Time Series Analysis:

4. Linear filters

P. F. Góra

http://th-www.if.uj.edu.pl/zfs/gora/

2012

Linear filters in the Fourier domain

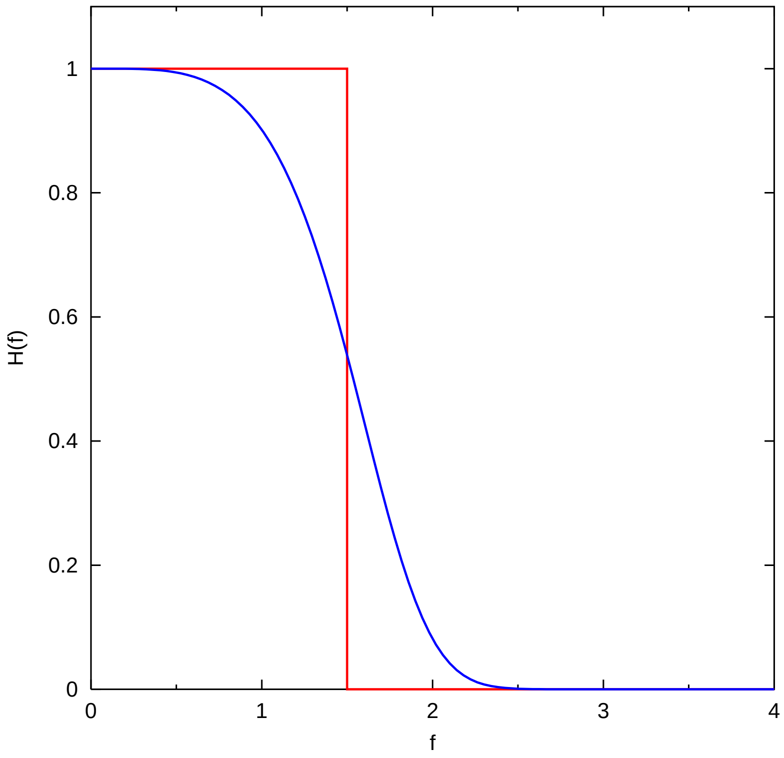
Filtering: Multiplying the transform by a *transfer function*.

$$g_n \xrightarrow[\text{DFT}]{} G_n \longrightarrow \mathcal{H}(f_n)G_n \xrightarrow[\text{inverse DFT}]{} \tilde{g}_n \tag{1}$$

where $\mathcal{H}(f_n)$ is the transfer function discretized over f_n.

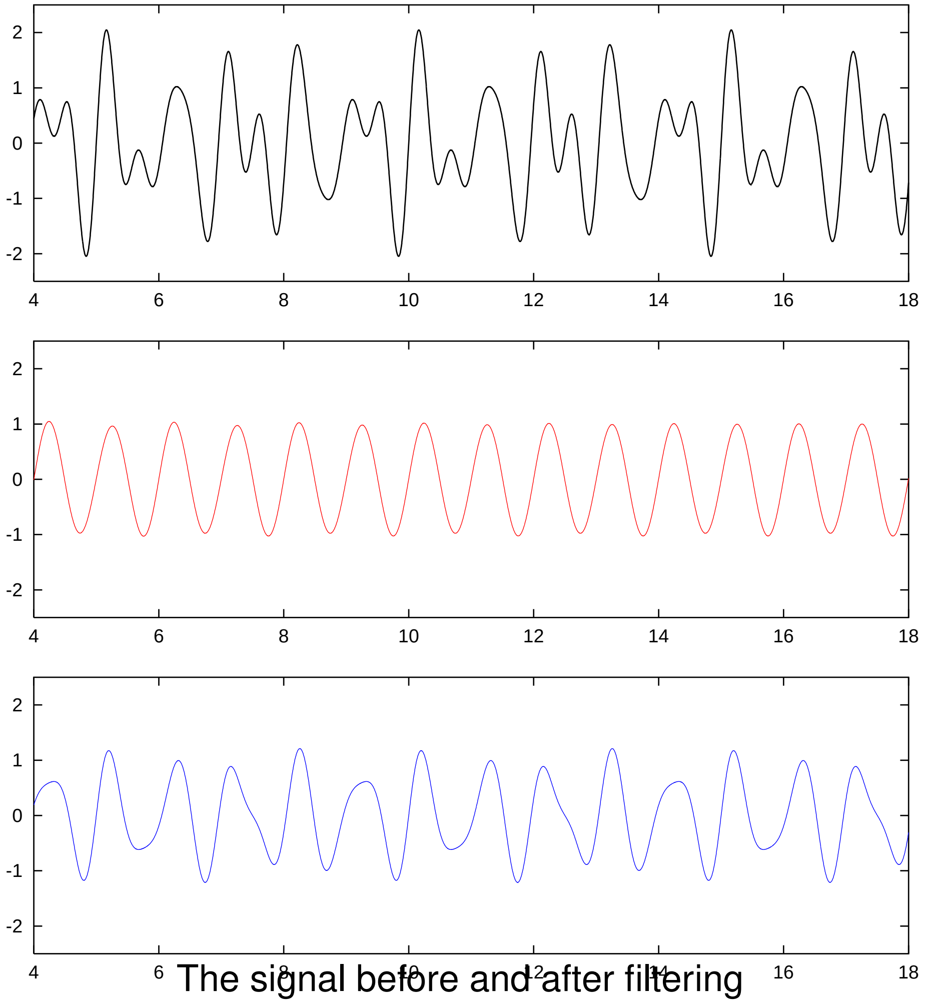
The most important drawback: You need to know the whole series, collect all data, *prior* to filtering.

Example



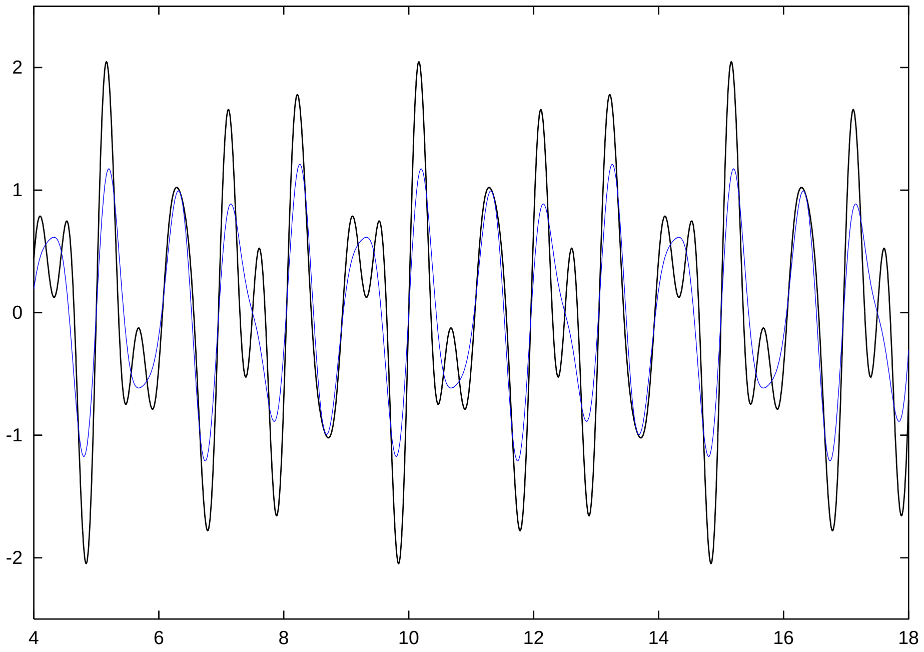
$$\mathcal{H}(f) = \begin{cases} 1 & |f| \leqslant f_c \\ 0 & |f| > f_c \end{cases}$$

$$\mathcal{H}(f) = \frac{2\exp\left(-(f/f_c)^4\right)}{1+\exp\left(-(f/f_c)^4\right)}$$



The signal before and after filtering

A reason for the strange result of the “blue” filtering



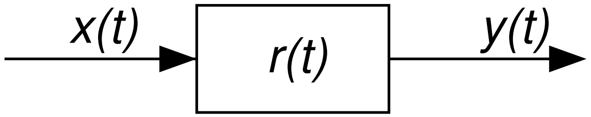
When filtering in the Fourier domain, we usually do not smooth out the filters.

Linear filters

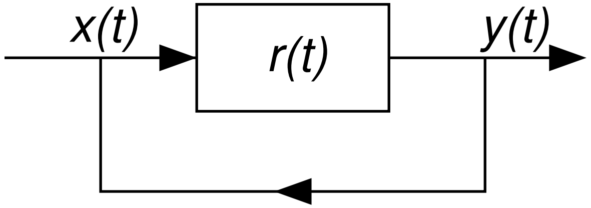
Filtering *in Fourier domain* is very easy: multiply the DFT of the input by the transfer function.

Filtering *in the signal domain* can be tricky:

a FIR filter:



an IIR filter:



The transfer function

Even though the filters are realized in the signal domain, it is convenient to analyse them in Fourier domain. Every possible *linear* filter is represented by

$$Y(f) = \mathcal{H}(f) X(f) \,. \tag{2}$$

$\mathcal{H}(f)$ is the *transfer function*.

Terminology

- $\mathcal{H}(f) = 0$ for $|f| \geqslant f_0$ — a low-pass filter.
- $\mathcal{H}(f) = 0$ for $|f| \leqslant f_0$ — a high-pass filter.
- $\mathcal{H}(f) \neq 0$ for $f_1 \leqslant |f| \leqslant f_2$ — a band-pass filter.
- $\mathcal{H}(f) = 0$ for $f_1 \leqslant |f| \leqslant f_2$ — a band-stop filter; if $f_2 - f_1$ is small, also known as a notch filter.

A general linear filter in the signal domain has the form

$$y_n = \sum_{k=-s}^{q} \alpha_k x_{n-k} + \sum_{k=1}^{p} \beta_k y_{n-k} \tag{3}$$

x_n is a (discretized) input signal, y_n is the output signal.

If $s > 0$, *future* values of the input are needed to construct the output. Such filter is *non-causal*. It cannot be realized on-line (or in real time).

Eq. (3):

$$y_n = \sum_{k=-s}^{q} \alpha_k x_{n-k} + \sum_{k=1}^{p} \beta_k y_{n-k}$$

- If $p = 0$, the filter is a *Finite Impulse Response* filter (FIR), or a *moving average* filter. The output of such a filter dies out in a finite time after the input has died out.

- If $p > 0$, the filter is *Infinite Impulse Response* filter (IIR), or an *autoregressive* filter. Its output can go on infinitely long after the input has died out (in fact, this is a parasitic behaviour).

Causal FIR filters

$$y_n = \sum_{k=0}^{q} \alpha_k x_{n-k} \,. \tag{4}$$

If the input is stationary, the output also is. q is called *the order of the filter*.

To find the transfer function, we Fourier transform Eq. (4).

$$
\begin{aligned}
Y_m &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} e^{2\pi i m n/N} y_n = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} e^{2\pi i m n/N} \sum_{k=0}^{q} \alpha_k x_{n-k} \\
&= \sum_{k=0}^{q} \alpha_k \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} e^{2\pi i m n/N} x_{n-k} = \sum_{k=0}^{q} \alpha_k \frac{1}{\sqrt{N}} \sum_{n'=-k}^{N-1-k} e^{2\pi i m (n'+k)/N} x_{n'} \,.
\end{aligned}
\tag{5}
$$

By the assumption of periodicity of the input, $x_{-l} \equiv x_{N-l}$. On the other hand, $\exp(2\pi i m(N-l)/N) = \exp(2\pi i m) \exp(2\pi i m(-l)/N) = \exp(2\pi i m(-l)/N)$. Thus

$$Y_m = \sum_{k=0}^{q} \alpha_k e^{2\pi i m k/N} \underbrace{\frac{1}{\sqrt{N}} \sum_{n'=0}^{N-1} e^{2\pi i m n'/N} x_{n'}}_{X_m} = \underbrace{\sum_{k=0}^{q} \alpha_k e^{2\pi i m k/N}}_{\mathcal{H}_m} X_m \,. \quad (6)$$

$m/N = m/(N\Delta)\,\Delta = f_m \Delta$, where f_m is the m-th discrete Fourier frequency. Therefore. . .

Transfer function of a causal FIR filter

has the form

$$\mathcal{H}(f_m) = \sum_{k=0}^{q} \alpha_k \left(e^{2\pi i f_m \Delta}\right)^k = \alpha\left(e^{2\pi i f_m \Delta}\right), \tag{7}$$

where $\alpha(z)$ is the following polynomial of order q

$$\alpha(z) = \sum_{k=0}^{q} \alpha_k z^k \tag{8}$$

(coefficients of the filter become coefficients of the polynomial (8)).

Usually, for the sake of simplicity, we assume that $\Delta = 1$, or that the sampling time is the time unit. Remember: With this notation, frequencies become dimensionless and the Nyquist interval equals $[-1/2, 1/2]$.

Transfer function of a general FIR filter

The above can be easily generalized to arbitrary (non-causal) FIR filters. The transfer function becomes

$$\mathcal{H}(f_m) = \sum_{k=-s}^{q} \alpha_k \left(e^{2\pi i f_m \Delta}\right)^k = \alpha\left(e^{2\pi i f_m \Delta}\right), \tag{9}$$

where $\alpha(\cdot)$ is now an appropriate rational function.

Simple low- and high-pass filters

Consider a filter

$$y_n = \frac{1}{4}x_{n-1} + \frac{1}{2}x_n + \frac{1}{4}x_{n+1}\,. \tag{10}$$

Its transfer function reads

$$\mathcal{H}_c(f) = \frac{1}{4}e^{2\pi i f} + \frac{1}{2} + \frac{1}{4}e^{-2\pi i f} = \frac{1}{2} + \frac{1}{2}\cos 2\pi f = \cos^2 \pi f\,. \tag{11}$$

Similarly, the transfer function of the filter

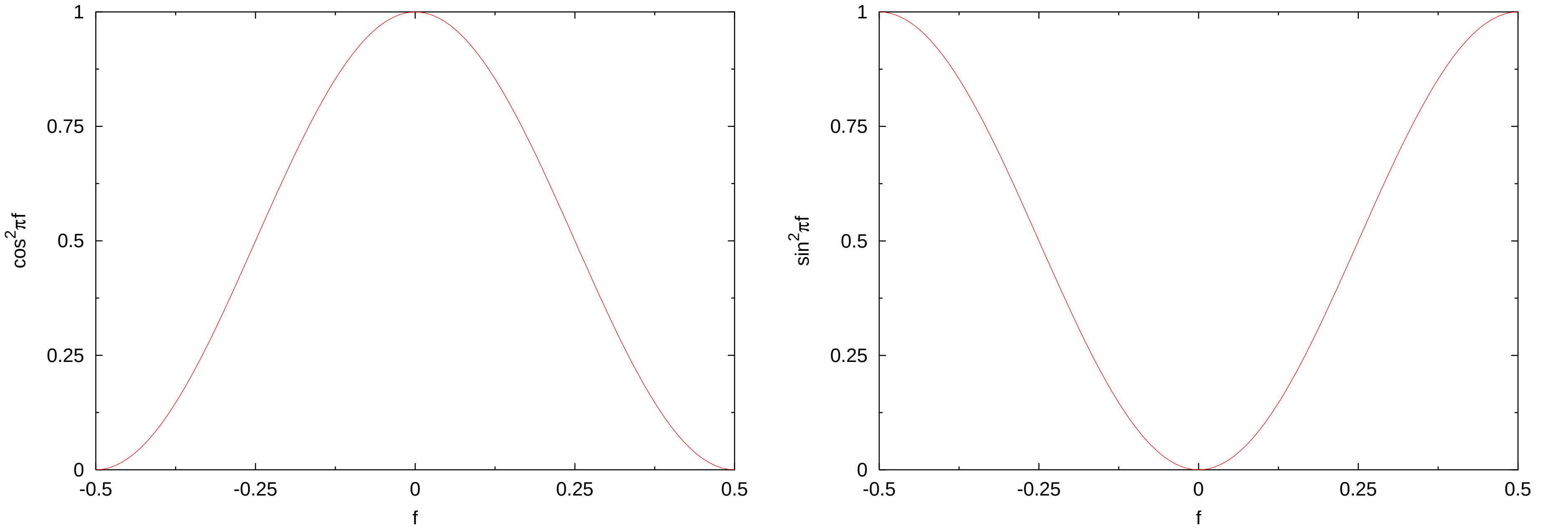
$$y_n = -\frac{1}{4}x_{n-1} + \frac{1}{2}x_n - \frac{1}{4}x_{n+1}\,. \tag{12}$$

reads

$$\mathcal{H}_s(f) = -\frac{1}{4}e^{2\pi i f} + \frac{1}{2} - \frac{1}{4}e^{-2\pi i f} = \frac{1}{2} - \frac{1}{2}\cos 2\pi f = \sin^2 \pi f\,. \tag{13}$$

(10) is a (poor) low-pass filter.

(12) is a (poor) high-pass filter.



The “usual” moving average

The (unweighted) moving average

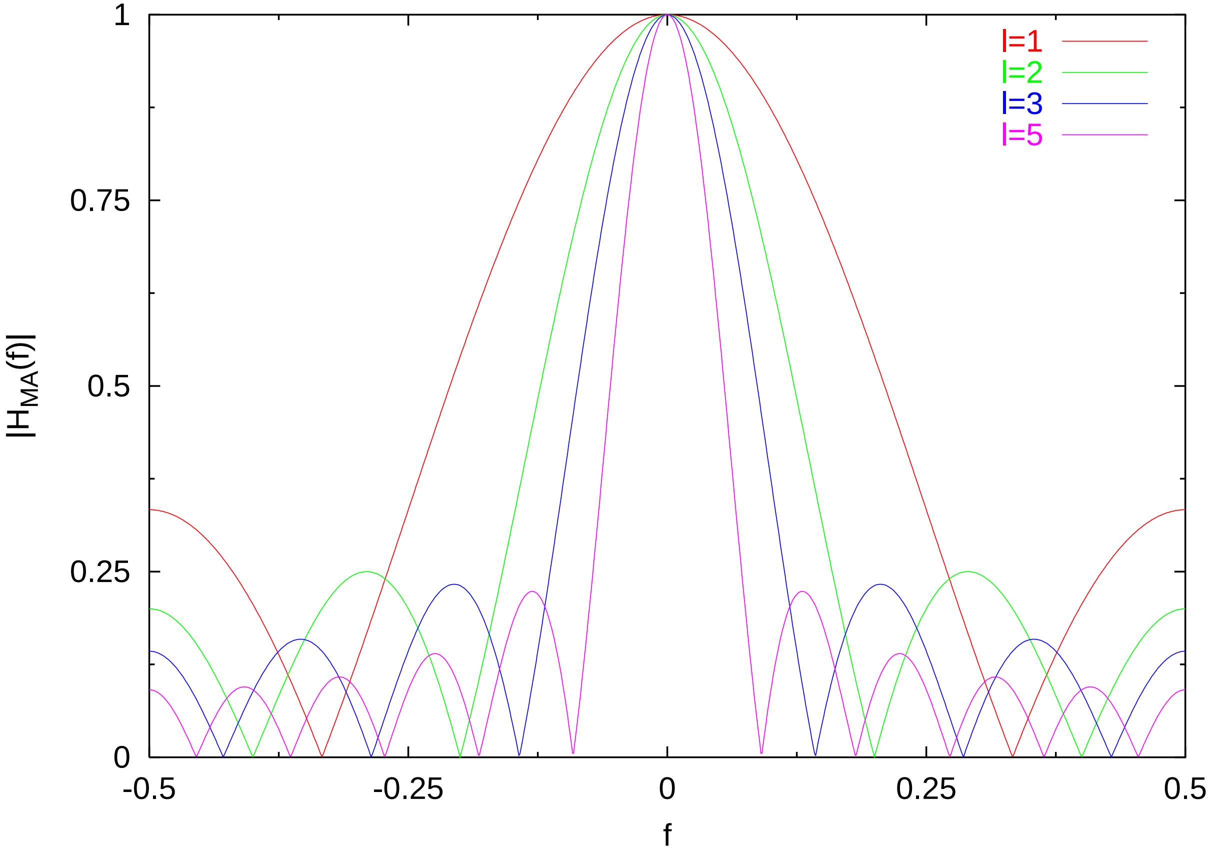
$$y_n = \frac{1}{2l+1}\Big(x_{n-l} + x_{n-l+1} + \cdots + x_n + \cdots + x_{n+l-1} + x_{n+l}\Big)\,, \qquad (14)$$

with the transfer function

$$\mathcal{H}_{\mathsf{MA}}(f) = \frac{1}{2l+1}\left(1 + 2\cos 2\pi f + 2\cos 4\pi f + \cdots + 2\cos 2l\pi f\right) \qquad (15)$$

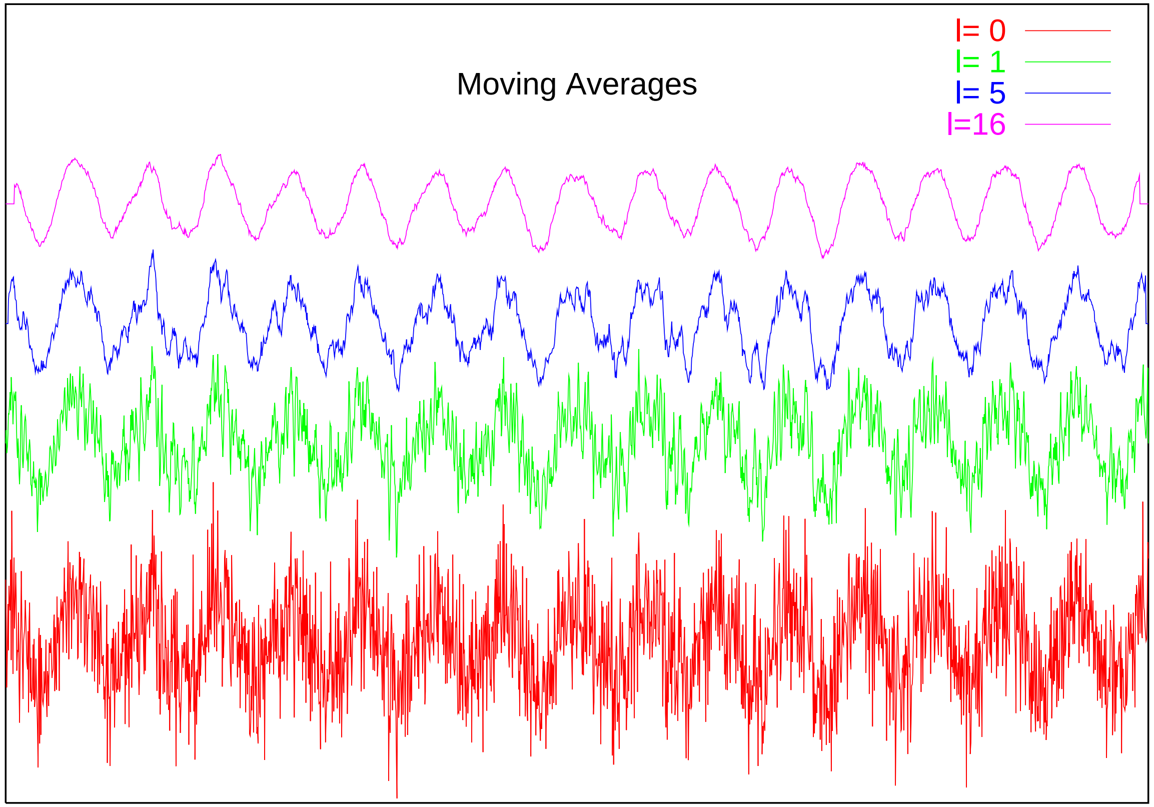
is a poor low-pass filter.

Transfer functions of unweighted moving averages



An example of the usual moving average

(the noise of the order of the signal)



Differentiating filters

First derivative:

$$\left.\frac{dg}{dx}\right|_x \simeq \frac{1}{2}\left(\frac{g(x)-g(x-\Delta)}{\Delta}+\frac{g(x+\Delta)-g(x)}{\Delta}\right)=\frac{1}{2\Delta}g(x+\Delta)-\frac{1}{2\Delta}g(x-\Delta) \tag{16a}$$

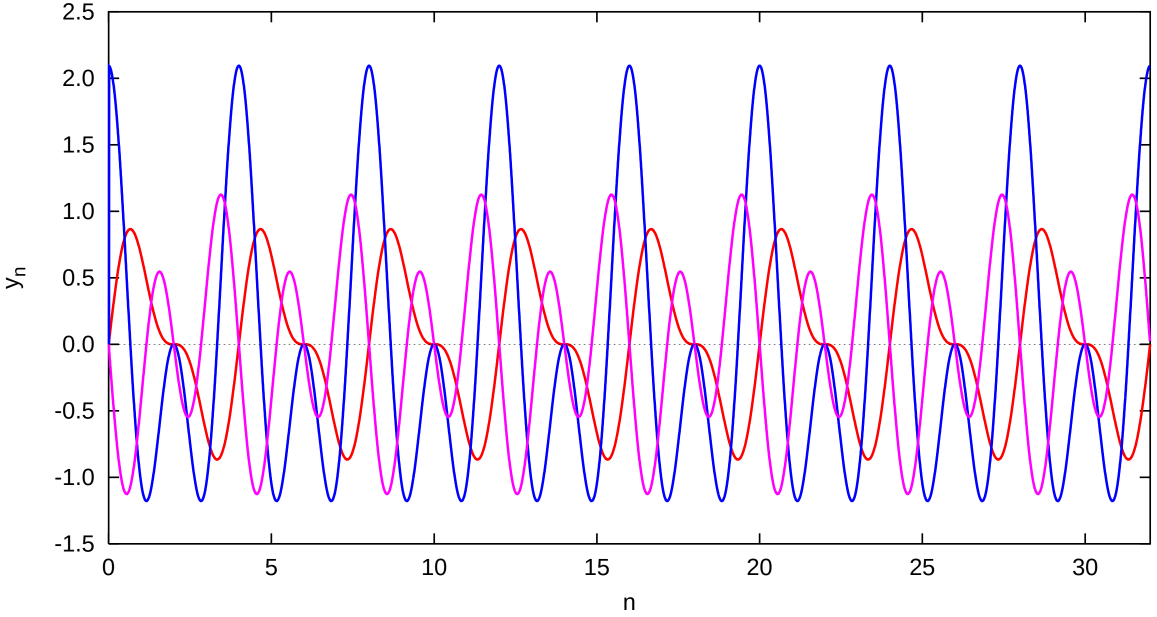
$$y_n=\frac{1}{2\Delta}x_{n+1}-\frac{1}{2\Delta}x_{n-1} \tag{16b}$$

$$\mathcal{H}(f)=\frac{i}{\Delta}\sin(2\pi f\Delta) \tag{16c}$$

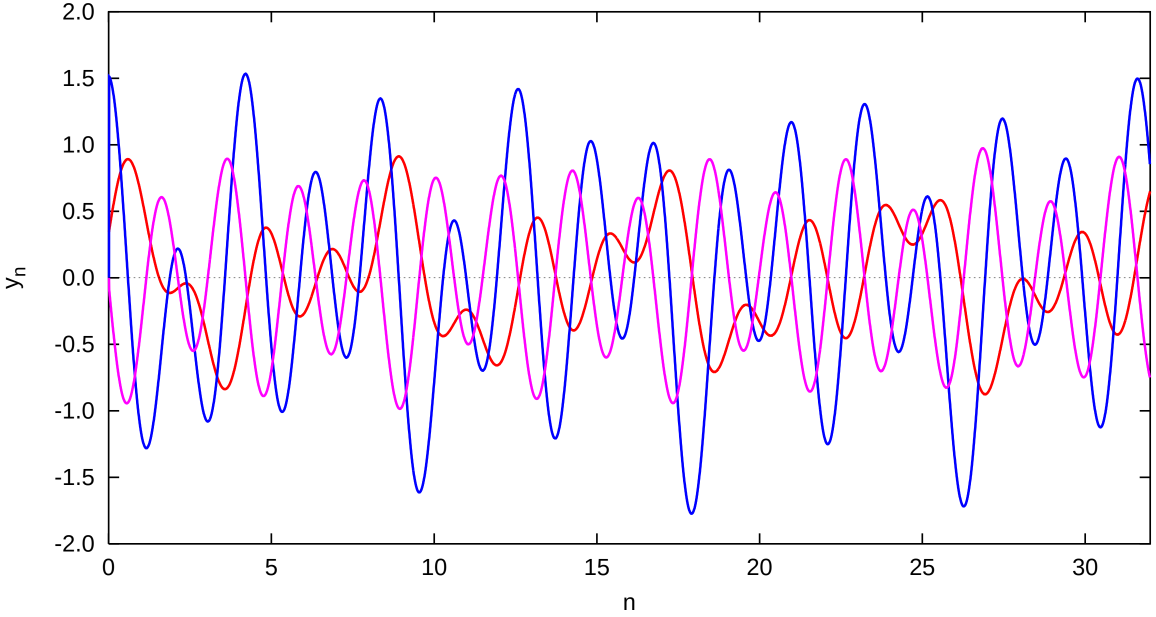
Second derivative:

$$y_n=\frac{1}{4\Delta^2}x_{n+1}-\frac{1}{2\Delta^2}x_n+\frac{1}{4\Delta^2}x_{n-1} \tag{17a}$$

$$\mathcal{H}(f)=-\frac{1}{\Delta^2}\sin(\pi f\Delta) \tag{17b}$$



2.5
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
y_n
0
5
10
15
20
25
30
n


2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
-2.0
y_n
0
5
10
15
20
25
30
n

Phase of the transfer function

In Fourier domain,

$$Y(f) = \mathcal{H}(f)X(f)\,. \tag{18}$$

$\mathcal{H}(f) = R(f)e^{i\phi(f)}$, $R(f) \geqslant 0$. The modulus of the transfer function amplifies/reduces contributions from the corresponding frequencies. What does the phase, $\phi(f)$, do?

The filter (10) is non-causal, but it is easy to find its causal version: we need to introduce *a time delay*:

$$y_n = \frac{1}{4}x_{n-2} + \frac{1}{2}x_{n-1} + \frac{1}{4}x_n \tag{19}$$

with the transfer function

$$\mathcal{H}(f) = e^{2\pi i f}\cos^2\pi f\,. \tag{20}$$

This transfer function differs from that of (10) only by a phase factor. The phase factor in (20) is responsible for the time delay!

Suppose that the phase of the transfer function *depends linearly on frequencies*, $\phi(f) = af\Delta$. Calculate the inverse transform:

$$
\begin{aligned}
y_k(t) \sim \sum_n \mathcal{H}(f_n) X(f_n) e^{-2\pi i f_n k\Delta} &= \sum_n R(f_n) e^{i a f_n \Delta} X(f_n) e^{-2\pi i f_n k\Delta} \\
&= \sum_n R(f_n) X(f_n) e^{-2\pi i f_n (k-a)\Delta}\,, \qquad (21)
\end{aligned}
$$

which corresponds to a time shift of a units ("channels"). Therefore, *filters of a linear phase introduce a uniform time shift* Filters that do not have a linear phase introduce *phase differences* between various Fourier components.

FIR filters design

If we have a FIR filter

$$y_n = \sum_{k=0}^{q} \alpha_k x_{n-k}\,, \tag{22}$$

we know that its transfer function has the form

$$\mathcal{H}(f_m) = \sum_{k=0}^{q} \alpha_k \left(e^{2\pi i f_m \Delta}\right)^k\,. \tag{23}$$

Note that (23) equals, up to a constant, to the DFT of the filter.

An inverse problem

In practice, we need to deal with an inverse problem*: Given an “ideal” transfer function $\mathcal{H}(f)$, find the order, q, and coefficients α_k of the filter (22) such that its transfer function is *as close as possible* to the “ideal” one.

*This is a technical term!

An intuitive approach

- Take the ideal transfer function $\mathcal{H}(f)$.
- Calculate the inverse transform[†]

$$h(t) = \int\limits_{-1/2}^{1/2} \mathcal{H}(f) e^{-2\pi i f t}\, df \,. \tag{24}$$

- Discretize $h(t)$ in as many points, as the desired order of the filter is.

This approach usually does not work. Usually we Fourier transform and discretize the ideal transform, at the price of distorting the transfer function.

[†] Remember that if $\Delta = 1$, $f_{\text{Nyq}} = 1/2$.

Example: A low-pass filter

The ideal transfer function reads

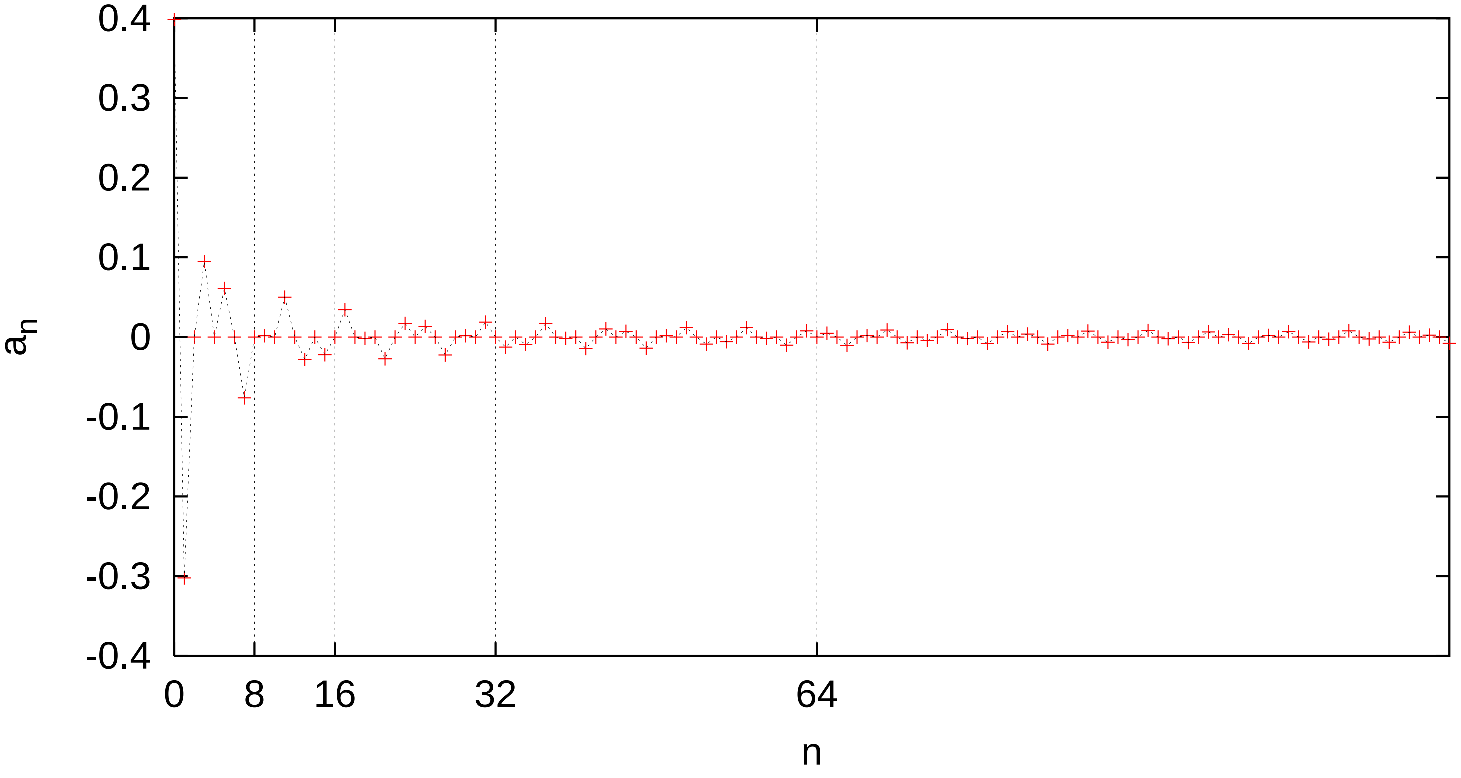
$$\mathcal{H}(f) = \begin{cases} 1 & |f| \leqslant f_0 < 1/2\,, \\ 0 & |f| > f_0\,. \end{cases} \tag{25}$$

The transfer function in the time domain equals

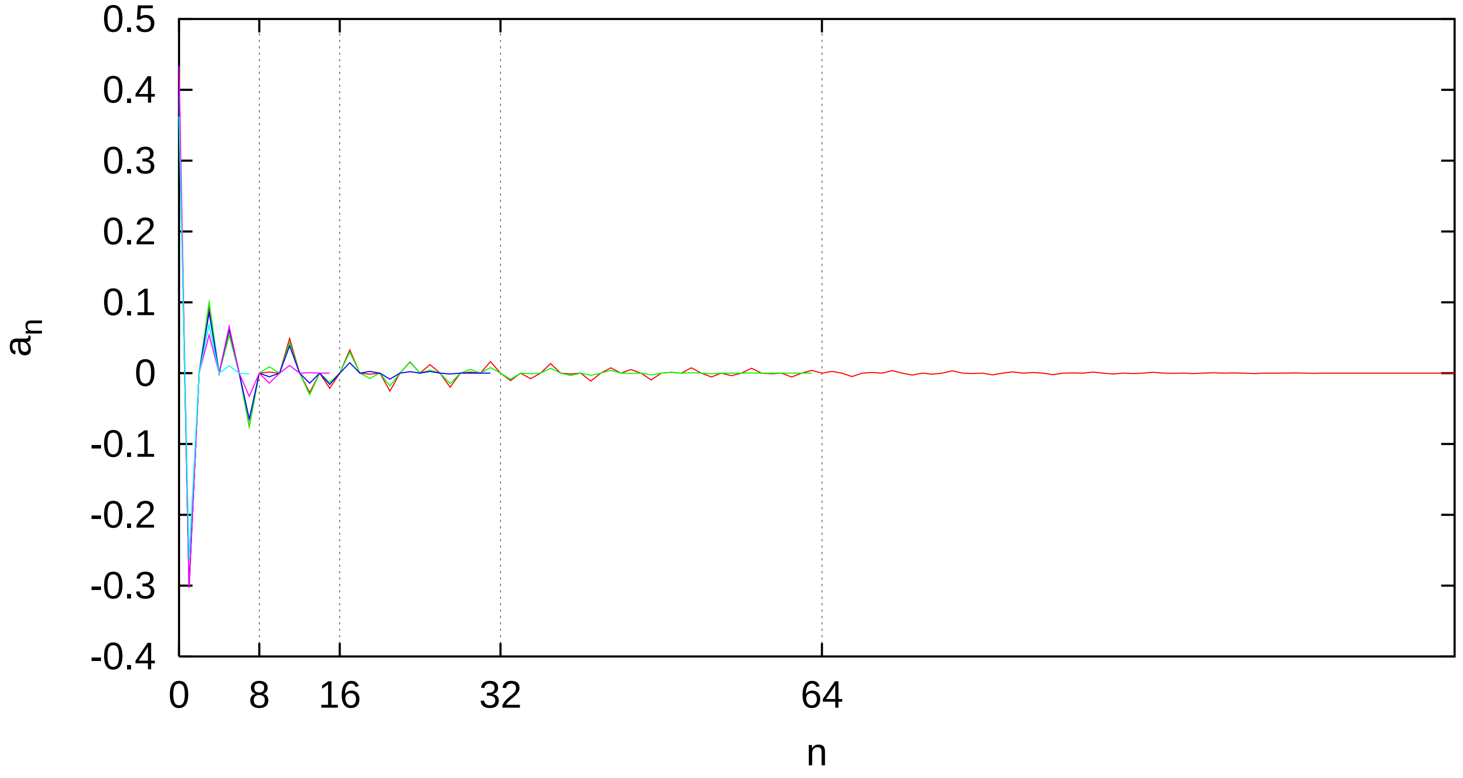
$$h(t) = \int\limits_{-1/2}^{1/2} \mathcal{H}(f) e^{-2\pi i f t}\, df = \int\limits_{-f_0}^{f_0} e^{-2\pi i f t}\, df = \frac{\sin 2\pi f_0 t}{\pi t}\,. \tag{26}$$

The amplitude of $h(t)$ falls off very slowly, there is no natural cut-off. A sharp edge contains *all* Fourier components. We either introduce an arbitrary cut-off, or do something else, mostly multiplying the ideal transfer function by a windowing function.

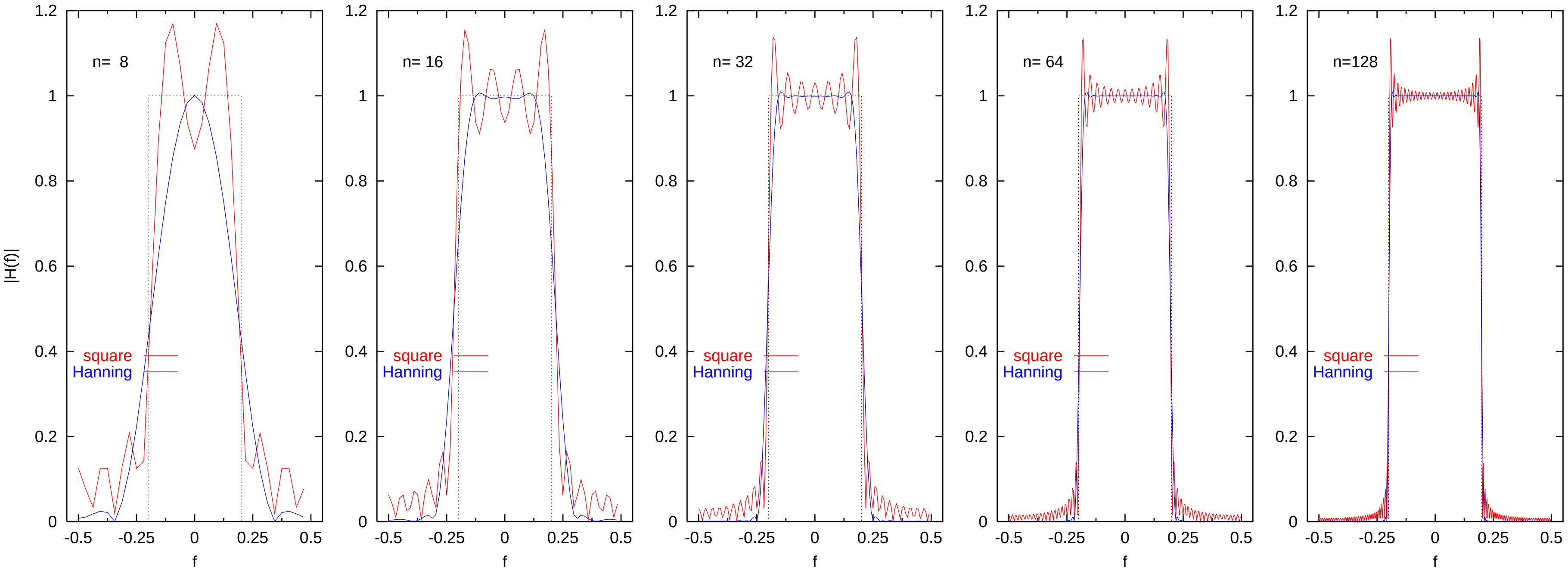
Coefficients of a low-pass FIR filter, square window



Coefficients of a low-pass FIR filter, Hannig window



Low-pass FIR filters discretized on a different number of points



Remarks

- Decent FIR filters require large orders.
- Windowing functions reduce *the ripple* (rapid oscillations near the edge of the band), but extend *the roll-off*.
- Filter design is an art. Design involves decisions on
  - *the ripple*,
  - *the roll-off*,
  - (non)linearity of the phase,
  - requirements on memory and computational complexity.

  Usually you can't optimize for all of the above simultaneously ☹.
- There is *vast* literature on filter design.

Linear IIR filters

. . . take the form

$$y_n = \sum_{k=-s}^{q} \alpha_k x_{n-k} + \sum_{k=1}^{p} \beta_k y_{n-k}\,, \tag{27}$$

where $p \geqslant 1$, x_n is a (discretized) input signal, y_n is the output signal. For convenience we discuss causal filters ($s = 0$) only; in fact, allowing for non-causality does not change much.

A problem

An *IIR* filter has a feedback loop. As a matter of principle, an IIR filter can produce a non-zero output infinitely long after the input has ceased. Can we avoid that?

How can we be sure that a stationary *input* signal produces a stationary *output*?

Linear difference equations

A homogeneous linear difference equation:

$$z_n = \beta_1 z_{n-1} + \beta_2 z_{n-2} + \cdots + \beta_p z_{n-p} \tag{28}$$

An inhomogeneous linear difference equation:

$$z_n = \beta_1 z_{n-1} + \beta_2 z_{n-2} + \cdots + \beta_p z_{n-p} + \varphi \tag{29}$$

Theorem: The general solution to an inhomogeneous difference equation equals a sum of the general solution to the corresponding homogeneous linear difference equation and any particular solution to the inhomogeneous equation.

Stability of IIR filters is determined by the homogeneous equations.

Embedding in higher dimensions

Let $\mathbf{z}_n = [z_n, z_{n-1}, \dots, z_{n-p+1}]^T \in \mathbb{R}^p$. Then the homogeneous equation (28) can be written as

$$\mathbf{z}_n = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 & \cdots & \beta_{p-1} & \beta_p \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \mathbf{z}_{n-1} \,. \tag{30}$$

Solutions to (30) are stable if *moduli of all eigenvalues are smaller that 1*.

The characteristic determinant:

$$W_p = \det \begin{bmatrix} \beta_1 - \lambda & \beta_2 & \beta_3 & \cdots & \beta_{p-1} & \beta_p \\ 1 & -\lambda & 0 & \cdots & 0 & 0 \\ 0 & 1 & -\lambda & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -\lambda \end{bmatrix}$$

$$= -\lambda W_{p-1} + (-1)^{p+1}\beta_p = \lambda^2 W_{p-2} + (-1)^{p+1}\beta_{p-1}\lambda + (-1)^{p+1}\beta_p = \dots$$
$$= (-1)^{p+1}\left(-\lambda^p + \beta_1\lambda^{p-1} + \beta_2\lambda^{p-2} + \cdots + \beta_p\right) \qquad (31)$$

Stability condition of an IIR filter

An *IIR* filter (27) is stable if and only if roots of the equation

$$\lambda^p - \beta_1\lambda^{p-1} - \beta_2\lambda^{p-2} - \cdots - \beta_p = 0 \tag{32}$$

lie inside the unit circle.

A confusion in terminology

Sometimes the above condition is formulated for the reciprocals of λ's:

An *IIR* filter (27) is stable if and only if roots of the equation

$$1 - \beta_1 u - \beta_2 u^2 - \cdots - \beta_p u^p = 0 \tag{33}$$

lie *outside* the unit circle.

The conditions (32), (33) are equivalent, but they should not be confused.

Example

Consider a filter

$$y_n = \beta_1 y_{n-1} + x_n \,. \tag{34}$$

Its characteristic polynomial in the form (33) reads

$$\beta(z) = 1 - \beta_1 z \,. \tag{35}$$

It can be seen that if $|\beta_1| > 1$, y_n “explodes”, and therefore we need to have $|\beta_1| < 1$, which means that the only root of (35), $1/\beta_1$, lies outside the unit interval (and of course, outside the unit circle).

IIR filter transfer function

Write (27) in the form

$$y_n - \beta_1 y_{n-1} - \cdots - \beta_p y_{n-p} = \alpha_0 x_n + \alpha_1 x_{n-1} + \cdots + \alpha_q x_{n-q}\,, \tag{36}$$

and Fourier transfer it. After some algebra,

$$\mathcal{H}(f_m) = \frac{\sum\limits_{k=0}^{q} \alpha_k \left(e^{2\pi i f_m \Delta}\right)^k}{1 - \sum\limits_{j=1}^{p} \beta_j \left(e^{2\pi i f_m \Delta}\right)^j}\,. \tag{37}$$

Note: The form of the denominator in (37) suggests that the stability condition (33) is “natural”.

IIR filters design

The trouble with IIR filters design is that one needs to avoid poles that lead to instability. Usually, it is not *easy* to verify whether a pole lies inside or outside the unit circle. The following *bilinear transform* is frequently used:

$$z = \frac{1 - iw}{1 + iw} \quad \text{or} \quad w = i\frac{z-1}{z+1} \,. \tag{38}$$

Calculate

$$|z|^2 = \frac{1-iw}{1+iw} \cdot \frac{1+i\bar{w}}{1-i\bar{w}} = \frac{1+i\bar{w}-iw+|w|^2}{1-i\bar{w}+iw+|w|^2} = \frac{1+|w|^2+2\,\mathrm{Im}\,w}{1+|w|^2-2\,\mathrm{Im}\,w} \,. \tag{39}$$

The filter is stable if $|z|^2 > 1$, or $\mathrm{Im}\,w > 0$: we allow only poles that lie in the upper half-plane.

IIR filter design procedure

- An "ideal" transfer function $\mathcal{H}(f)$ is given.

- Find *a rational function* that approximates $\mathcal{H}(f)$ sufficiently well. Let $H(f)$ be this function. It must be real and nonnegative.

- Find poles of $H(f)$. Half of them lie in the upper half-plane, half in the lower half-plane. Take a product of terms with poles from the upper half-plane only, substitute $f = i(z-1)/(z+1)$, simplify and identify the coefficients.

Example: Butterworth filter of order N

As before, we are trying to design a low-pass filter. The step function is approximated by

$$H(f) = \frac{1}{1 + \left(\frac{f}{f_0}\right)^{2N}}\,, \tag{40}$$

where f_0 is the cut-off frequency. A filter based on (40) is called *Butterworth filter of order N*. Its poles lie on a circle with a radius f_0, symmetrically with respect to the real axis.

$N = 1$ is the simplest case.

$$H(f) = \frac{1}{1+(\frac{f}{f_0})^2} = \underbrace{\frac{i}{\frac{f}{f_0}+i}}_{\text{lower}} \cdot \underbrace{\frac{-i}{\frac{f}{f_0}-i}}_{\text{upper}} \,. \tag{41}$$

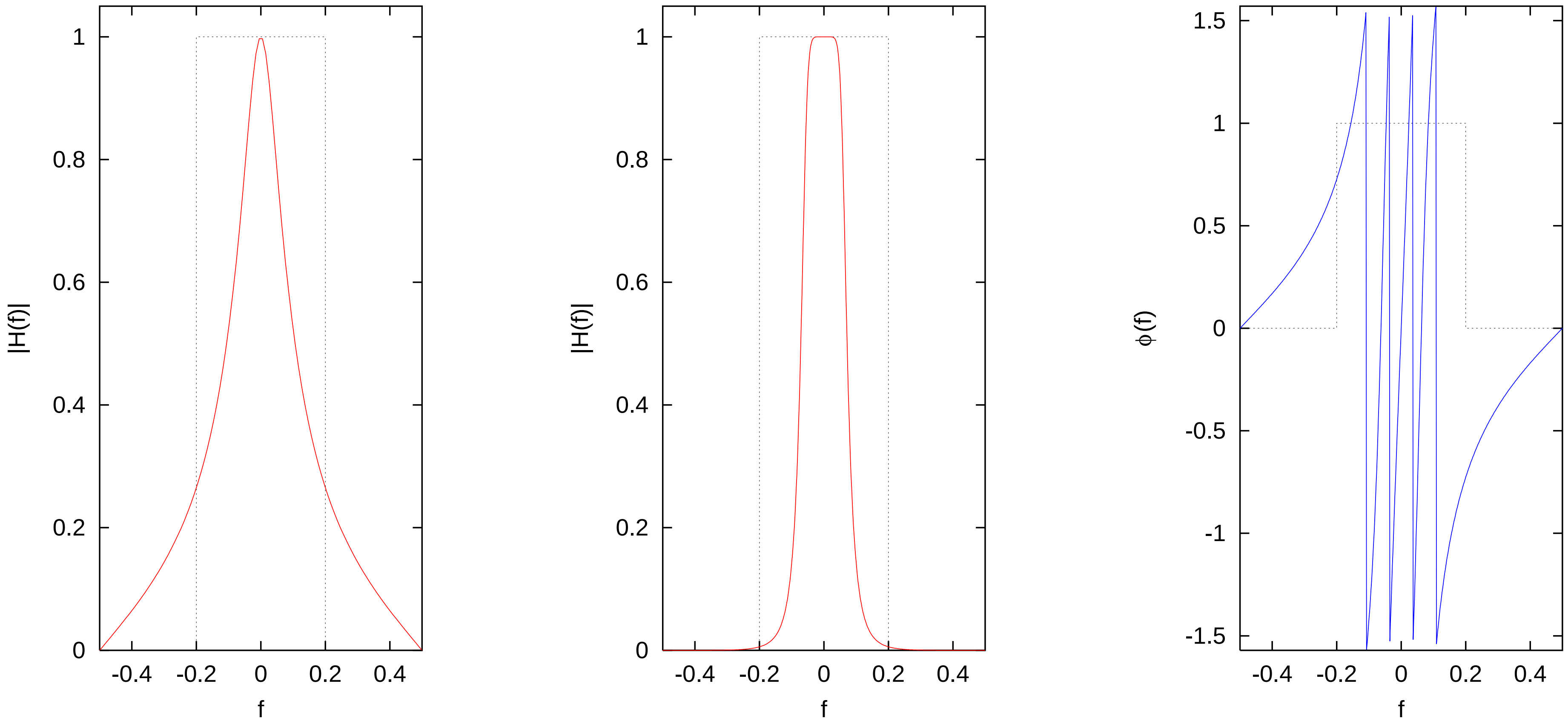
We thus take

$$\frac{-if_0}{f-if_0} = \frac{-if_0}{i\frac{z-1}{z+1}-if_0} = \frac{-f_0-f_0z}{z-1-f_0z-f_0} = \frac{-f_0-f_0z}{-(1+f_0)+(1-f_0)z}$$

$$= \frac{\frac{f_0}{1+f_0}+\frac{f_0}{1+f_0}z}{1-\frac{1-f_0}{1+f_0}z} \tag{42}$$

$\alpha_0 = \alpha_1 = f_0/(1+f_0)$, $\beta_1 = (1-f_0)/(1+f_0)$ are the coefficients of the filter.

Left: Transfer function of Butterworth filter of order 1. Middle: Transfer function of Butterworth filter of order 4. Right: Phase of Butterworth filter of order 4.



Most popular filters

1. Butterworth filters

   - See above (40).
   - The first $2N-1$ derivatives of $\mathcal{H}(f)$ vanish at $f=0$ — the filter is *maximally flat*.
   - Poles lie on a circle.
   - Used in audio processing.

2. Chebyshev filters

- A steeper *rolloff*, but *ripples* appear.
- There are two kinds of Chebyshev filters, with *ripples* in the*passband*:

$$\mathcal{H}(f) = \frac{1}{1 + \left[T_N\left(f/f_p\right)\right]^2}\,, \tag{43}$$

Ripple and with *ripples* in the *stopband*:

$$\mathcal{H}(f) = \frac{1}{1 + \left[\frac{T_N(f_s/f_p)}{T_N(f_s/f)}\right]^2}\,. \tag{44}$$

T_N in the above stands for a Chebyshev polynomial of order N, f_p is the upper bound of the passband, $f_s > f_p$ is the lower bound of the stopband.

- Because of ripples, they are not used in audio processing, but it is most excellent if the passband contains a single “interesting” frequency (for example, if higher harmonics are to be eliminated).

3. Elliptic filters
$$\mathcal{H}(f) = \frac{1}{1 + \left[R_N\left(f/f_p\right)\right]^2}\,, \tag{45}$$
where R_N is a rational function, with roots of the numerator within $[-1/2, 1/2]$, and roots of the denominator outside, with $R_N(1/z) = 1/R_N(z)$. Poles of such a filter lie on an ellipsis.

4. Bessel filters, with a constant delay in the passband.

All above filters can have analog realizations.

Design of analog and digital filters is very important in electronics, telecommunication etc.

Because rational approximation is much better than polynomial approximation, IIR filters require significantly lower orders than FIR filters of similar performance.