Discrete-time filtering

Dr. Ing. Rodrigo Gonzalez

rodralez@frm.utn.edu.ar

Técnicas Digitales III

Universidad Tecnológica Nacional,
Facultad Regional Mendoza.

Resumen

Filtering in different domains

- Filtering in time domain (signal restoration, smoothing, denoising).
  - Describes when something occurs and what the amplitude of the occurrence is.
  - Each sample contains information that is interpretable without reference to any other sample.
  - The step response describes how information represented in the time domain is being modified by the system.
  - Example: EKG signal, accelerometer, gyroscope.
- Filtering in frequency domain (signal separation).
  - The information is contained in the relationship between many points in the signal.
  - The frequency response shows how information represented in the frequency domain is being changed.
  - Example: digital filter for a hearing aid, equalizer.

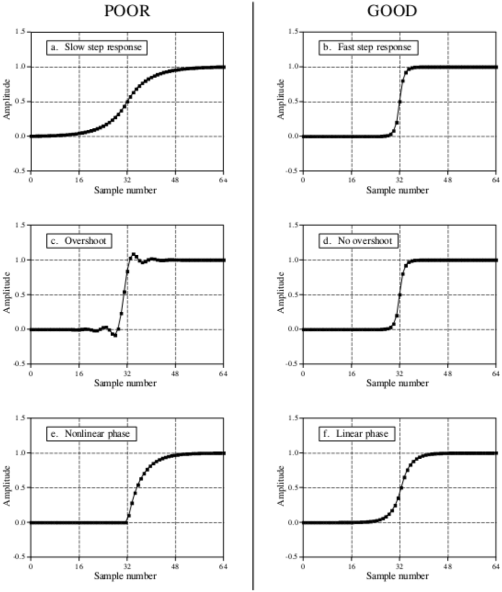
It is not possible to optimize a filter for both domains. Good performance in the time domain results in poor performance in the frequency domain, and vice versa.

Classification of discrete filters

Table : Classification of discrete filters

	Finite impulse response (FIR)	Infinite impulse response (IIR)
Filtering in time domain	Moving average	Leaky Integrator
Filtering in frequency domain	Windowed Filters Equiripple Minimax	Bilinear z-transform

- Risetime (between 10%$\sim$90% amplitude).
- Overshoot.
- Linear phase.



- The moving average filter is a convolution of the input signal with a rectangular pulse having an area of one.
- *Local average*.
- There is a delay of N/2 samples between input and output.

$$y[n] = x[n] * h[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k]\,, \tag{1}$$

$$h[n] = \frac{1}{N} \sum_{k=0}^{N-1} \delta[n-k]\,, \tag{2}$$

$$SNR = 10 log_{10}(\sqrt{N})\,. \tag{3}$$

- It can be seen that the moving average filter is a FIR filter. Why?

Noise Reduction vs. Step Response

- It reduce random white noise while trying to keep the sharpest step response.

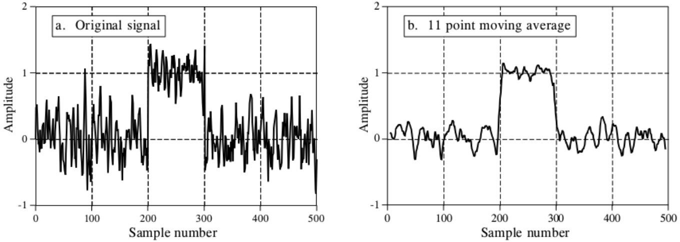
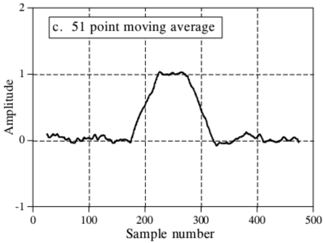
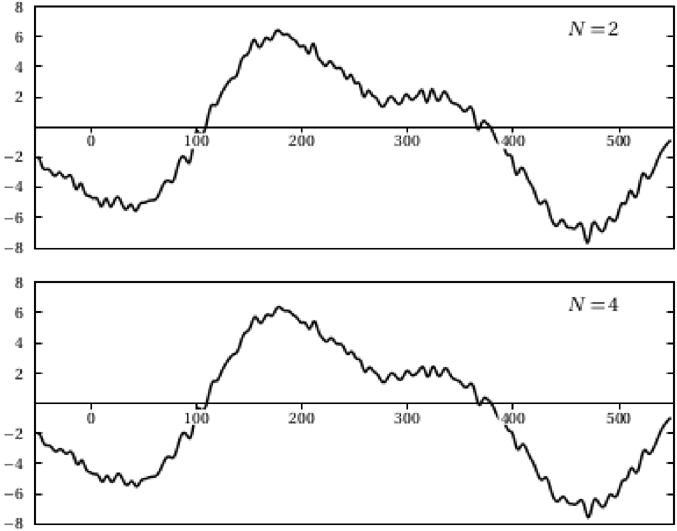


FIGURE 15-1
Example of a moving average filter. In (a), a rectangular pulse is buried in random noise. In (b) and (c), this signal is filtered with 11 and 51 point moving average filters, respectively. As the number of points in the filter increases, the noise becomes lower; however, the edges becoming less sharp. The moving average filter is the *optimal* solution for this problem, providing the lowest noise possible for a given edge sharpness.

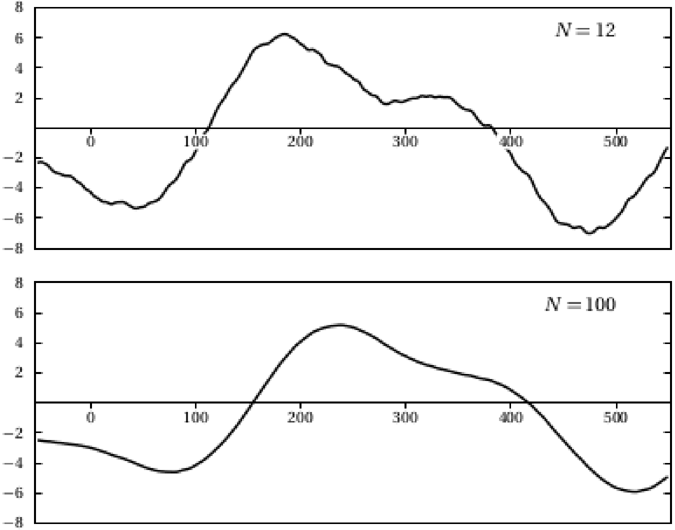


Noise Reduction

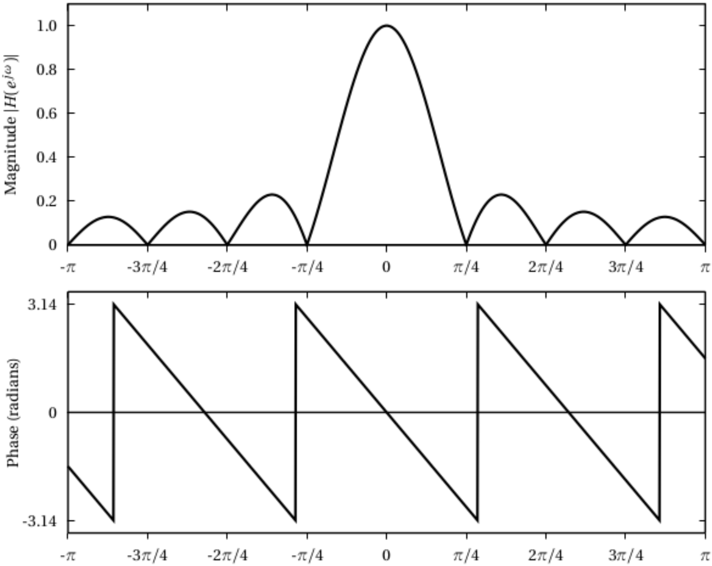


Noise Reduction

- Note how the signal is delayed as *N* grows.



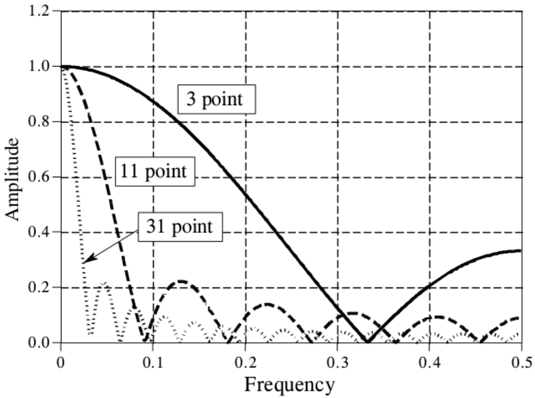
Frequency Response



Frequency Response

- The moving average filter cannot separate one band of frequencies from another.
- In short, the moving average is a good *smoothing filter* (the action in the time domain), but a bad low-pass filter (the action in the frequency domain).

$$H[e^{j\omega}] = \frac{1}{N}\frac{\sin(\omega N/2)}{\sin(\omega/2)} e^{-j\frac{N-1}{2}\omega} . \quad (4)$$



Leaky integrator filter

- The moving average filter is a convolution of the input signal with a rectangular pulse having an area of one.
- *Local average*.
- There is a delay of N/2 samples between input and output.

$$y[n] = x[n] * h[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[n-k] \,, \tag{5}$$

$$y[n] = \frac{1-M}{M} \, y_{M-1}[n-1] + \frac{1}{M} \, x[n-k] \,. \tag{6}$$

Doing,

$$\lambda = \frac{1-M}{M} \,, \tag{7}$$

Then,

$$y[n] = \lambda \, y_{M-1}[n-1] + (1-\lambda) \, x[n-k] \,. \tag{8}$$

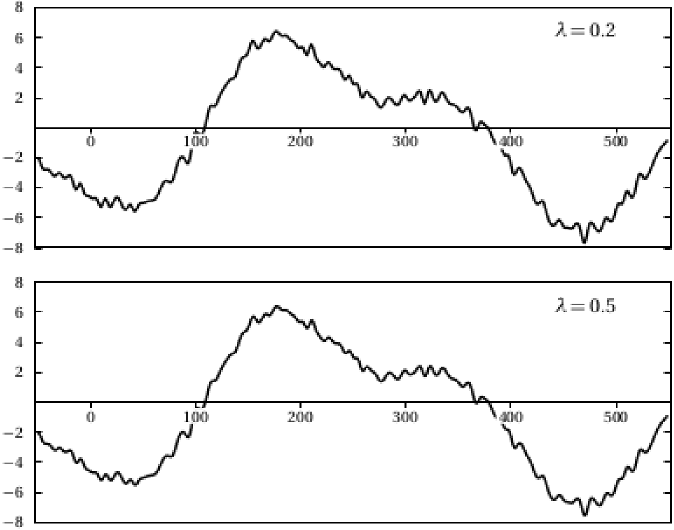
- It can be seen that the leaky integrator filter is an IIR filter. Why?

Leaky integrator filter

$$y[n] = \lambda\, y_{M-1}[n-1] + (1-\lambda)\, x[n-k]\,.$$

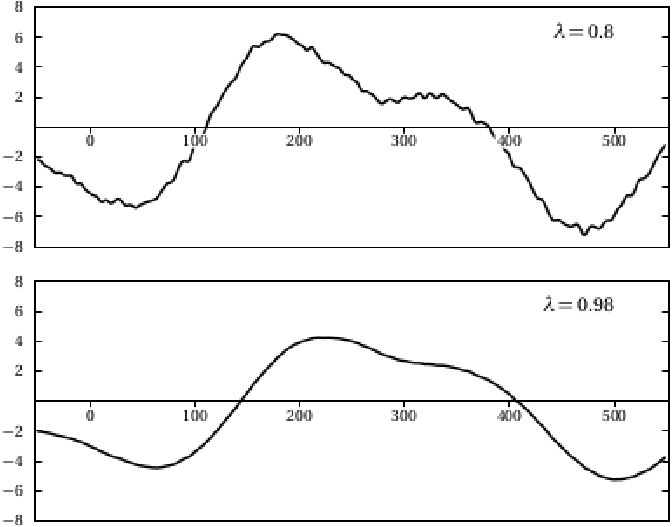
- No longer a convolution sum; instead, an instance of a *constant coefficient difference equation*.
- We need to set the initial conditions.
- The value of λ (which is the pole of the system) determines the smoothing power of the filter.

Noise Reduction

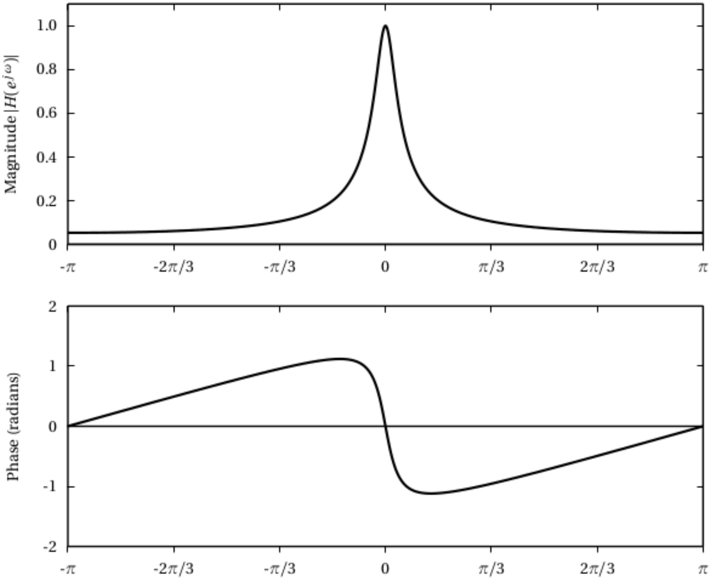


Noise Reduction

- Note how the signal is delayed as λ grows.

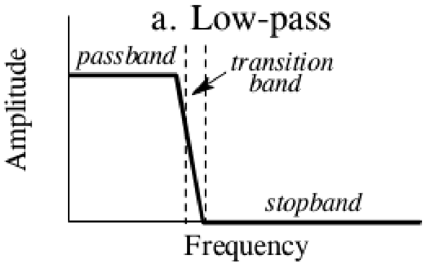


Frequency Response

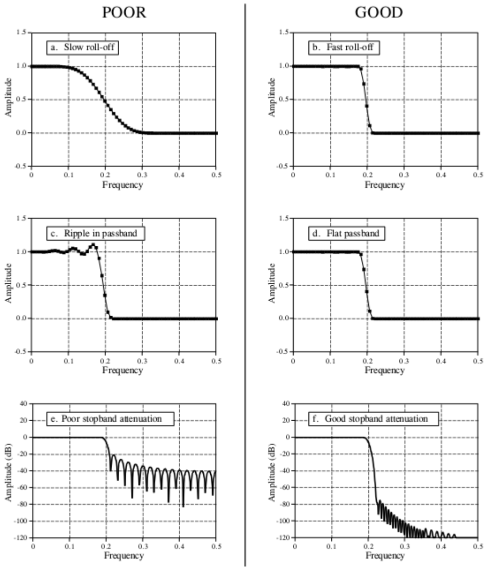


Frequency domain parameters

- Passband.
- Stopband.
- Transition band (fast roll-off).
- Passband ripple.
- Stopband ripple.



Frequency response parameters, cont'd



Strategy of filtering by windowing

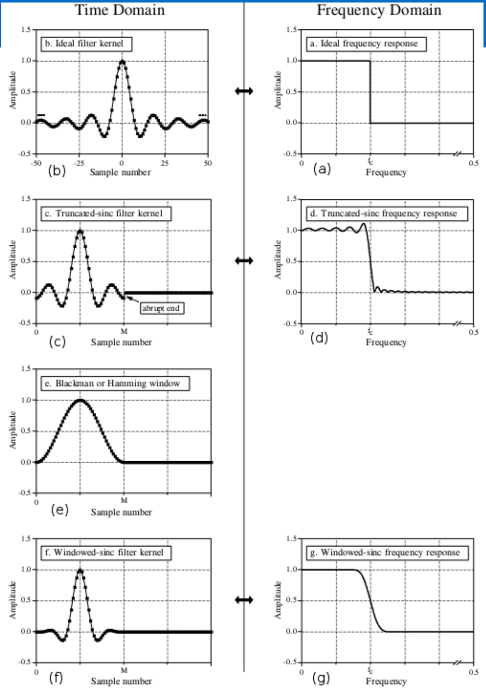
- Taking the Inverse Fourier Transform of an ideal frequency response (a) produces an ideal sinc filter kernel (b, impulse response).

$$hs[i] = \frac{\sin(2\pi f_C i)}{i\pi}$$

- Truncated-sinc (c) and frequency response (d). Discontinuity is significant no matter how long M is made (Gibbs phenomenon).

$$h[i] = hs[i] \cdot w[i]$$

- Multiplying the truncated-sinc (c) by the Blackman window (e) results in the windowed-sinc filter kernel (f) with frequency response (g).



Name of window function $w(n)$	Mathematical definition
Rectangular	1
Hanning	$0.5-0.5\cos\left[\frac{2\pi n}{N-1}\right]$
Hamming	$0.54-0.46\cos\left[\frac{2\pi n}{N-1}\right]$
Blackman	$0.42-0.5\cos\left[\frac{2\pi n}{N-1}\right]+0.08\cos\left[\frac{2\pi n}{N-1}\right]$
Kaiser	$\frac{I_0\left[\beta\sqrt{1-\left(\frac{\lvert 2n-N+1\rvert}{N-1}\right)^2}\right]}{-I_o(\beta)}$ Where, $I_0(x)=\sum_{k=0}^{\infty}\left(\frac{x^k}{2^k\,k!}\right)^2$

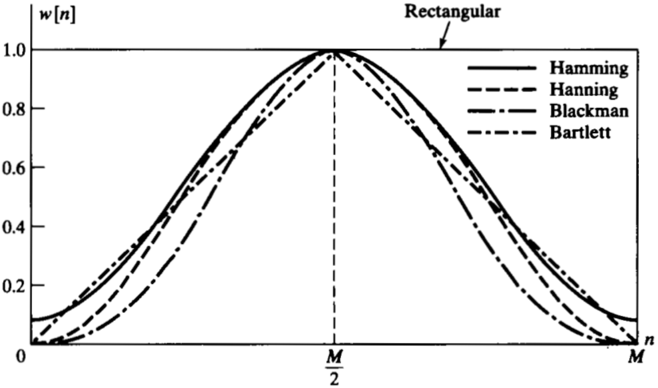
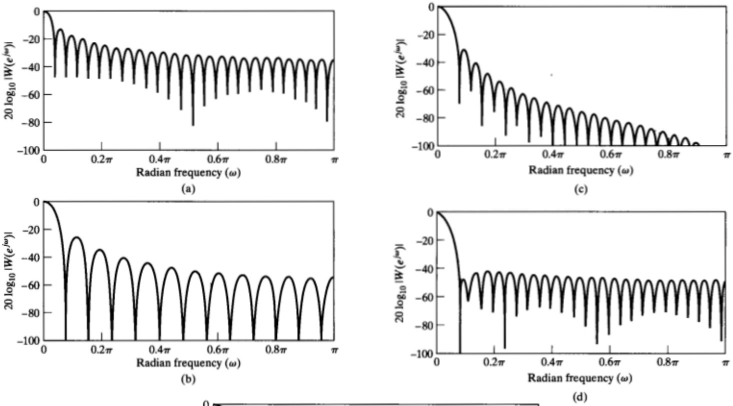


Figure 7.21 Commonly used windows.



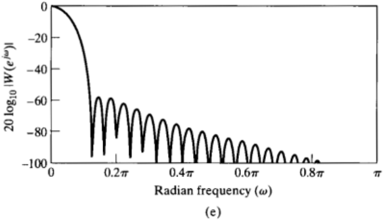


Figure 7.22 Fourier transforms (log magnitude) of windows of Figure 7.21. with $M = 50$. (a) Rectangular. (b) Bartlett. (c) Hanning. (d) Hamming. (e) Blackman.

Kaiser window filter

- The Kaiser window has two parameters:
  - Lenght, M+1.
  - Shape parameter, β.
- Trade-off between side-lobe amplitude and main-lobe width.

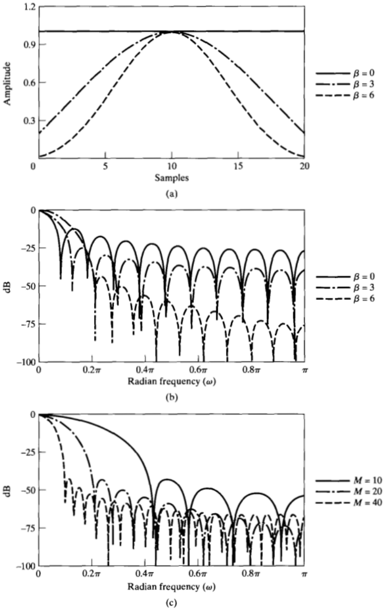
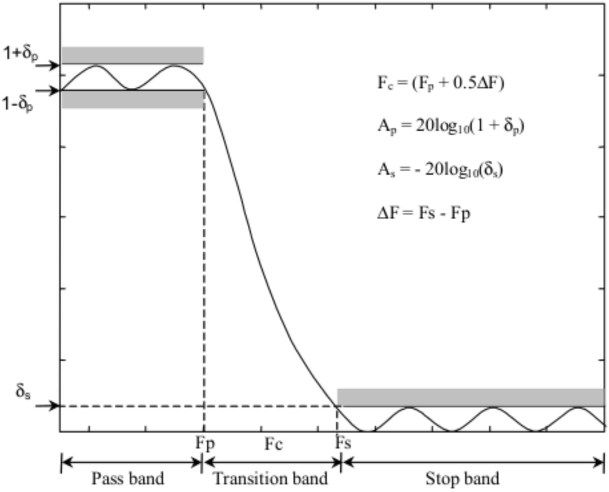


TABLE 7.1 COMPARISON OF COMMONLY USED WINDOWS

Type of Window	Peak Side-Lobe Amplitude (Relative)	Approximate Width of Main Lobe	Peak Approximation Error, $20\log_{10}\delta$ (dB)	Equivalent Kaiser Window, β	Transition Width of Equivalent Kaiser Window
Rectangular	−13	$4\pi/(M+1)$	−21	0	$1.81\pi/M$
Bartlett	−25	$8\pi/M$	−25	1.33	$2.37\pi/M$
Hanning	−31	$8\pi/M$	−44	3.86	$5.01\pi/M$
Hamming	−41	$8\pi/M$	−53	4.86	$6.27\pi/M$
Blackman	−57	$12\pi/M$	−74	7.04	$9.19\pi/M$



$1+\delta_p$
$1-\delta_p$
δ_s
$F_c = (F_p + 0.5\Delta F)$
$A_p = 20\log_{10}(1 + \delta_p)$
$A_s = -20\log_{10}(\delta_s)$
$\Delta F = Fs - Fp$
Fp
Fc
Fs
Pass band
Transition band
Stop band

FIR filter design, cont'd

Name of window function $w[n]$	Transition width ΔF in (Hz), (normalised)	Pass-band ripple A_p in (dB)	Ripple δ_p, δ_s	Side-lobe level in (dB)	Stop-band attenuation A_s in (dB)
Rectangular	0.9/N	0.741	0.089	-13	21
Hanning	3.1/N	0.0546	0.063	-31	44
Hamming	3.3/N	0.0194	0.0022	-41	53
Blackman	5.5/N	0.0017	0.000196	-57	74
Kaiser β=4.54	2.93/N	0.0274			50
β=5.65	3.63/N	0.00867			60
β=6.76	4.32/N	0.00275			70
β=8.96	5.71/N	0.000275			90

A FIR low-pass filter is required to have the following specifications:

1. Pass-band edge frequency $f_p = 2$ kHz
2. Transition band $\Delta f = 200$ Hz
3. Pass-band ripple $A_p = 0.1$dB
4. Minimum stop-band attenuation $A_s = 50$dB
5. Sampling frequency of $f_s = 10$ kHz

Example of FIR design, cont'd

Pass-band ripple, $A_p = 20\log_{10}(1 + \delta_p)$

$$\delta_p = \log_{10}^{-1}\left[\frac{0.1}{20}\right] - 1 = 0.0116$$

Minimum stop-band attenuation $A_s = -\ 20\log_{10}(\delta_s)$

$$\delta_s = \log_{10}^{-1}\left[\frac{-50}{20}\right] = 0.00316$$

The normalised pass-band edge frequency

$$F_p = f_p / f_s = \frac{2x10^3}{10x10^3} = 0.2$$

The normalised transition width

$$\Delta F = \Delta f / f_s = \frac{200}{10x10^3} = 0.02$$

$$N = \frac{3.3}{0.02} = 165$$

The bilinear transformation corresponds to replacing s by,

$$s = \frac{2}{T_d}\left(\frac{1-z^{-1}}{1+z^{-1}}\right). \tag{9}$$

Solving for z,

$$z = \frac{1+(T_d/2)s}{1-(T_d/2)s}. \tag{10}$$

Ω is the analog frequency, $-\infty, < \Omega < \infty$, and ω is the "digital" frequency, $-\pi, < \omega < \pi$, the relationship between both can be found by replacing $z = e^{j\omega}$ in Eq. 9,

$$s = \frac{2}{T_d}\left(\frac{1-e^{-j\omega}}{1+e^{-j\omega}}\right), \tag{11}$$

$$s = \sigma + j\Omega = \frac{2}{T_d}\left[\frac{2e^{-j\omega/2}(j\sin\omega/2)}{2e^{-j\omega/2}(\cos\omega/2)}\right] = j\frac{2}{T_d}\tan(\omega/2). \tag{12}$$

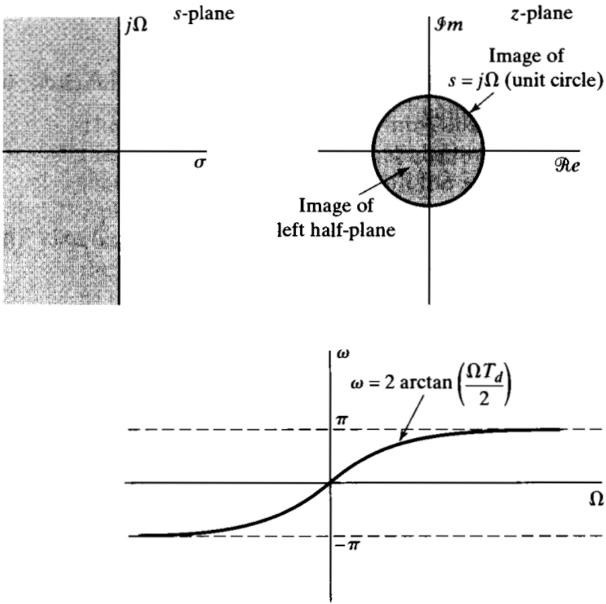
Bilinear transform, cont'd

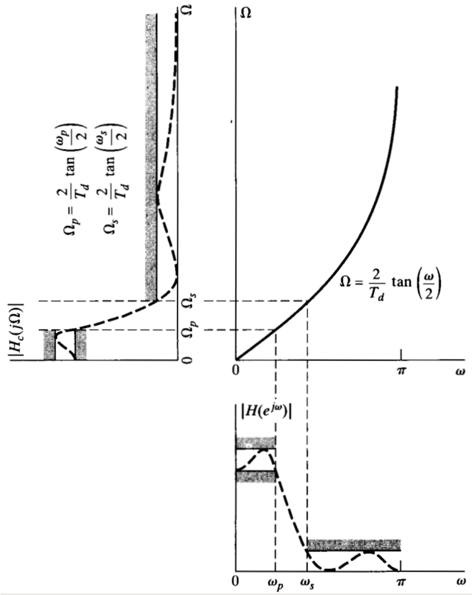
Equating real and imaginary parts on both sides of Eq. 12

$$\Omega = \frac{2}{T_d} \tan(\omega/2)\,. \tag{13}$$

Or

$$\omega = \arctan(\Omega T_d/2)\,. \tag{14}$$


$j\Omega$
s-plane
σ
$\mathcal{I}m$
z-plane
Image of
$s = j\Omega$ (unit circle)
$\mathcal{R}e$
Image of
left half-plane
ω
$\omega = 2 \arctan\left(\frac{\Omega T_d}{2}\right)$
π
$-\pi$
Ω


$|H_c(j\Omega)|$
$\Omega_p = \frac{2}{T_d}\tan\left(\frac{\omega_p}{2}\right)$
$\Omega_s = \frac{2}{T_d}\tan\left(\frac{\omega_s}{2}\right)$
Ω
Ω_s
Ω_p
0
$\Omega = \frac{2}{T_d}\tan\left(\frac{\omega}{2}\right)$
π
ω
$|H(e^{j\omega})|$
ω_p
ω_s

The nonlinear warping of the frequency axis introduced by the bilinear transformation will not preserve linearity in phase response.

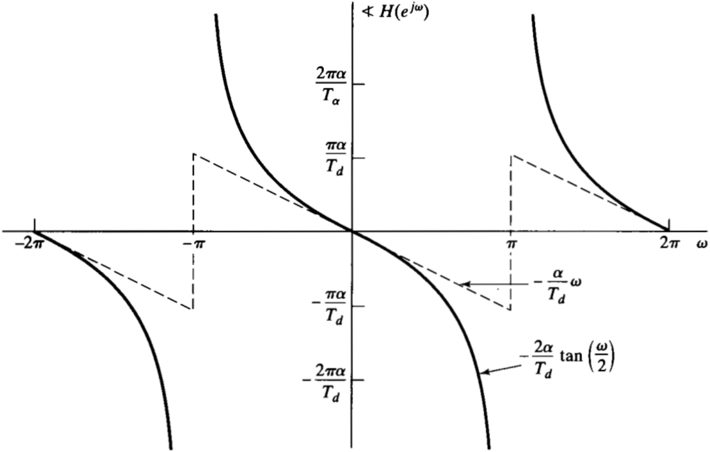


Figure 7.9 Illustration of the effect of the bilinear transformation on a linear phase characteristic. (Dashed line is linear phase and solid line is phase resulting from bilinear transformation.)

Design a digital filter equivalent of a 2nd order Butterworth low-pass filter with a cut-off frequency f_c = 100 Hz and a sampling frequency f_s = 1000 samples/sec. Derive the finite difference equation and draw the realisation structure of the filter. Given that the analogue prototype of the frequency-domain transfer function $H(s)$ for a Butterworth filter is:

$$H(s) = \frac{1}{s^2 + \sqrt{2} \cdot s + 1}$$

The normalised cut-off frequency of the digital filter is given by the following equation:

$$\Omega_c = \frac{2\pi f_c}{f_s} = \frac{2\pi 100}{1000} = 0.628$$

Now determine the equivalent analogue filter cut-off frequency ω_{ac}, using the pre-warping function of Equation 5.9. The value of K is immaterial so let $K = 1$.

$$\omega_{ac} = K \cdot \tan\left(\frac{\Omega_c}{2}\right) = 1 \cdot \tan\left(\frac{0.628}{2}\right)$$

$$\omega_{ac} = 0.325 \quad rads/\sec$$

Example of IIR design, cont'd

Now denormalise the frequency-domain transfer function $H(s)$ of the Butterworth filter, with the corresponding low-pass to low-pass frequency transformation of Equation 5.10. Hence the transfer function of the Butterworth filter becomes:

$$H(s) = \frac{1}{\left[\frac{s}{0.325}\right]^2 + \sqrt{2}\cdot\left[\frac{s}{0.325}\right] + 1}$$

Next, convert the analogue filter into an equivalent digital filter by applying the bilinear z-transform. This is achieved by making a substitution for s in the transfer function.

$$s = \frac{z-1}{z+1} \equiv \frac{1-z^{-1}}{1+z^{-1}}$$

$$H(z) = \frac{1}{\frac{1}{0.325^2}\cdot\left[\frac{1-z^{-1}}{1+z^{-1}}\right]^2 + \frac{\sqrt{2}}{0.325}\cdot\left[\frac{1-z^{-1}}{1+z^{-1}}\right] + 1}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{0.067 + 0.135z^{-1} + 0.067z^{-2}}{1 - 1.1429z^{-1} + 0.4127z^{-2}}$$

The finite difference equation of the filter is found by inverting the transfer function.

$$y(n) = 1.1429y(n-1) - 0.4127y(n-2) + 0.067x(n) + 0.135x(n-1) + 0.067x(n-2)$$

FIR vs IIR

FIR, pros:

- Unconditional stability (no poles).
- Precise control of the phase response and, in particular, exact linear phase.
- Optimal algorithmic design procedures.
- Robustness with respect to finite numerical precision hardware.

FIR, cons:

- Longer input-output delay.
- Higher computational cost with respect to IIR solutions.

IIR, pros:

- Lower computational cost with respect to an FIR with similar behavior.
- Shorter input-output delay.
- Compact representation.

IIR, cons:

- Stability is not guaranteed.
- Phase response is difficult to control.
- Design is complex in the general case.
- Sensitive to numerical precision.

Direct form FIR implementation

$$H(z) = b_0 + b_1 z^{-1} + ... + b_{M-1} z^{M-1}$$

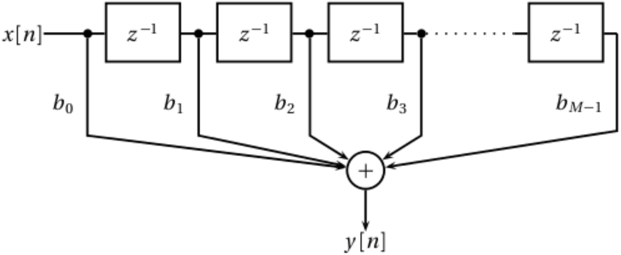


Figure 7.22 Direct FIR implementation.

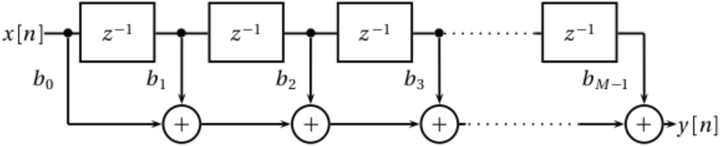


Figure 7.23 Transversal FIR implementation.

Direct form I IIR implementation

$$H(z) = \frac{b_0 + b_1 z^{-1} + ... + b_{N-1} z^{N-1}}{1 + a_1 z^{-1} + ... + a_{M-1} z^{M-1}}$$

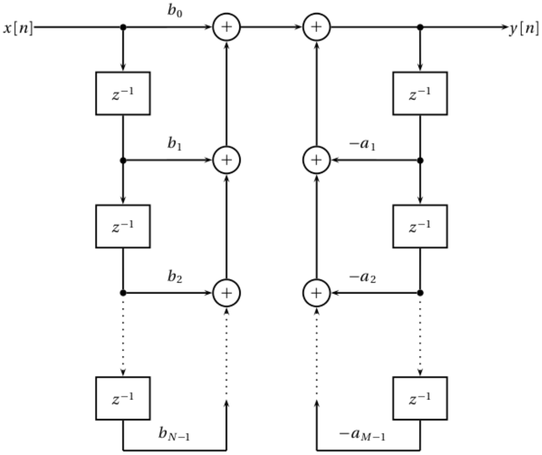


Figure 7.24 Direct Form implementation of an IIR filter.

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

By the commutative properties of the z-transform, we can invert the order of computation to turn the Direct Form I structure into a new structure.

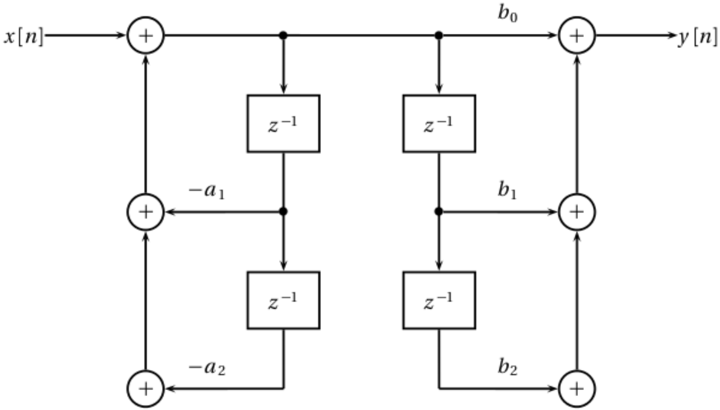


Figure 7.25 Direct form I with inverted order.

Direct form II IIR implementation

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

We can then combine the parallel delays together. This implementation is called Direct Form II; its obvious advantage is the reduced number of the required delay elements (hence of memory storage).

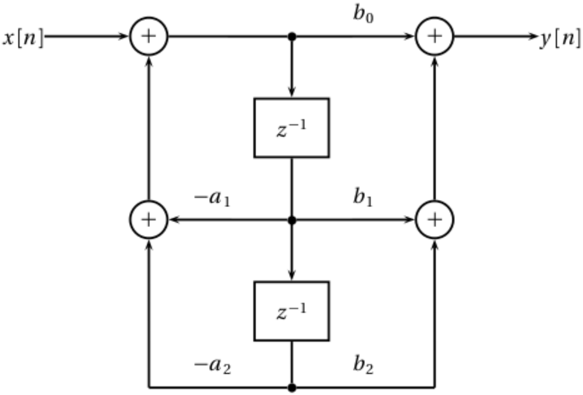


Figure 7.26 Direct Form II implementation of a second-order section.

IIR cascade implementation

The cascade structure of N second-order sections is much less sensitive to quantization than the previous Direct form II of order $2 \cdot N$.

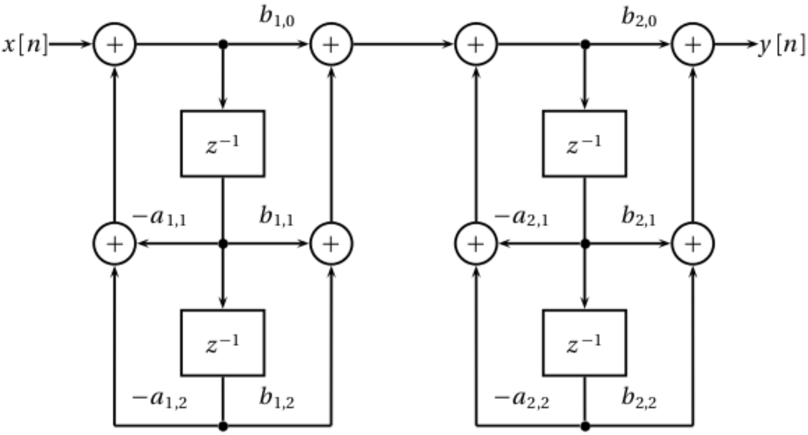


Figure 7.27 4th order IIR: cascade implementation.

Bibliography

- Alan V. Oppenheim and Ronald W. Schafer. *Discrete-time signal processing, 2nd Ed*. Prentice Hall. 1999. Chapter 7.
- Paolo Prandoni and Martin Vetterli. Signal processing for communications. Taylor and Francis Group, LLC. 2008. Sections 5.2, 5.3, and 7.4.
- Steven W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing. Chapters 14, 15, and 16. www.dspguide.com
- Oliver Hinton. Digital Signal Processing Resources for EEE305 Course. Chapters 4 and 5. www.staff.ncl.ac.uk/oliver.hinton/eee305/