Infinite impulse response filters

Dr. Ing. Rodrigo Gonzalez

rodralez@frm.utn.edu.ar

Técnicas Digitales III

Universidad Tecnológica Nacional,
Facultad Regional Mendoza.

Summary

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	ZOH method Bilinear z-transform

Leaky integrator filter

The MA filter equation,

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

$$y[n] = \frac{1}{M} \left[\sum_{k=1}^{M-1} x[n-k] + x[n] \right] . \tag{2}$$

Since,

$$y[n-1] = \frac{1}{M-1} \left[\sum_{k=1}^{M-1} x[n-k] \right] . \tag{3}$$

Then,

$$y[n] = \frac{1}{M} x[n] + \frac{M-1}{M} y[n-1] \,. \tag{4}$$

Defining $\lambda = \frac{M-1}{M}$,

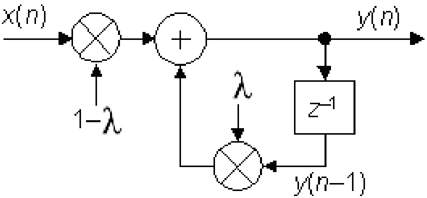
$$y[n] = \lambda \, y[n-1] + (1-\lambda) \, x[n] \,. \tag{5}$$

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

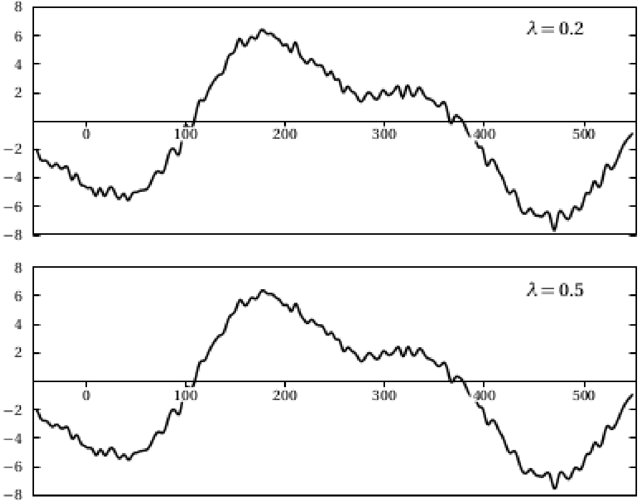
Leaky integrator filter

$$y[n] = \lambda\, y[n-1] + (1-\lambda)\, x[n]\,.$$

- No longer a convolution.
- Instead, a *constant coefficient difference equation*. Initial conditions must be set.
- The new system is LTI [2].
- System is stable for $|\lambda| < 1$.
- The value of λ (which is the pole of the system) determines the smoothing power of the filter.

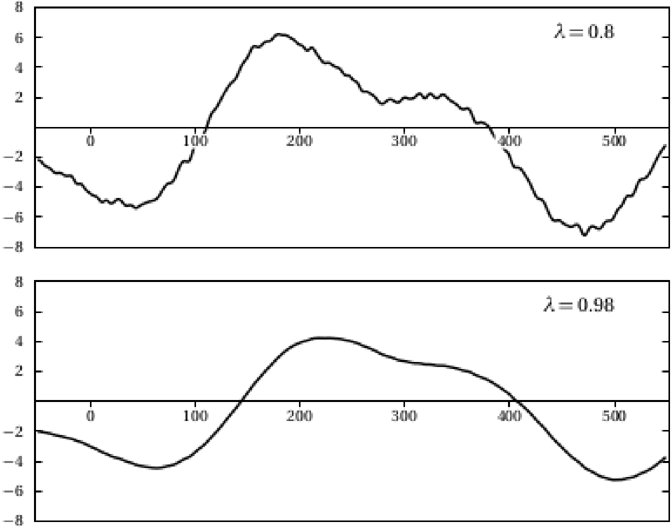


Noise Reduction



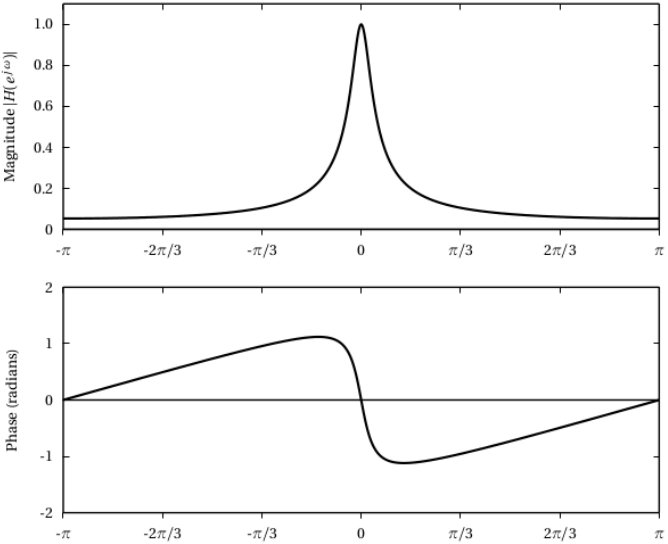
Noise Reduction

Note how the signal is delayed as λ grows.



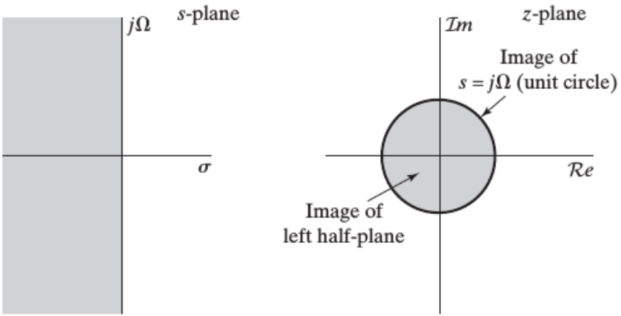
Frequency Response

Magnitude and phase response of the leaky integrator for $\lambda = 0.9$.



IIR filtering in frequency domain

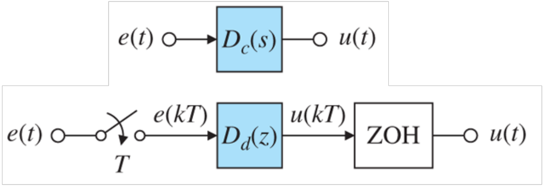
- The main idea is to transform an analog filter to the discrete domain.
- From *s* domain to *z* domain.
- This way, all the theory behind analog filter can be reused to implement a filter in a computer.



Zero-Order Hold (ZOH) Method

- ZOH Method assumes that the input signal remains constant throughout the sample period.
- Given an input, *e(k)*, the system is essentially responding to a positive step at *k*, followed by a negative step one cycle delayed.
- In other words, one input sample produces a square pulse of height, *e*.

$$\frac{U(z)}{E(z)} = \mathcal{Z}\{D_c(s)\} - z^{-1}\mathcal{Z}\{D_c(s)\} = (1 - z^{-1})\mathcal{Z}\{D_c(s)\}\,. \tag{6}$$



Suppose the following integrator,

$$\frac{U(s)}{E(s)} = D_c(s) = \frac{1}{s}\,. \tag{7}$$

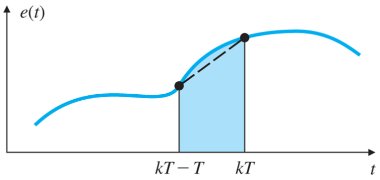
The area under $e(t)$ over kT periods is,

$$u(kT) = \int_0^{kT-T} e(t)dt + \int_{kT-T}^{T} e(t)dt\,. \tag{8}$$

Tustin's method uses the trapezoidal integration, to approximate $e(t)$ by a straight line between two samples.

$$u(kT) = u(kT-T) + \frac{T}{2}\left[e(kT-1) + e(kT)\right], \tag{9}$$

$$\Longrightarrow \frac{U(z)}{E(z)} = \frac{T}{2}\left(\frac{1+z^{-1}}{1-z^{-1}}\right) = \frac{1}{\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}}\,. \tag{10}$$



Comparing Eq. 7 and 10,

$$s = \frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) . \tag{11}$$

The technique is an algebraic transformation between variables s and z.
Doing $s = j\Omega$, where Ω is the analog frequency, $-\infty, < \Omega < \infty$,

$$z = \frac{1 + (T/2)j\Omega}{1 - (T/2)j\Omega} . \tag{12}$$

The relationship between Ω and ω, the "digital" frequency, $-\pi, < \omega < \pi$, can be found by replacing $z = e^{j\omega}$ in Eq. 11,

$$s = \frac{2}{T}\left(\frac{1 - e^{-j\omega}}{1 + e^{-j\omega}}\right) = \sigma + j\Omega = \frac{2}{T}\left[\frac{2e^{-j\omega/2}(j\sin\omega/2)}{2e^{-j\omega/2}(\cos\omega/2)}\right] = j\frac{2}{T_d}\tan(\omega/2) . \tag{13}$$

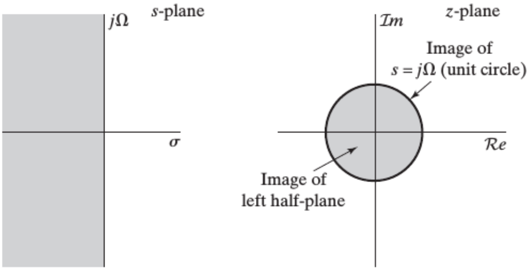
Real and imaginary parts on both sides of Eq. 13 are,

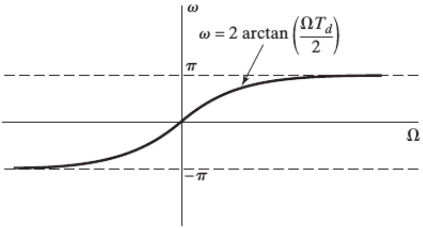
$$\sigma = 0 , \tag{14}$$

$$\Omega = \frac{2}{T}\tan(\omega/2) , \tag{15}$$

$$\omega = \arctan(\Omega T/2) . \tag{16}$$

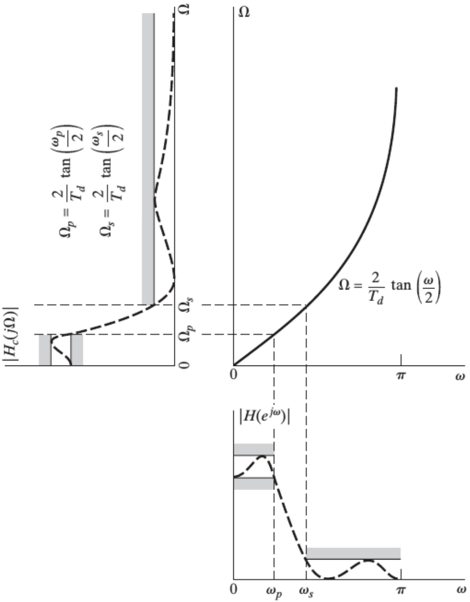
Map from s to z





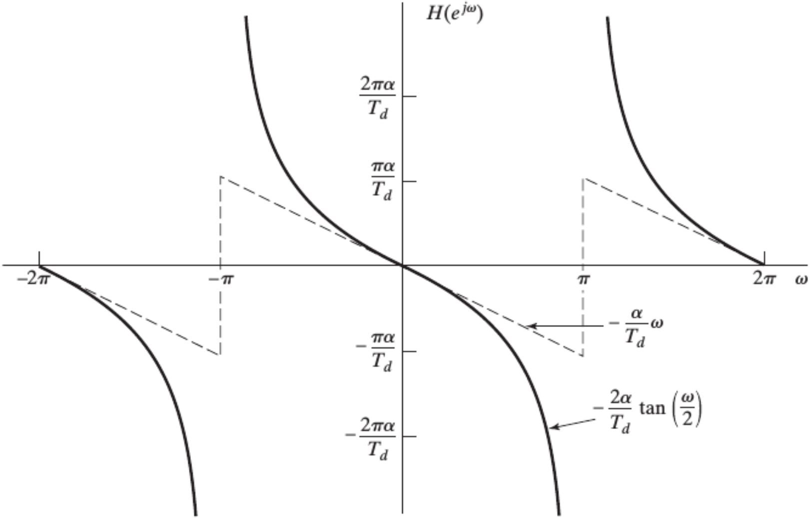
Bilinear transform, Frequency Warping

- Non-linear compression of the frequency axis.
- The design of discrete-time filters using the bilinear transformation is useful only when this compression can be tolerated or compensated for.



Bilinear transform, Phase response

Suppose a continuous-time filter with linear phase response. The nonlinear warping of the frequency axis introduced by the bilinear transformation will not preserve linearity in phase response.



Example of IIR design using bilinear transform [3]

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 using bilinear transform [3] II

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.

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

Direct form I IIR implementation

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

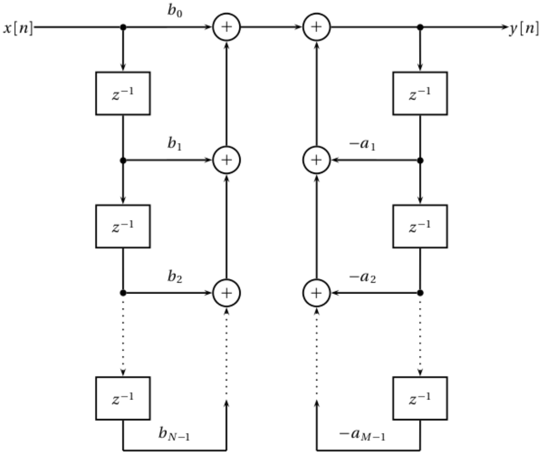


Figure 7.24 Direct Form implementation of an IIR filter.

Direct form I IIR implementation inverted

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

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.

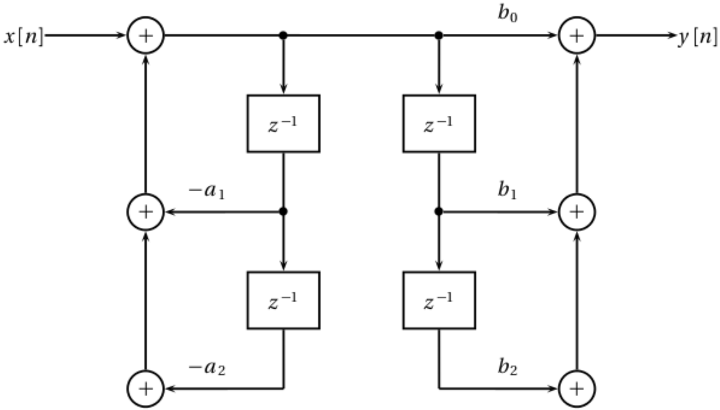


Figure 7.25 Direct form I with inverted order.

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).

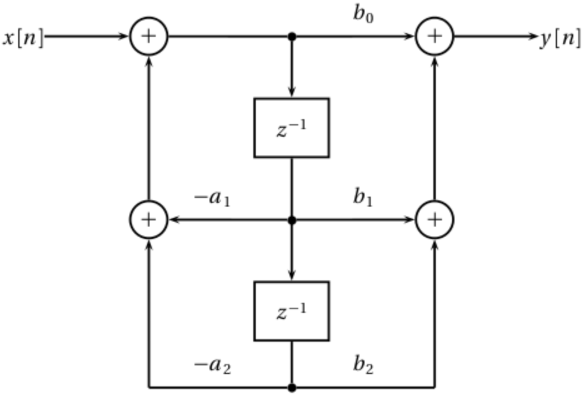


Figure 7.26 Direct Form II implementation of a second-order section.

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 $2 \cdot N$.

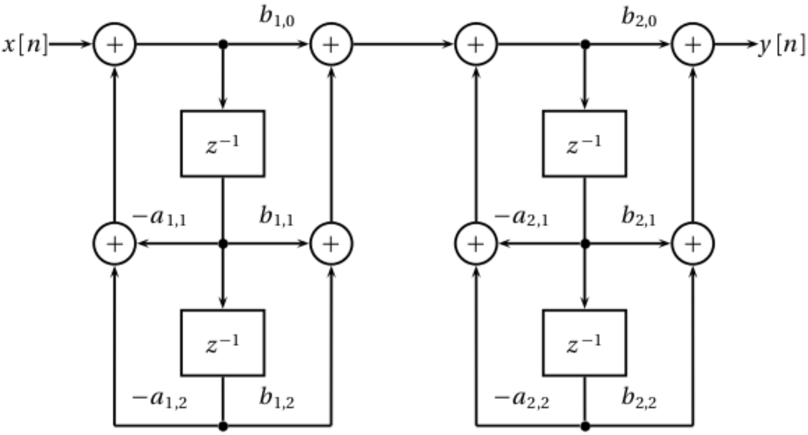


Figure 7.27 4th order IIR: cascade implementation.

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.

Bibliography

1 Alan V. Oppenheim and Ronald W. Schafer. *Discrete-time signal processing, 3rd Ed*. Prentice Hall. 2010. Sections 7.2 and 7.3.

2 Paolo Prandoni and Martin Vetterli. Signal processing for communications. Taylor and Francis Group, LLC. 2008. Sections 5.3.2, 7.3, and 7.4.2.

3 Oliver Hinton. Digital Signal Processing Resources for EEE305 Course. Chapter 5. www.staff.ncl.ac.uk/oliver.hinton/eee305/

4 Gene F. Franklin, J. David Powell and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems*. 7th Edition. 2014. Section 8.3.