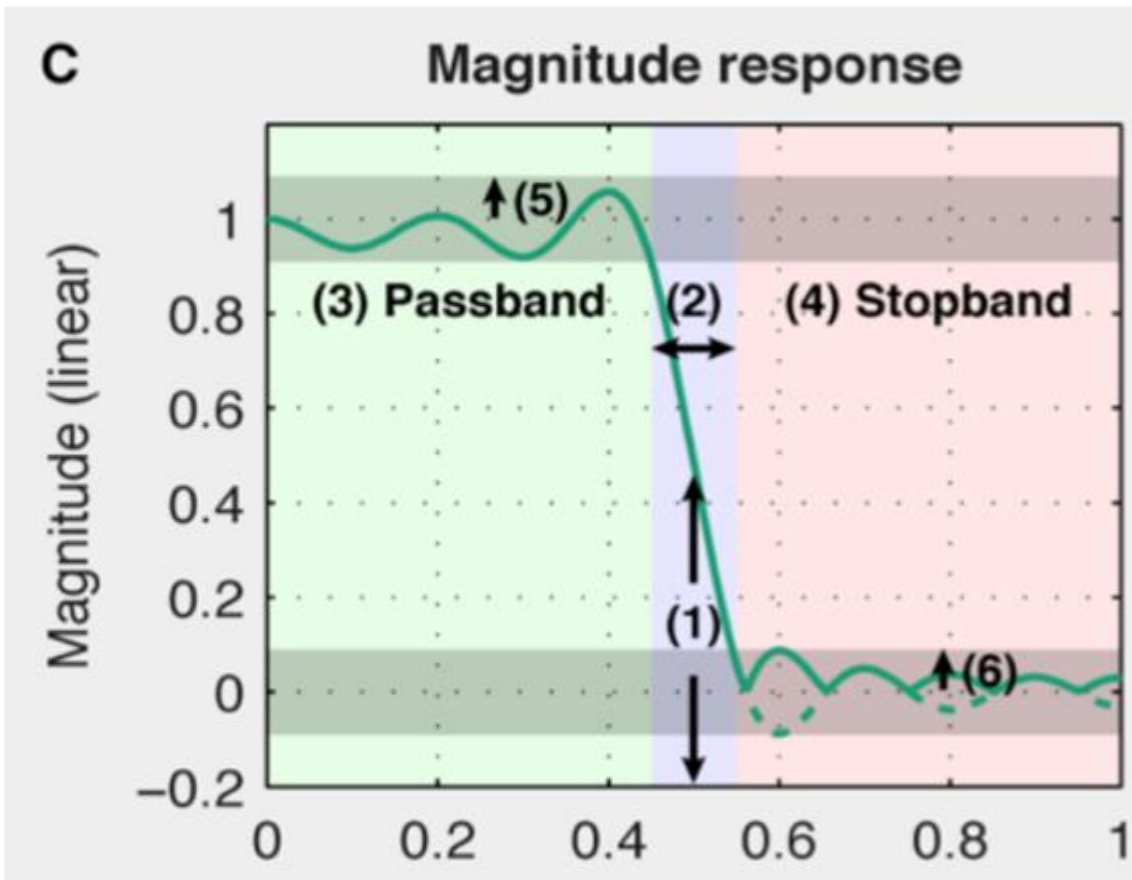
# Filter design – Passband ripple/stopband attenuation

- Stopband ripple/attenuation is best evaluated in the logarithmically scaled magnitude response



# Filter design – Passband ripple/stopband attenuation

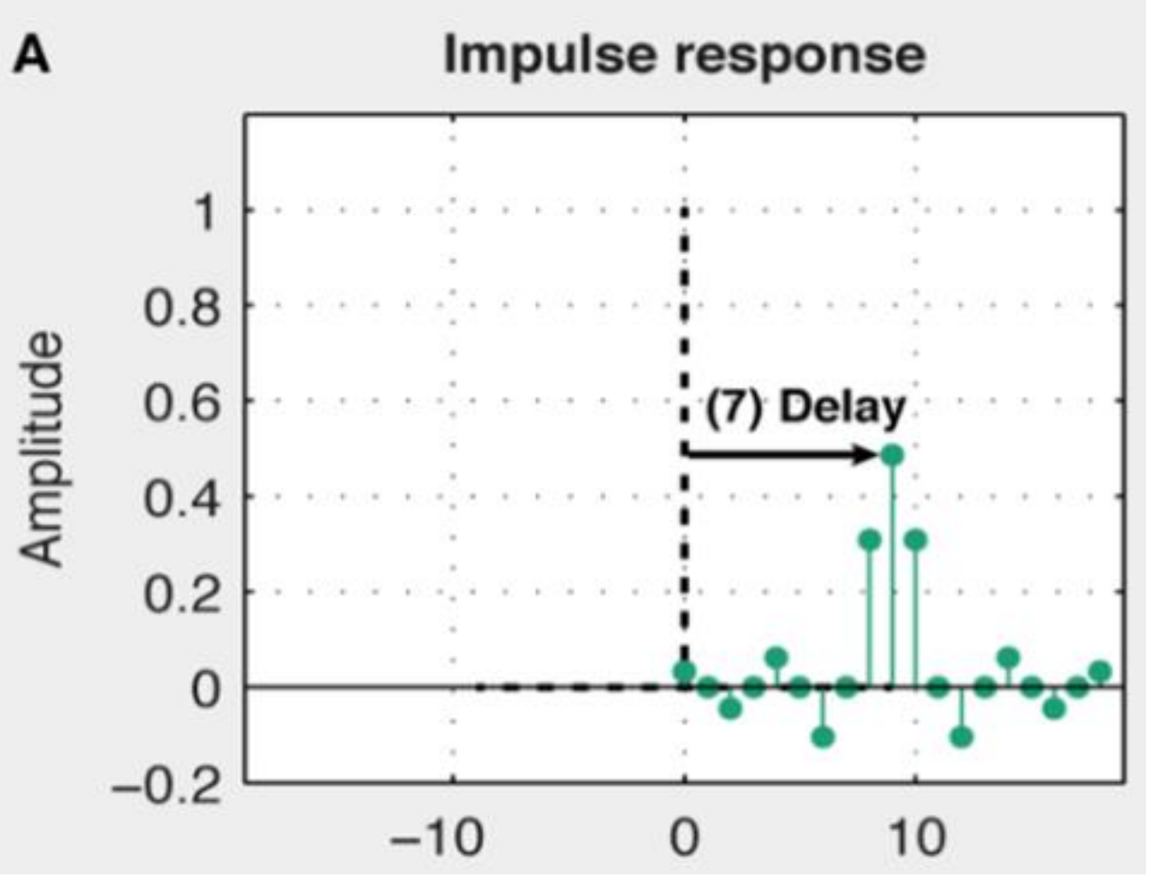
- Passband ripple is better evaluated in the linearly scaled magnitude response



# Filter design – Delay

# Filter design – Delay

- Every (non-trivial) filter necessarily delays the filter output relative to the filter input



# Filter design – Delay

- Linear-phase filters introduce an equal (group) delay at all frequency bands – the slope of the phase response is constant within the passband.

- Consequently, a signal with all its spectral components in the passband **will not change its temporal shape**.

# Filter design – Delay

- The group delay of linear-phase filters can be easily computed based on the length of the filter's impulse response as $(N - 1) / 2$ (in samples).

# Filter design – FIR Filters

# Filter design – FIR Filters

- There are 4 types of FIR filters with linear phase, i.e. constant group delay (M = length of impulse response)

1. Impulse response symmetrical, M = odd
2. Imp. resp. symmetrical, M = even
3. Imp. resp. anti-symmetrical, M = odd
4. Imp. resp. anti-symmetrical, M = even

# Filter design – FIR

- In electrophysiology almost exclusively **odd length**, symmetric (type I) FIR filters are applied (only odd-length FIR filters can be corrected to zero-phase delay by left-shifting).

# Filter design – IIR vs FIR

- A FIR filter has linear phase if its impulse response satisfies the condition of symmetry or asymmetry of its coefficients

$$\boldsymbol{G(w) = \sum_{k=0}^{2N} g_k e^{-jkwT}}$$

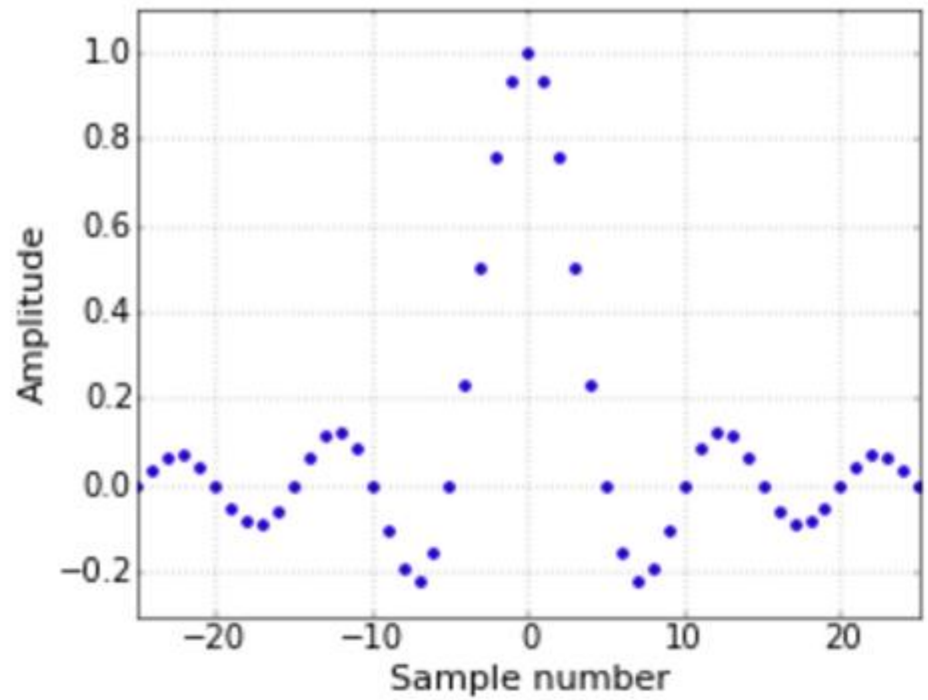
The symmetry/asymmetry is defined as:

$$\boldsymbol{g_0 = \pm g_{2N}}$$
$$\boldsymbol{g_1 = \pm g_{2N-1}}$$
$$\dots$$
$$\boldsymbol{g_{N-1} = \pm g_{N+1}}$$

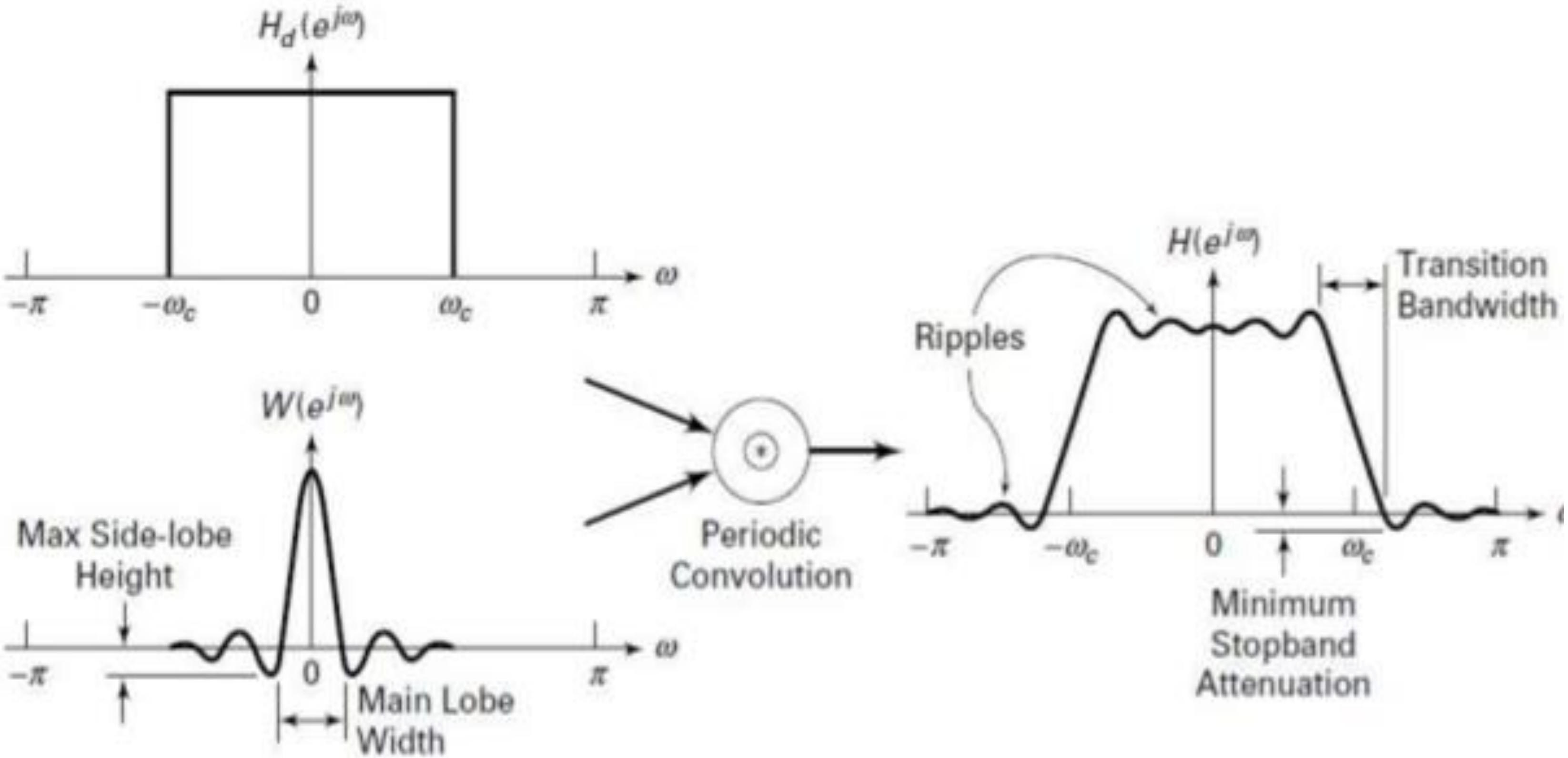
# Filter design – FIR

- Windowed sinc FIR filters are based on the sinc-function approximating a rectangular magnitude response, thus, sometimes termed “ideal” filters.

$$h[n] = 2f_c \mathrm{sinc}(2f_c n)$$



# Filter design – FIR

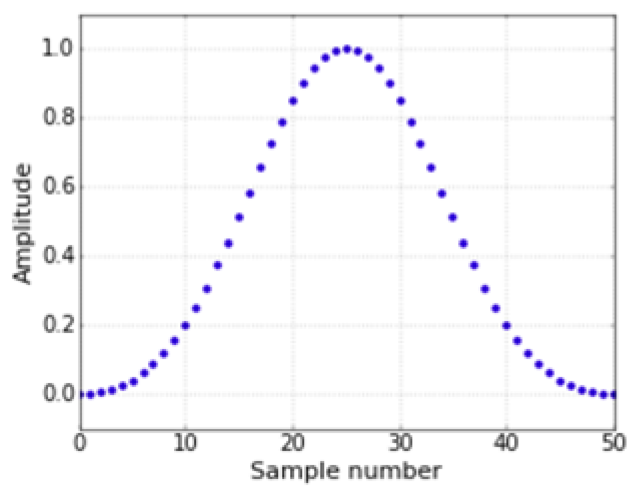



*Windowing operation in the frequency domain*

# Filter design – FIR

- For finite filter orders, **the impulse response has to be windowed** by a window function to reduce passband and stopband ripple (equal for windowed sinc filters).

$$w[n] = 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right)$$



# Filter design – FIR

- The transition bandwidth is a function of filter order (filter length minus one) and window type.

- The requested transition bandwidth as well as passband and stopband ripple determine the filter order

# Filter design – FIR

**Table 1**
Properties of selected window types for windowed sinc FIR filters (adapted from Ifeachor and Jervis, 2002, p. 357). Required filter order m for requested transition bandwidth $\Delta f$ can be computed as m = $\Delta F / (\Delta f / f_s)$. Transition bandwidth $\Delta f$ provided by a filter order m is computed as $\Delta f = (\Delta F / m) f_s$ (Smith, 1999).

| Window type | Beta | Stopband attenuation (dB) | Max. passband deviation | Normalized transition width $\Delta F$ |
|---|---|---|---|---|
| Rectangular | | −21 | 0.0891 (8.91%) | 0.9 / m |
| Hann | | −44 | 0.0063 (0.63%) | 3.1 / m |
| Hamming | | −53 | 0.0022 (0.22%) | 3.3 / m |
| Blackman | | −75 | 0.0002 (0.02%) | 5.5 / m |
| Kaiser | 5.65 | −60 | 0.001 (0.1%) | 3.6 / m |
| Kaiser | 7.85 | −80 | 0.0001 (0.01%) | 5.0 / m |

# Filter design – FIR

| Ventana | Anchura del lóbulo ppal de la ventana | Anchura de la banda de transición del filtro diseñado $\Delta\omega$ | Pico Lóbulo secundario de la ventana(dB) | Atenuación del filtro diseñado con esta ventana Rs(dB) |
|---|---|---|---|---|
| Rectangular | $\frac{4\pi}{N}$ | $\frac{1.8\pi}{N}$ | -13 | -21 |
| Bartlett (triangular) | $\frac{8\pi}{N}$ | $\frac{6.1\pi}{N}$ | -25 | -25 |
| Von Hann (Hanning) | $\frac{8\pi}{N}$ | $\frac{6.2\pi}{N}$ | -31 | -44 |
| Hamming | $\frac{8\pi}{N}$ | $\frac{6.6\pi}{N}$ | -41 | -53 |
| Blackman | $\frac{12\pi}{N}$ | $\frac{11\pi}{N}$ | -57 | -74 |

# Filter design – FIR

- From the low-pass to the high-pass

$$x_{\text{lpf}}[n] = x[n] * h_{\text{lpf}}[n]$$

The high-pass signal is obtained:

$$x_{\text{hpf}}[n] = x[n] - x_{\text{lpf}}[n]$$

$$\begin{aligned} x_{\text{hpf}}[n] = x[n] - x_{\text{lpf}}[n] &= x[n] * \delta[n] - x[n] * h_{\text{lpf}}[n] \\ &= x[n] * (\delta[n] - h_{\text{lpf}}[n]) \end{aligned}$$

Then: $$h_{\text{hpf}}[n] = \delta[n] - h_{\text{lpf}}[n]$$

# Filter design – FIR

Band-pass filter

We can obtain the band-pass filtered signal as:

$$x_{\mathrm{bp,LH}}[n] = (x[n] * h_{\mathrm{lpf,H}}[n]) * h_{\mathrm{hpf,L}}[n] = x[n] * (h_{\mathrm{lpf,H}}[n] * h_{\mathrm{hpf,L}}[n])$$

Then:

$$h_{\mathrm{bp,LH}}[n] = h_{\mathrm{lpf,H}}[n] * h_{\mathrm{hpf,L}}[n]$$

# Filter design – FIR

Band-reject filter

We can obtain the band-reject filtered signal as:

$$x_{\mathrm{br,LH}}[n] = x[n] * h_{\mathrm{lpf,L}}[n] + x[n] * h_{\mathrm{hpf,H}}[n] = x[n] * (h_{\mathrm{lpf,L}}[n] + h_{\mathrm{hpf,H}}[n]$$

Then:

$$h_{\mathrm{br,LH}}[n] = h_{\mathrm{lpf,L}}[n] + h_{\mathrm{hpf,H}}[n]$$

# Implementation

- Hamming windows and fixed maximum transition bandwidth

```
#Constants
TRANSWIDTHRATIO = 0.25;
fNyquist = srate/2;
```

```
filtorder = 3.3 / (df / srate); # Hamming window
filtorder = np.ceil(filtorder / 2) * 2; # Filter order must be even.
```

```
# Window
winArray = signal.hamming(int(filtorder) + 1);
```

# Implementation

- The impulse response of the ideal filter is truncated by the order of the filter and windowed

```
# Compute filter kernel
def fkernel(m, f, w):
    m = np.arange(-m/2, (m/2)+1)
    b = np.zeros((m.shape[0]))
    b[m==0] = 2*np.pi*f # No division by zero
    b[m!=0] = np.sin(2*np.pi*f*m[m!=0]) / m[m!=0] # Sinc
    b = b * w # Windowing
    b = b / np.sum(b) # Normalization to unity gain at DC
    return b
```

# Implementation

- Being the filter a system is possible to obtain its impulse response in frequency domain

```
w,h = signal.freqz(b,a);
h_dB = 20 * np.log10 (abs(h));

plt.figure();
plt.subplot(311);
plt.plot((w/max(w))*nyq_rate,abs(h));
```

magnitude

```
plt.ylabel('Magnitude');
plt.xlabel(r'Normalized Frequency (x$\pi$rad/sample)');
plt.title(r'Frequency response. Order: ' + str(order));
[xmin, xmax, ymin, ymax] = plt.axis();
plt.grid(True);

plt.subplot(312);
plt.plot((w/max(w))*nyq_rate,h_dB);
```

magnitude (dB)

```
plt.ylabel('Magnitude (db)');
plt.xlabel(r'Normalized Frequency (x$\pi$rad/sample)');
plt.title(r'Frequency response. Order: ' + str(order));
plt.grid(True)
plt.grid(True)


plt.subplot(313);
```

phase

```
h_Phase = np.unwrap(np.arctan2(np.imag(h),np.real(h)));
plt.plot((w/max(w))*nyq_rate,h_Phase);
plt.ylabel('Phase (radians)');
plt.xlabel(r'Normalized Frequency (x$\pi$rad/sample)');
plt.title(r'Phase response. Order: ' + str(order));
```

# Implementation

- Double filtering of the signal, forward and reverse, to implement a non-causal filter (only offline!!)

```
#plot the response of the filter
mfreqz(b,1,filtorder, fNyquist);

signal_filtered = signal.filtfilt(b, 1, senal);#
return signal_filtered;
```