12. The z-transform

1. The z-domain
2. Analysis of recursive systems
3. Filter design in practice

Textbook: [Smith, ch. 33]

1. The z-domain

Just as **analog filters** are designed using the **Laplace transform**, **recursive digital filters** are developed with a technique called the **z-transform**.

The **z-transform** deals with:

- **difference equations**,
- the **z-domain**, and
- the **z-plane** (the z-plane uses a polar format).

Recursive digital filters are often designed by starting with one of the classic analog filters, such as the **Butterworth, Chebyshev**, or **elliptic**.

The design **strategy**: **probe the impulse response** with **sinusoids and exponentials** to find the system's **poles** and **zeros**.

1) The Laplace transform of discrete-time signals

Recall the **Laplace transform** and the s-domain signal:

$$\boxed{X(s) = \int_{t=-\infty}^{\infty} x(t)\, e^{-st}\, dt}$$

where $x(t)$ and $X(s)$ are the time domain and s-domain representations.

Using the alternate notation, the Laplace transform becomes:

$$X(\sigma,\omega) = \int_{t=-\infty}^{\infty} [x(t)\, e^{-\sigma t}]\, e^{-j\omega t}\, dt$$

If we are only concerned with ***real* time domain signals** (the usual case), the top and bottom halves of the s-plane are **mirror images** of each other.

$e^{-j\omega t}$ reduces to **cosine** and **sine waves**.

Sampled signal

- Let us assume, the variable, $t = mT_s$, represents sampling time points.
- The signal $x(mT_s)$, normally denoted as $x[n]$, is a **discrete-time (a sampled) signal** with sampling period T_s .
- With the sampling period $T_s = 1$: $m = n$.

Thesis. The Laplace transform of a sampled signal is **periodic** with respect to the **frequency axis** jω of the complex frequency variable: $s = \sigma + j\omega$.

Proof. For the sampled signal $x[n]$, the Laplace transform becomes

$$X(s) = \sum_{n=-\infty}^{\infty} x_n \cdot e^{-sn}$$

Assume, $s' = s + j\,k\,2\pi$, where k is an integer variable. Then:

$$X(s') = X(s + jk2\pi) = \sum_{n=-\infty}^{\infty} x_n \cdot e^{-(s+jk2\pi)n} =$$

$$= \sum_{n=-\infty}^{\infty} x_n \cdot e^{-sn} e^{-jk2\pi n} = \sum_{n=-\infty}^{\infty} x_n \cdot e^{-sn} = X(s)$$

Notice: $e^{-jk2\pi n} = 1$. Hence the Laplace transform of a discrete-time signal is periodic with a **period of 2π** as.

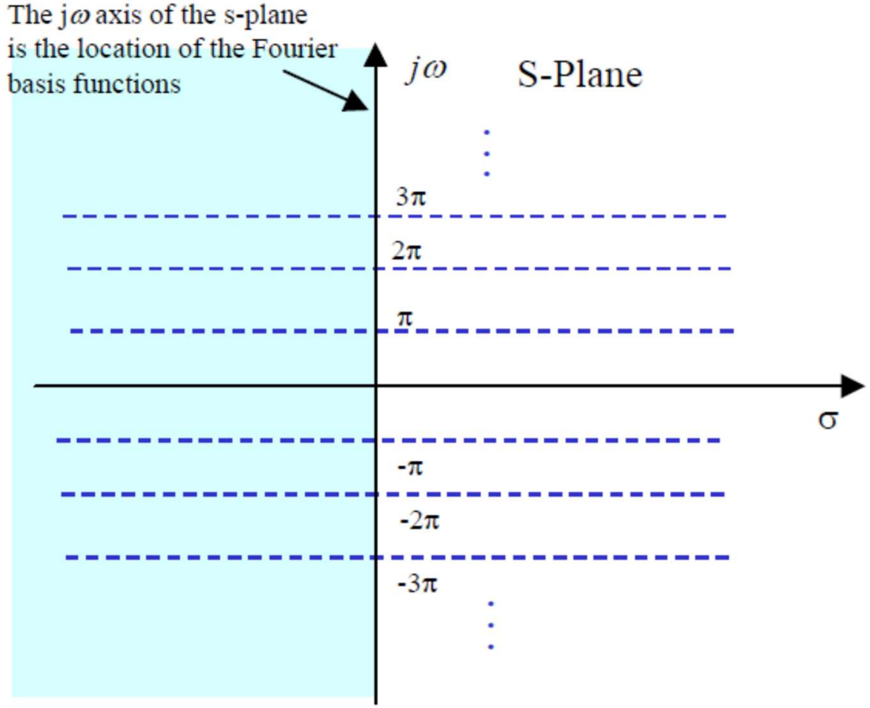


Fig. 1.

2) The z-transform

The **Laplace transform** can be **changed** into the **z-transform** in 3 steps:

1. Change from **continuous** to **discrete** time signals

   Replace the time variable, t, with the sample number, n, and change the integral into a summation:

$$X(\sigma, \omega) = \sum_{n=-\infty}^{\infty} (x_n \; e^{-\sigma n} \; e^{-j\omega n})$$

   Notice that $X(\,)$ is ***continuous***, not discrete.

2. Rewrite the **exponential term**.

   An exponential signal can be represented in either of two ways:

$$y_n = e^{\sigma n}$$

or

$$y_n = r^n$$

$$y_n = e^{\sigma n}$$

The first form controls the decay of the signal through the **parameter σ**:
- If σ is negative, the waveform will *decrease* in value as *n* becomes larger. The curve will *increase* if σ is positive.
- If σ is exactly zero, the signal will have a constant value of *one*.

$$y_n = r^n$$

The second expression uses the ***parameter r*** to control the decay:
- The waveform will decrease if $r < 1$, and increase if $r > 1$,
- The signal will have a constant value when $r = 1$.

These two equations are just different ways of expressing the same thing:

$$r^n = [e^{\ln(r)}]^n = e^{n \ln(r)} = e^{\sigma n},$$

where

$$\boxed{\sigma = \ln(r)}.$$

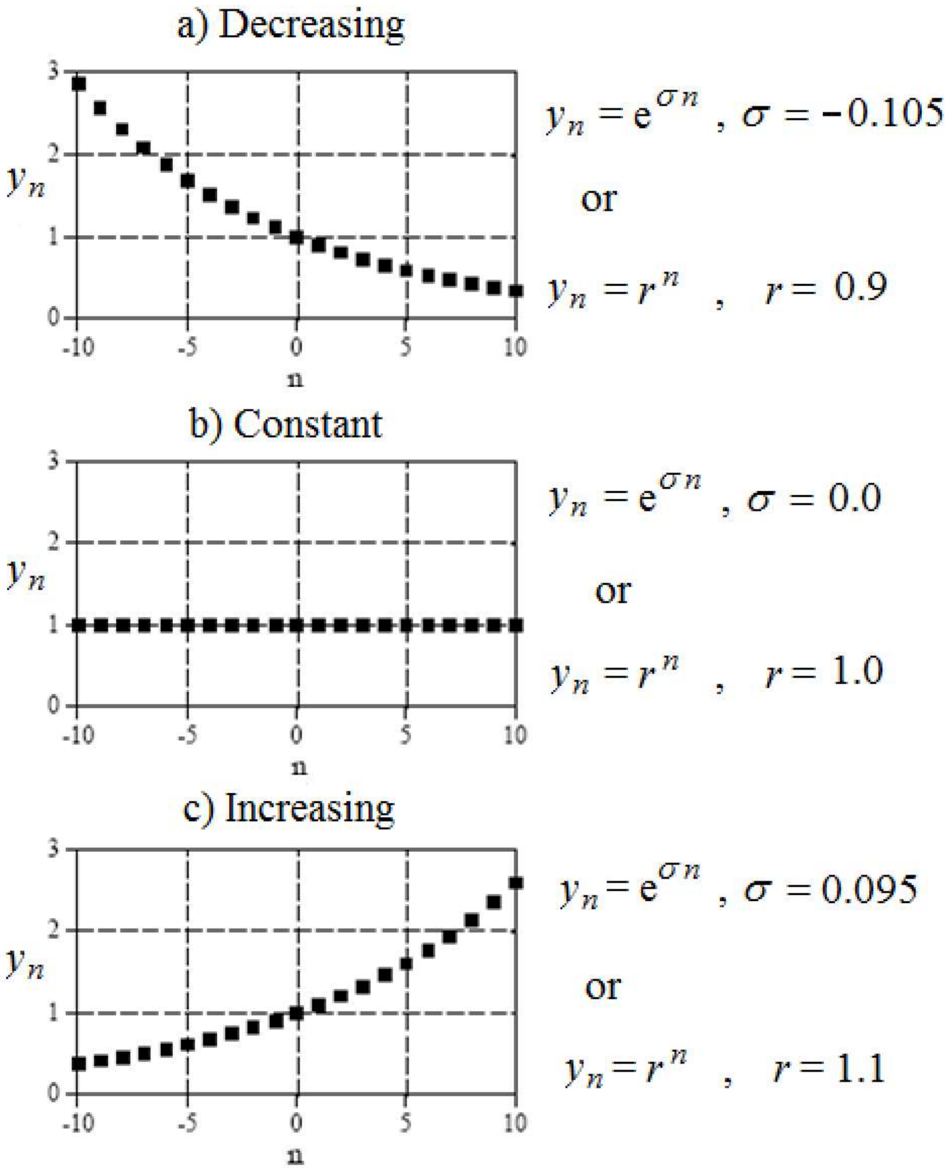


Fig. 2 Parameters σ and *r* of the exponential signals.

Then use the ***other*** exponential form:

$$X(r,\omega) = \sum_{n=-\infty}^{\infty} x_n \, r^{-n} \, e^{-j\omega n}$$

3. Finally define a **new variable** for the z-transform:

$$\boxed{z = r\, e^{j\omega}}$$

The complex variable, *z*, in the polar notation, is a combination of the two real variables, *r* and ω.

Replace *r* and ω, with *z*. This gives the z-transform in **standard form**:

$$X(z) = \sum_{n=-\infty}^{\infty} x_n \, z^{-n} \qquad (12.1)$$

2) Properties of the z-domain

The **z-domain** uses the variables, *r* and ω, arranged in ***polar* coordinates**:

- the distance from the origin, *r*, is the value of the **exponential decay**;
- the angular distance measured from the positive horizontal axis, ω, is the **frequency**.

Vertical lines in the s-plane are matching **circles** in the z-plane.

- s-plane's vertical axis (σ =0) corresponds to z-plane's **unit circle** (*r*=1).
- Vertical lines in the **left half** of the s-plane correspond to circles **inside** the z-plane's unit circle.
- Vertical lines in the **right half** of the s-plane match with circles on the **outside** of the z-plane's unit circle.

Poles

- Periodic processes can be conveniently represented using a circular **polar diagram** such as the z-plane and its associated **unit circle**.
- A discrete system is **unstable** when poles are **outside** the unit circle in the z-plane.

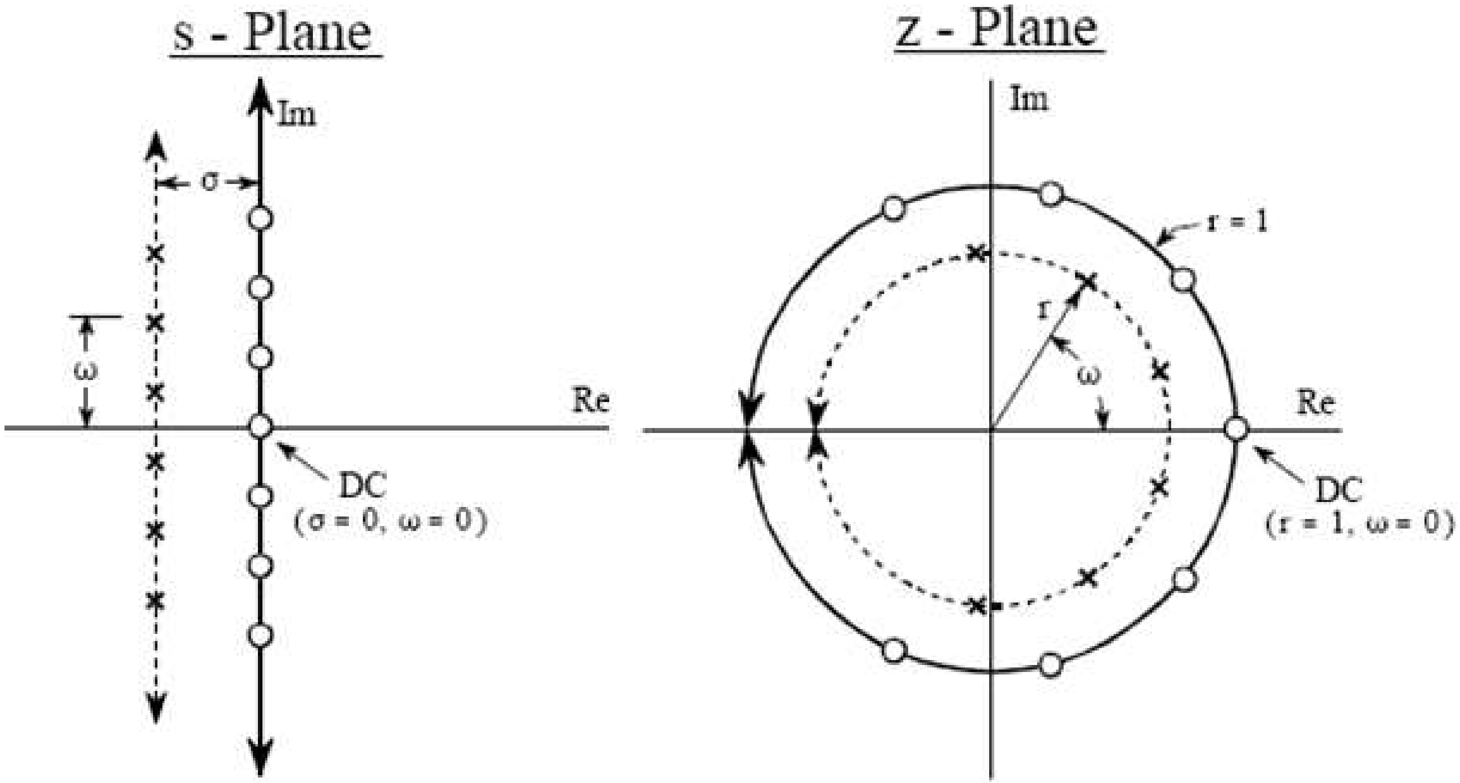


Fig. 3 Relationship between the s-plane and the z-plane

Imagine bending the jω−axis of the *s*-plane of the sampled signal of Fig. 3 in the direction of the left hand side half of the *s*-plane to form a circle such that the points π and –π meet. The resulting circle is called the *unit circle*, and the resulting diagram is called the *z*-plane.

Values

- When the time domain signal is completely real (the most common case), the upper and lower halves of the z-plane are **mirror** images of each other.

Frequencies

- A ***continuous*** sinusoid can have any frequency between DC and infinity – in the s-plane ω runs from negative to positive infinity.
  But a ***discrete* sinusoid** can only have a frequency **between DC** and **one-half** the sampling rate, i.e. the frequency must be between 0 and 0.5 (as a fraction of the sampling rate) or between 0 and π (as a **natural frequency** (i.e., $\boxed{\omega = 2\pi f}$)).
- In the z-plane we interpret ω to be an **angle** expressed in *radians* - the positive frequencies correspond to angles of 0 to π radians, while the negative frequencies – of 0 to $-\pi$ radians.

The **frequency response** in the z-domain

- It is found **along the unit circle** – evaluate the z-transform at $r = 1$ – this reduces it to the Discrete Time Fourier Transform (DTFT).

- Zero frequency (**DC**) is at a **value of *one*** on the horizontal axis in the z-plane.

- The **positive frequencies** are counted in a counter-clockwise pattern from DC position - upper semicircle.

- The **negative frequencies** are arranged from the DC along the ockwise path, forming the lower semicircle.

- The positive and negative frequencies meet at the common point: $\omega=\pi$, $\omega=-\pi$. This circular geometry corresponds to the ***periodic*** **frequency** spectrum of a discrete signal.

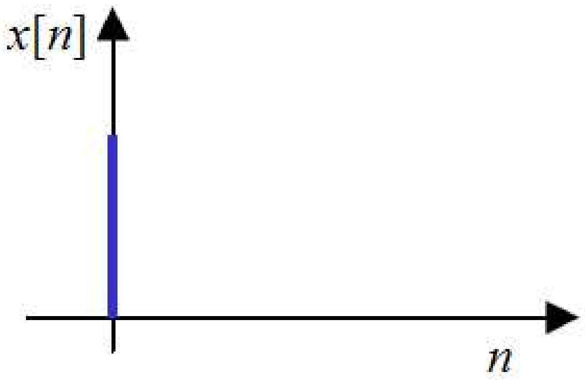
The region of convergence (ROC)

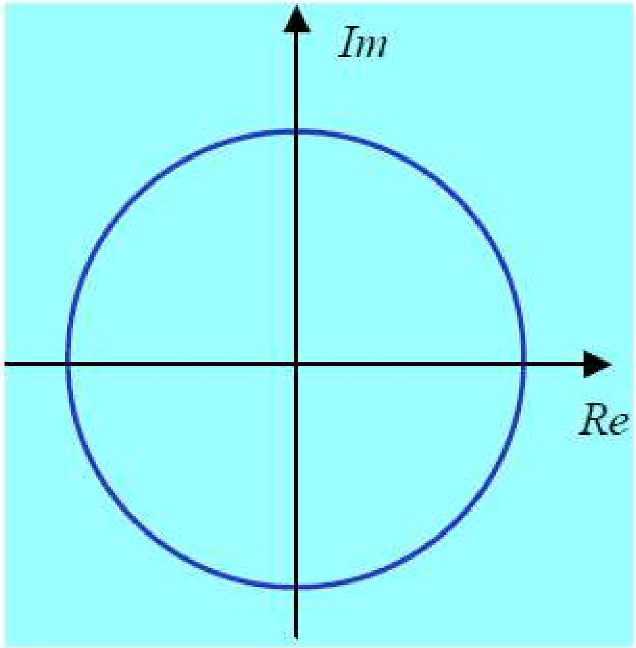
Since the z-transform is an infinite power series, it exists only for those values of the variable z for which the series converges to a finite sum.

The region of convergence (**ROC**) of $X(z)$ is the set of all the values of z for which $X(z)$ attains a finite computable value.

Example 1. The unitary impulse.

$$x[n] = \delta[n] = \begin{cases} 1 \ , n = 0 \\ 0 \ , n \neq 0 \end{cases}$$





Its region of convergence is the entire z-plane, as its z-transform is:

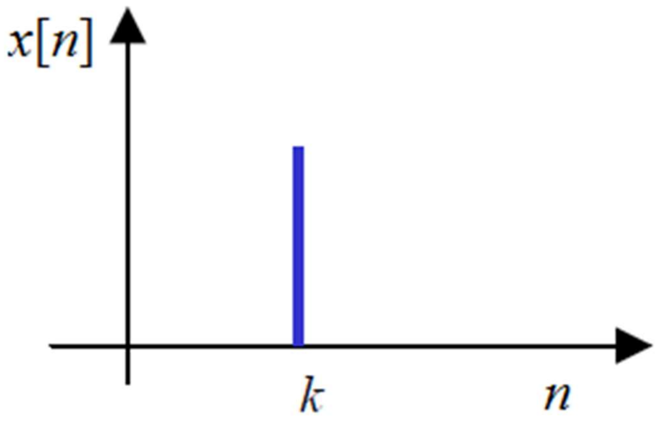
$$X(z) = \sum_{n=-\infty}^{\infty} x_n \, z^{-n} = \delta(0) \cdot z^0 = 1$$

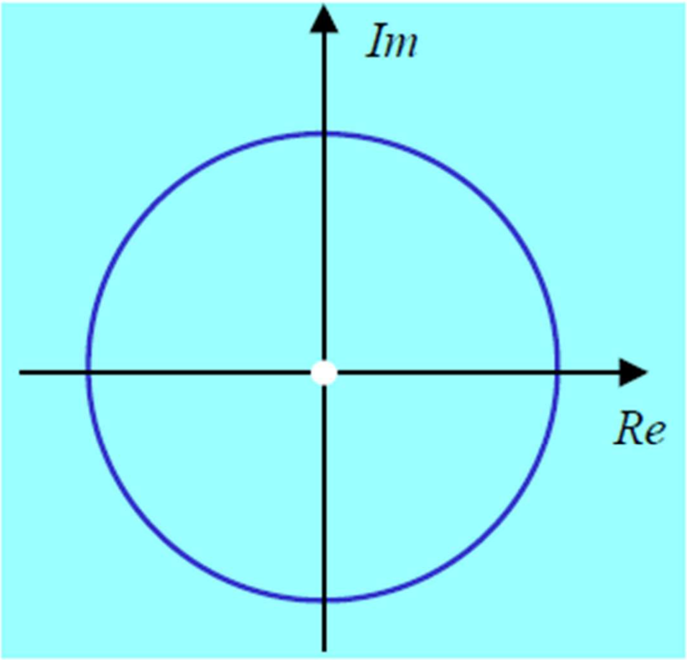
Hence, independent of z the value is always finite and equal 1.

Its Fourier transform: $X(z = e^{j\omega}) = 1$

Example 2. The shifted impulse.

$$x[n] = \delta[n - k] = \begin{cases} 1 \ , n = k \\ 0 \ , n \neq k \end{cases}$$





Its *z*-transform is:

$$\boldsymbol{X(z) = \sum_{n=-\infty}^{\infty} \delta_{n-k}\, z^{-n} = z^{-k} = \frac{1}{z^{k}}}$$

Hence $X(z)$ is of finite value for all *z*-s except of the origin point: z=0. The region of convergence is the entire *z*-plane except the point z=0.

Its Fourier transform is:

$$X(z = e^{j\omega}) = e^{-j\omega k}$$

3) Properties

Linearity of transform

Property	Laplace transform		Z transform
Additivity (superposition)	$\mathsf{L}(u+v)= \mathsf{L}(u)+\mathsf{L}(v)= U(s) + V(s)$		$\mathsf{Z}(u+v)= \mathsf{Z}(u)+\mathsf{Z}(v)= U(z) + V(z)$
Scaling	$\mathsf{L}(au)= a\, \mathsf{L}(u) = a\, U(s)$		$\mathsf{Z}(au)= a\, \mathsf{Z}(u) = a\, U(z)$

Dual operations in the time- and z-domain (or s-domain)

Operation in the time domain	Time domain	$\leftrightarrow$	z-domain (s-domain) operation
Time shift	$x[n-k]$		$z^{-k} \cdot X(z)$
Amplitude gain	$g\, x[n]$		$g\, X(z)$
Modulation	$e^{at} \cdot x(t)$		$X(s - a)$
Convolution	$x[n] * \mathrm{y[n]}$		$X(z)\, Y(z)$
Time reversal	$x[-n]$		$X(1/z)$
Product	$x[n] \cdot \mathrm{y[n]}$		$X(z) * Y(z)$

Exercises

Task 12-1

Obtain the z-transform of the signal, $x[n] = a^n$, in 3 cases:

a) a causal signal, $n \in <0, +\infty>$

b) an unlimited signal domain, $n \in <-\infty, +\infty>$

c) a finite, causal signal, $n \in <0, N-1>$

2. Analysis of Recursive Systems

1) Difference equation

A recursive filter is described by a **difference equation**:

$$y_n = a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2} + \cdots a_M x_{n-M} \\ + b_1 y_{n-1} + b_2 y_{n-2} + \cdots + b_N y_{n-N} \qquad (12.2)$$

where $x[n]$ and $y[n]$ are the input and output signals, respectively, and the "a" and "b" terms are the **recursion coefficients**.

Recursive discrete systems operate by this difference equation.

From this equation we can derive the key characteristics of the system:

1. the **impulse** response,
2. **step** response,
3. **frequency** response,
4. **pole-zero** plot.

2) System's transfer function

To transform the **convolution operation**

$$\boldsymbol{y}[\boldsymbol{n}] = \boldsymbol{x}[\boldsymbol{n}] * \boldsymbol{h}[\boldsymbol{n}] \quad \leftrightarrow \quad \boldsymbol{y_n} = \sum_{k=-\infty}^{\infty} x_{n-k} \cdot h_k$$

is the **main goal** of using a z-transform:

$$\boldsymbol{Y}(\boldsymbol{z}) = \sum_{\boldsymbol{n}=-\infty}^{\infty} \left[\sum_{\boldsymbol{k}=-\infty}^{\infty} x_{n-k} \cdot h_k \right] \boldsymbol{z}^{-\boldsymbol{n}} = \sum_{\boldsymbol{k}=-\infty}^{\infty} \left[\boldsymbol{h}_k \sum_{\boldsymbol{n}=-\infty}^{\infty} x_{n-k}\ \boldsymbol{z}^{-\boldsymbol{n}} \right] =$$

(substitute: $m = n\text{-}k$; so: $-n = -k - m$)

$$\boldsymbol{Y}(\boldsymbol{z}) = \sum_{\boldsymbol{k}=-\infty}^{\infty} \left[\boldsymbol{h}_k \sum_{\boldsymbol{m}=-\infty}^{\infty} x_m \boldsymbol{z}^{-\boldsymbol{k}-\boldsymbol{m}} \right] = \sum_{\boldsymbol{k}=-\infty}^{\infty} \boldsymbol{h}_k \boldsymbol{z}^{-\boldsymbol{k}} \cdot \sum_{\boldsymbol{m}=-\infty}^{\infty} x_m \boldsymbol{z}^{-\boldsymbol{m}}$$

$$\rightarrow \quad \boldsymbol{Y}(\boldsymbol{z}) = H(\boldsymbol{z}) \cdot X(\boldsymbol{z})$$

In above equation, $H(z)$ is called "the **transfer function**" of a system.

A **system** is characterized in the z-domain by its **transfer function**:

$$H(z) = \frac{Y(z)}{X(z)} \tag{12.3}$$

The **transfer function** is the z-domain representation of the output signal **divided** by the z-domain representation of the input signal.

3) The H1 transfer function form

For a **recursive filter** we can show by algebraic transformations that (**H1**):

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + \cdots + a_M z^{-M}}{1 - b_1 z^{-1} - b_2 z^{-2} - b_3 z^{-3} \cdots - b_N z^{-N}} \tag{12.4}$$

- This form (**H1**) **directly** contains the **recursion coefficients**.
- Notice that the "b" coefficients enter the transfer function with a ***negative*** sign in front of them.

Proof of form H1

Observe that the recursive equation (11.2) can be rewritten as:

$$y_n - \sum_{k=1}^{N} b_k y_{n-k} = \sum_{k=0}^{M} a_k \; x_{n-k}$$

Consider the simplified case when N=0 (this will be an **FIR filter**):

$$y_n = \sum_{k=0}^{M} a_k \; x_{n-k}$$

$$\boldsymbol{Y(z)} = \sum_{\boldsymbol{n=-\infty}}^{\infty} \left[\sum_{\boldsymbol{k=0}}^{\boldsymbol{M}} \boldsymbol{a_k} \cdot x_{n-k} \right] \boldsymbol{z^{-n}} = \sum_{\boldsymbol{k=0}}^{\boldsymbol{M}} \left[\boldsymbol{a_k} \sum_{\boldsymbol{n=-\infty}}^{\infty} x_{n-k} \; \boldsymbol{z^{-(n-k)-k}} \right]$$

$$Y(z) = \sum_{k=0}^{M} a_k \; X(z) \; z^{-k} \; = X(z) \; \sum_{k=0}^{M} a_k \; z^{-k}$$

$$H(z) = \sum_{k=0}^{M} a_k \; z^{-k}$$

$$Y(z) = X(z) \cdot H(z)$$

The general case (**IIR**):

$$y_n - \sum_{k=1}^{N} b_k y_{n-k} = \sum_{k=0}^{M} a_k \; x_{n-k}$$

$$\sum_{\boldsymbol{n}=-\infty}^{\infty} \left[y_n - \sum_{\boldsymbol{k}=1}^{N} b_{\boldsymbol{k}} \cdot y_{n-k} \right] \boldsymbol{z^{-n}} = \sum_{\boldsymbol{n}=-\infty}^{\infty} \left[\sum_{\boldsymbol{k=0}}^{\boldsymbol{M}} \boldsymbol{a_k} \cdot x_{n-k} \right] \boldsymbol{z^{-n}}$$

$$Y(z) - \sum_{k=1}^{N} b_k \; Y(z) \; z^{-k} = \sum_{k=0}^{M} a_k \; X(z) \; z^{-k}$$

$$Y(z) \; (1 - \sum_{k=1}^{N} b_k \; z^{-k} \;) = X(z) \; \sum_{k=0}^{M} a_k \; z^{-k}$$

$$Y(z) \; = X(z) \; \frac{\sum_{k=0}^{M} a_k \; z^{-k}}{1 - \sum_{k=1}^{N} b_k \; z^{-k}}$$

$$H(z) = \frac{\sum_{k=0}^{M} a_k \; z^{-k}}{1 - \sum_{k=1}^{N} b_k \; z^{-k}}$$

Example 3.

Suppose we know the recursion coefficients of some digital filter:

$a_0 = 0.389$

$a_1 = -1.558$	$b_1 = 2.161$
$a_2 = 2.338$	$b_2 = -2.033$
$a_3 = -1.558$	$b_3 = 0.878$
$a_4 = 0.389$	$b_4 = -0.161$

We can directly write down the system's transfer function:

$$H(z) = \frac{0.389 - 1.558z^{-1} + 2.338z^{-2} - 1.558z^{-3} + 0.389z^{-4}}{1 - 2.161z^{-1} + 2.033z^{-2} - 0.878z^{-3} + 0.161z^{-4}}$$

Above equation expresses the transfer function using *negative* powers of z. A more conventional form that uses ***positive* powers** is obtained by multiplying both the numerator and denominator by z^4:

$$H(z) = \frac{0.389z^4 - 1.558z^3 + 2.338z^2 - 1.558z + 0.389}{z^4 - 2.161z^3 + 2.033z^2 - 0.878z + 0.161}$$

Properties of the transfer function

The transfer function of a recursive system is useful because it can express:

1) combining **cascade** and **parallel** stages into a single system,
2) designing filters by specifying the **pole** and **zero** locations,
3) converting **analog** filters into **digital**.

They are expressed by algebraic operations performed in the z-domain:

- **multiplication**, **addition**, and **factoring**.

After these operations are completed, the transfer function is placed in the **form H1** of $H(z)$, allowing the new **recursion coefficients** to be identified.

4) H2 form of the transfer function

The transfer function can be expressed by **poles** and **zeros**. This provides the second general form in the z-domain (the **pole-zero form**):

$$H(z) = \frac{(z - z_1)(z - z_2)(z - z_3)\cdots}{(z - p_1)(z - p_2)(z - p_3)\cdots} \tag{12.5}$$

Remarks

- Moving from **form (H2)** to **form (H1)** requires only *multiplication* and *reordering* of expressions. .
- Moving **from (H1)** to **(H2)** is more difficult because it requires *factoring* of the polynomials.
- The quadratic equation can be used for the factoring if the transfer function is **second** order or less.
- To factor systems greater than second order numerical methods must be used.

3. Filter design in practice

Digital filter design *starts* with the **pole-zero locations** (H2) and *ends* with the **recursion coefficients** (H1).

1) Notch filter design

We can design a **notch filter** by the following steps:

(1) Specify the pole-zero placement in the z-plane,

(2) Write down the transfer function (H2),

(3) Rearrange the transfer function into form (H1), and

(4) Identify the recursion coefficients needed to implement the filter.

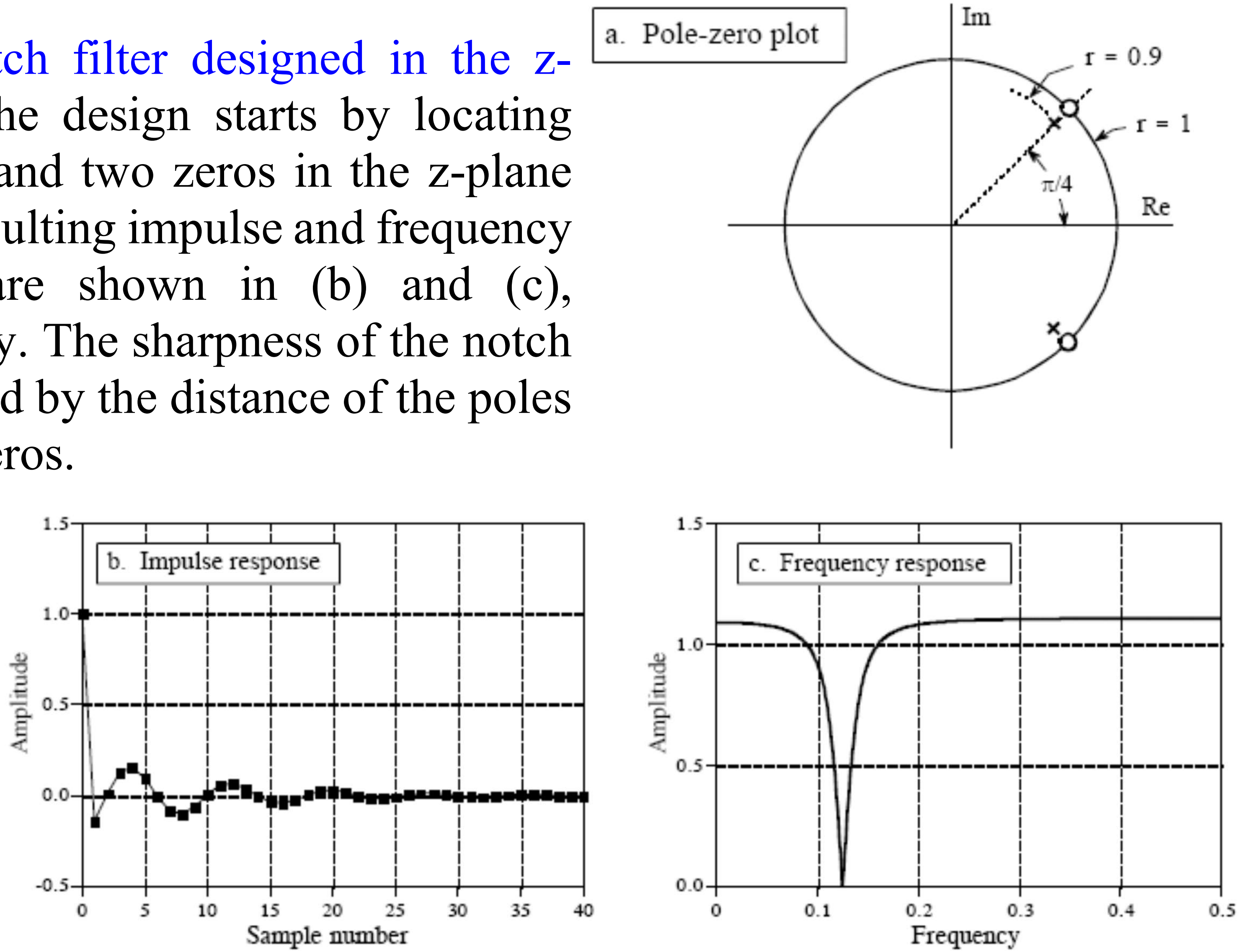
Exercises

Task 12-2

Design a notch filter with two zeros and poles given as follows:

$$z_1 = 1.0\, e^{j(\pi/4)}, z_2 = 1.0\, e^{j(-\pi/4)}, p_1 = 0.9\, e^{j(\pi/4)}, p_2 = 0.9\, e^{j(-\pi/4)}.$$

Fig. 4 Notch filter designed in the z-domain. The design starts by locating two poles and two zeros in the z-plane (a). The resulting impulse and frequency response are shown in (b) and (c), respectively. The sharpness of the notch is controlled by the distance of the poles from the zeros.



2) A *biquad* system

A very useful system, containing two poles and two zeros, is called as **biquad**. It has the following direct relations for recursion coefficients:

$$\boxed{\begin{aligned} a_0 &= 1 \\ a_1 &= -2r_0\cos(\omega_0) \\ a_2 &= r_0^2 \end{aligned}}$$

$$\boxed{\begin{aligned} b_1 &= 2r_p\cos(\omega_p) \\ b_2 &= -r_p^2 \end{aligned}}$$

where the positions of the poles and zeros are: (r_p , ω_p), (r_0, ω_0).

3) Frequency response from the transfer function

How do we find the **frequency response** having the transfer function?

- **The analytical method** - to find the values in the z-plane that lie on the **unit circle**, i.e. evaluate the transfer function, $H(z)$, at $r = 1$.
  Replace z with $e^{-j\omega}$ (that is, $re^{-j\omega}$ with r=1) in the function (H1) or (H2).
  This provides a **mathematical equation** of the frequency response, $\boldsymbol{H}(\omega)$.
- **The computational method** – to calculate ***samples*** of the frequency response, not a mathematical solution for the entire curve; i.e. get samples at N equally spaced frequencies between $\omega = 0$ and $\omega = \pi$.
- **The measurement-based method** - find the impulse response of the filter by passing an impulse through the system, then take the DFT of the impulse response to find the system's frequency response (the discarded samples must be *insignificant*).

4) Cascade and Parallel Stages

Sophisticated recursive filters are usually designed in stages. The two common ways that individual stages can be arranged are:

1. **cascaded stages** and
2. **parallel stages** with **added** outputs.

The frequency responses of systems in a **cascade** are combined by ***multiplication***.

The frequency responses of systems in **parallel** are combined by ***addition*****.**

These same operations (multiplication or addition) are performed on the z-domain **transfer functions**.

For example:

- a low-pass and high-pass stage can be cascaded to form a *band-pass* filter;
- parallel low-pass and high-pass stages can form a *band-reject* filter.

Example 4 (fig. 5)

- The two stages being combined, *system* 1 and *system* 2, with their recursion coefficients: a_0, a_1, a_2, b_1, b_2 and A_0, A_1, A_2, B_1, B_2, respectively.
- The combined single recursive filter, call it *system* 3, with recursion coefficients given by: a_0, a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4.

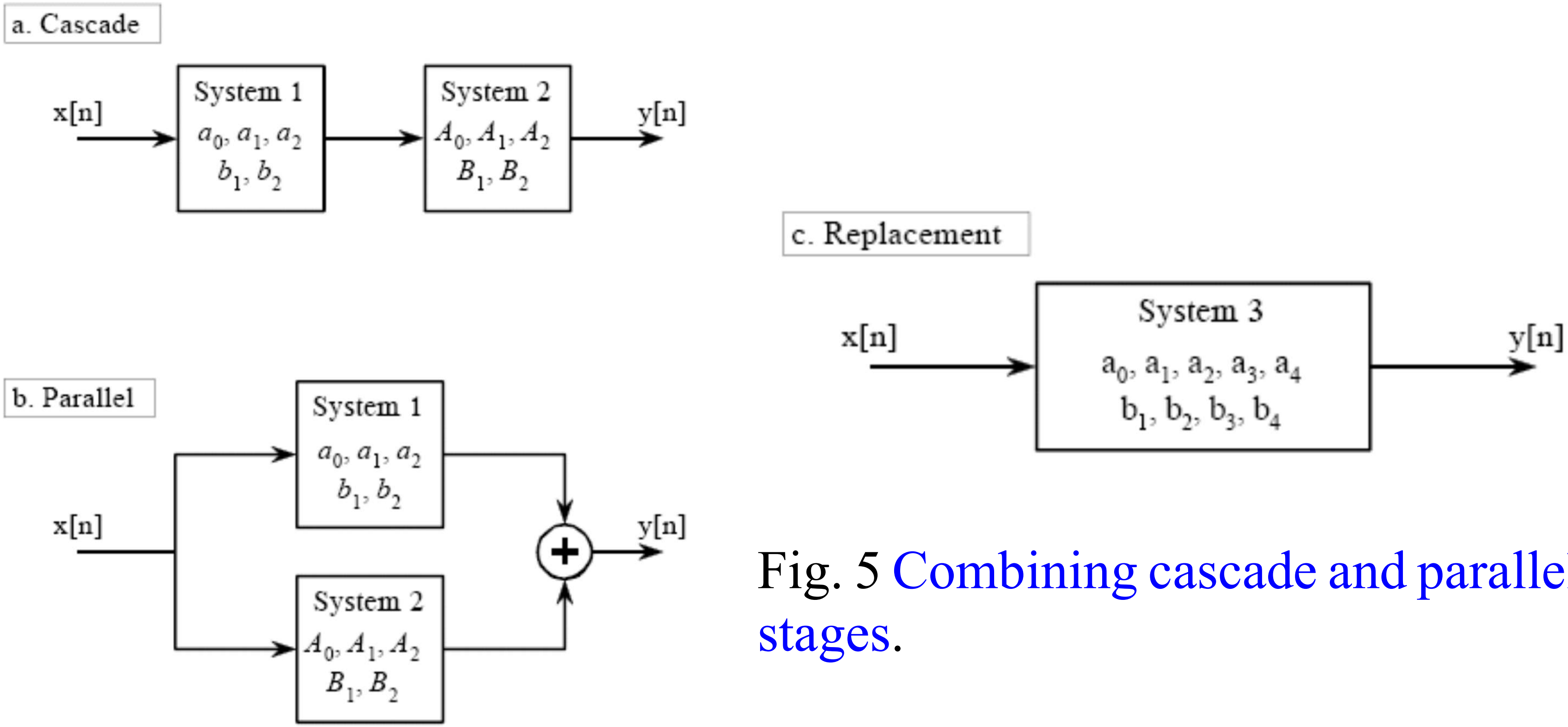


Fig. 5 Combining cascade and parallel stages.

Exercises

Task 12-3 Combine two ***biquad* stages in a cascade**.

- The two stages being combined, *system* 1 and *system* 2, with their recursion coefficients: a_0, a_1, a_2, b_1, b_2 and A_0, A_1, A_2, B_1, B_2, respectively.
- The combined single recursive filter, with recursion coefficients given by: a_0, a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4.

Task 12-4 Combine two ***biquad* stages in parallel**.

5) Spectral Inversion

The FIR filter technique called ***spectral inversion*** is a way of changing the filter kernel such that the frequency response is flipped top-for-bottom. All the pass-bands are changed into stop-bands, and vice versa.

For example, a low-pass filter is changed into high-pass, a band-pass filter into band-reject, etc.

Spectral inversion for an IIR filter (fig. 6) - subtracting the output of the system from the original signal.

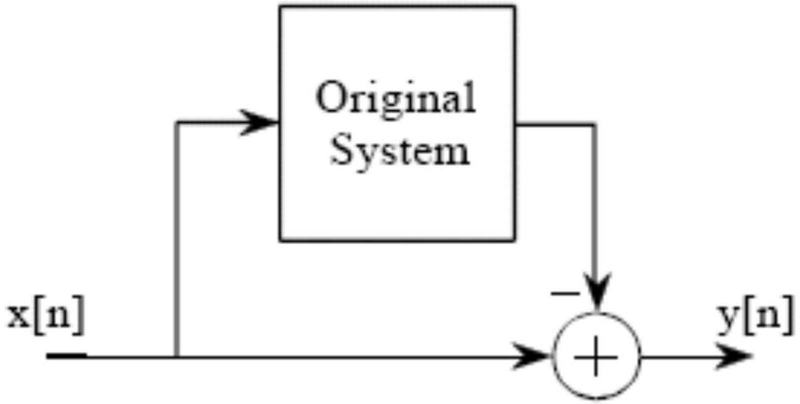


Fig. 6 Spectral inversion for an IIR filter. This procedure is the same as subtracting the output of the system from the original signal.

Spectral inversion for an IIR filter:

- Can be viewed as combining two stages **in parallel**, where one of the stages is the ***identity system*** (the output is identical to the input);
- It can be shown that the "b" coefficients are left unchanged, and the modified "a" coefficients are given by:

$$\mathrm{a}_0 = 1 - a_0$$

$$\mathrm{a}_1 = -a_1 - b_1$$

$$\mathrm{a}_2 = -a_2 - b_2$$

$$\mathrm{a}_3 = -a_3 - b_3$$

$$\ldots$$

The results of spectral inversion for IIR filter are **disappointing** in comparison to the excellent performance of this procedure for FIR filter.

The reason is due to the ***phase response*** - two sinusoids will exactly **cancel** only if they have the **same magnitude *and* phase**.

1. Spectral inversion works well for a **FIR** filter because it has a ***zero phase*** - the filter kernels have a left-right symmetry.
2. Since **recursive** filters are plagued with **phase shift**, spectral inversion for IIR generally produces unsatisfactory filters.

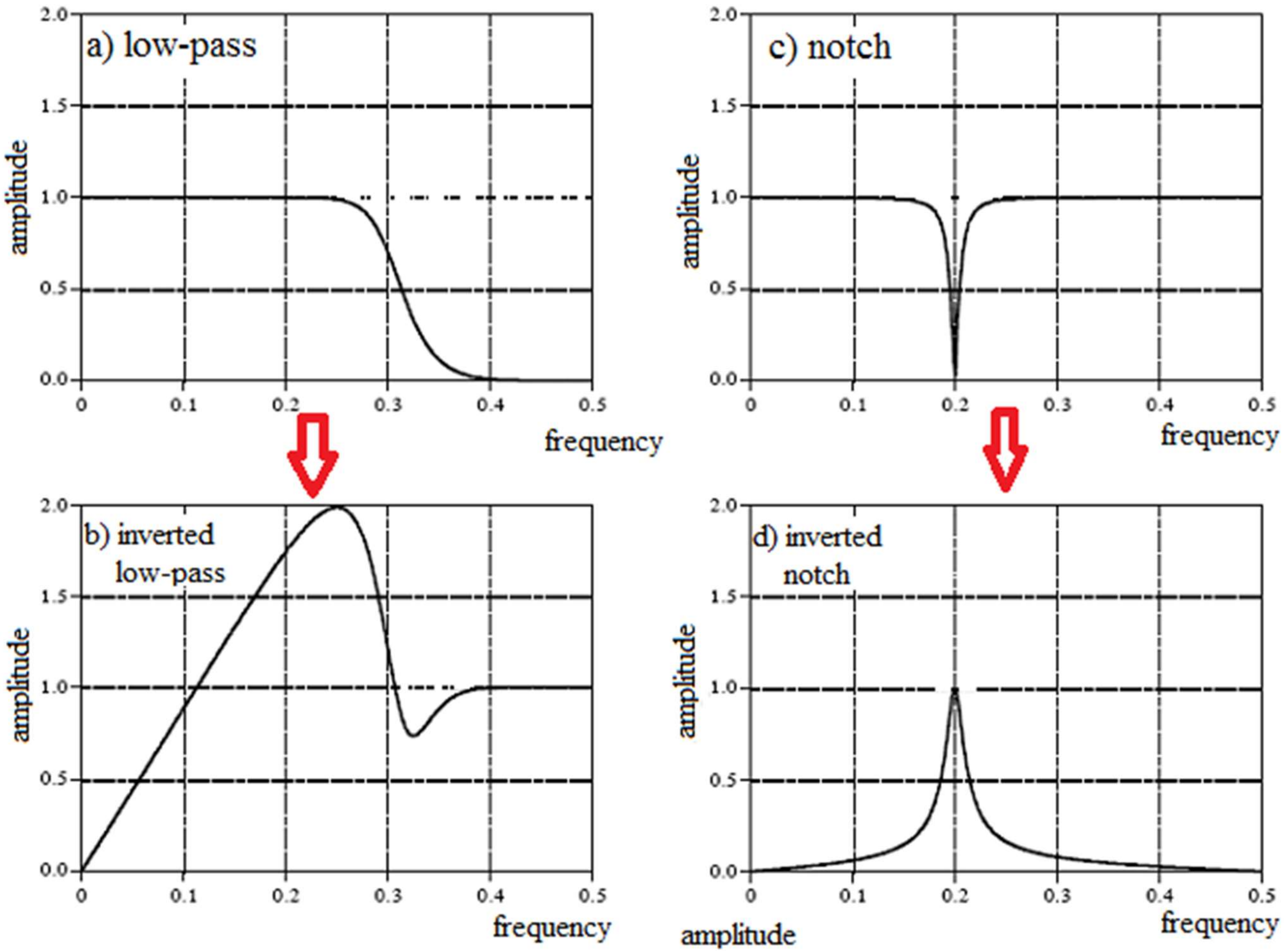


Fig. 7 Examples of spectral inversion. (a) the frequency response of a 6 pole low-pass Butterworth filter. (b) the corresponding high-pass filter obtained by spectral inversion. In (c) and (d) a notch filter is transformed into a band-pass frequency response (a more successful case).

6) Gain Changes

How to modify the recursion coefficients such that the output signal is changed in amplitude (for example to have **unity gain** in the pass-band)?

Answer:

Multiply the "*a*" coefficients by whatever factor we want the gain to change by and leave the "*b*" coefficients unchanged.

Remark:

Low-pass filters have their gain measured at a frequency of **zero**, while high-pass filters at frequency **of 0.5**, the maximum frequency allowed.

Exercises

Task 12-5

Modify the recursion coefficients of a low-pass filter such that the output signal is changed to have **unity gain** in the pass-band?

Task 12-6

Modify the recursion coefficients of a high-pass filter such that the output signal is changed to have **unity gain** in the pass-band?

4. Chebyshev-Butterworth filter design

A common method of designing recursive digital filters is the **Chebyshev-Butterworth method**:

1. It starts with a pole-zero diagram of an *analog* filter in the s-plane, and converts it into the desired *digital* filter through several mathematical *transforms*.
2. The filter is designed as a **cascade of several stages**, with each stage implementing one pair of poles.
3. The recursive coefficients for each stage are then **combined** into the recursive coefficients for the entire filter.

1) Calculate pole locations in the s-plane

Start with a Butterworth low-pass filter in the *s-plane*, with a cutoff frequency of $\omega = 1$ [rad/s].

1. Butterworth filters have poles that are **equally spaced around a circle** in the s-plane.
2. Since the filter is low-pass, **no zeros** are used.
3. The **radius of the circle is *one***, corresponding to the cutoff frequency of $\omega=1$.

The pole location, $\sigma \pm j\omega$, is calculated from the **number of poles** N_p in the filter and the stage being worked on.

2) Warp from Circle to Ellipse

To implement a Chebyshev filter, this *circular* pattern of poles must be transformed into an *elliptical* pattern.

The **relative flatness of the ellipse** determines how much **ripple** P_R will be present in the pass-band of the filter.

If the pole location on the circle is given by, σ and ω, the corresponding location on the ellipse, σ' and ω', is given by:

$$\sigma' = \sigma \sinh(v)/\,k$$

$$\omega' = \omega \cosh(v)/\,k$$

where

$$v = \frac{\sinh^{-1}(1/\varepsilon)}{N_p}\;, \quad k = \cosh\left(\frac{1}{N_p}\cosh^{-1}\left(\frac{1}{\varepsilon}\right)\right), \quad \varepsilon = \sqrt{\left(\frac{100}{100-P_R}\right)^2 - 1}$$

These equations use hyperbolic sine and cosine functions to define the ellipse, just as ordinary sine and cosine functions operate on a circle.

The flatness of the ellipse is controlled by the variable P_R, which is equal to the percentage of ripple in the filter's pass-band.

The variables: v, ε and k are used to correct the pole locations to keep a unity cutoff frequency.

Hyperbolic functions can be calculated as:

$$\sinh(x) = \frac{e^x - e^{-x}}{2}, \quad \cosh(x) = \frac{e^x + e^{-x}}{2}$$

$$\sinh^{-1}(x) = \log_e[x + (x^2 + 1)^{1/2}],$$

$$\cosh^{-1}(x) = \log_e[x + (x^2 - 1)^{1/2}]$$

3) Continuous to discrete conversion

The most common method of converting a pole-zero pattern from the s-domain into the z-domain is the **bilinear transform**.

The bilinear transform changes $H(s)$, into $H(z)$, by the substitution:

$$\boxed{s \rightarrow \frac{2(1-z^{-1})}{T(1+z^{-1})}}$$

That is, we write an equation for $H(s)$, and then replace each s with the above expression.

In most cases,

$$T = 2\tan(1/2) \approx 1.093$$

is used.

This results in the s-domain's frequency range of 0 to π radians/second, being mapped to the z-domain's frequency range of 0 to π radians / sec, and vertical lines being mapped into circles.

Example 5.

For a continuous system with a single pole-pair located at p_1 and p_2, the s-domain transfer function is given by:

$$H(s) = \frac{1}{(s-p_1)(s-p_2)}$$

The bilinear transform converts this into a discrete system by replacing each s with the above expression.

This creates a z-domain transfer function also containing two poles:

$$H(z) = \frac{1}{\left(\frac{2(1-z^{-1})}{T(1+z^{-1})} - (\sigma + j\omega)\right)\left(\frac{2(1-z^{-1})}{T(1+z^{-1})} - (\sigma - j\omega)\right)}$$

This expression can be placed in the standard form of (H1), and the recursion coefficients are identified as:

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} - b_2 z^{-2}}$$

$$a_0 = \frac{T^2}{D}$$

$$a_1 = \frac{2T^2}{D} \qquad\qquad b_1 = \frac{(8 - 2MT^2)}{D}$$

$$a_2 = \frac{T^2}{D} \qquad\qquad b_2 = \frac{(-4 - 4\sigma T - MT^2)}{D}$$

where

$$M = \sigma^2 + \omega^2,$$

$$T = 2\tan(1/2) \approx 1.093,$$

$$D = 4 - 4\sigma T + MT^2$$

So far we have calculated an **intermediate** result:

- the recursion coefficients for **one stage** of a *low-pass* **digital filter** with a **cutoff frequency** of *one*.

4) Low-pass to Low-pass Frequency Change

If we know the transfer function of a recursive low-pass filter with a unity cutoff frequency then the transfer function of a similar low-pass filter with a new cutoff frequency, Ω, is obtained by using a **low-pass to low-pass transform**.

This is carried out by *substituting variables*, just as with the bilinear transform. We start by writing the transfer function of the unity cutoff filter, and then replace each z^{-1} with the following:

$$\boxed{z^{-1} \rightarrow \frac{z^{-1} - k}{1 - kz^{-1}}}$$

where

$$k = \frac{\sin(1/2 - \Omega/2)}{\sin(1/2 + \Omega/2)}$$

This provides the transfer function of the filter with the new **cutoff frequency Ω**.

Example 6.

The following design equations result from applying this substitution to the biquad, i.e., a filter having no more than two poles and two zeros:

$$a_0' = \frac{a_0 - a_1 k + a_2 k^2}{D}$$

$$a_1' = \frac{-2a_0 k + a_1 + a_1 k^2 - 2a_2 k}{D} \qquad b_1' = \frac{2k + b_1 + b_1 k^2 - 2b_2 k}{D}$$

$$a_2' = \frac{a_0 k^2 - a_1 k + a_2}{D} \qquad b_2' = \frac{-k^2 - b_1 k + b_2}{D}$$

where

$$D = 1 + b_1 k - b_2 k^2, \quad k = \frac{\sin(1/2 - \Omega/2)}{\sin(1/2 + \Omega/2)}$$

Exercises

Task 12-7

Starting with a Butterworth low-pass filter (with two poles) in the *s-plane*, with a cutoff frequency of $\omega_1 = 1$ [rad/s], obtain the recursion coefficients of an IIR version of the Butterworth low-pass filter with cutoff-frequency $\omega_C = 2\pi \cdot 0.25 = 0.5\,\pi \cdot$[rad/s] (note: $f_C = 0.25$).

5) Low-pass to High-pass Frequency Change

The above transform can be modified to change the response of the system from low-pass to high-pass while simultaneously changing the cutoff frequency. This is accomplished by using a **low-pass to high-pass transform**, via the substitution:

$$\boxed{z^{-1} \to \frac{-z^{-1} - k}{1 + kz^{-1}}}$$

where

$$k = -\frac{\cos(\Omega/2 + 1/2)}{\cos(\Omega/2 - 1/2)}$$

As before, this can be reduced to the design of equations for changing the coefficients of a biquad stage.

These equations are identical to those of low-pass-to-low-pass transform (example 5) with only two minor changes:

- The value of k is different as shown above, and
- two coefficients, a_1 and b_1, are negated in value.

$$a_0' = \frac{a_0 - a_1 k + a_2 k^2}{D}$$

$$a_1' = -\frac{-2a_0 k + a_1 + a_1 k^2 - 2a_2 k}{D} \qquad b_1' = -\frac{2k + b_1 + b_1 k^2 - 2b_2 k}{D}$$

$$a_2' = \frac{a_0 k^2 - a_1 k + a_2}{D} \qquad b_2' = \frac{-k^2 - b_1 k + b_2}{D}$$

where

$$D = 1 + b_1 k - b_2 k^2,$$

$$k = -\frac{\cos(\Omega/2 + 1/2)}{\cos(\Omega/2 - 1/2)}$$