Transformasi Z dan Implementasinya pada sistem

Bahan Ajar Sinyal dan Sistem

Pascasarjana Terapan

P E N S

374

The Z-Transform

- In mathematics and signal processing, the **Z-transform** converts a discrete-time signal, which is a sequence of real or complex number, into a complex frequency domain representation.
- Convolution of discrete-time signals simply becomes multiplication of their z-transforms.

375

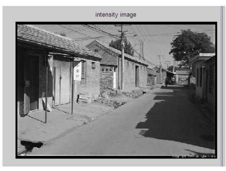
The Z-Transform

$$
\begin{aligned}
\mathcal{Z}\{x_1(n) * x_2(n)\} &= \mathcal{Z}\left\{\sum_{l=-\infty}^{\infty} x_1(l)x_2(n-l)\right\} \\
&= \sum_{n=-\infty}^{\infty}\left[\sum_{l=-\infty}^{\infty} x_1(l)x_2(n-l)\right] z^{-n} \\
&= \sum_{l=-\infty}^{\infty} x_1(l)\left[\sum_{n=-\infty}^{\infty} x_2(n-l)z^{-n}\right] \\
&= \left[\sum_{l=-\infty}^{\infty} x_1(l)z^{-l}\right]\left[\sum_{n=-\infty}^{\infty} x_2(n)z^{-n}\right] \\
&= X_1(z)X_2(z)
\end{aligned}
$$

376

The Z-Transform

Convolution

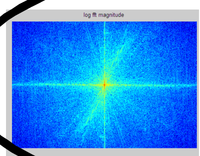


*

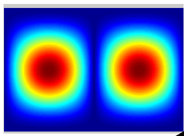
FFT

FFT

Multiplication



×



Frequency

Slide: Hoiem 377

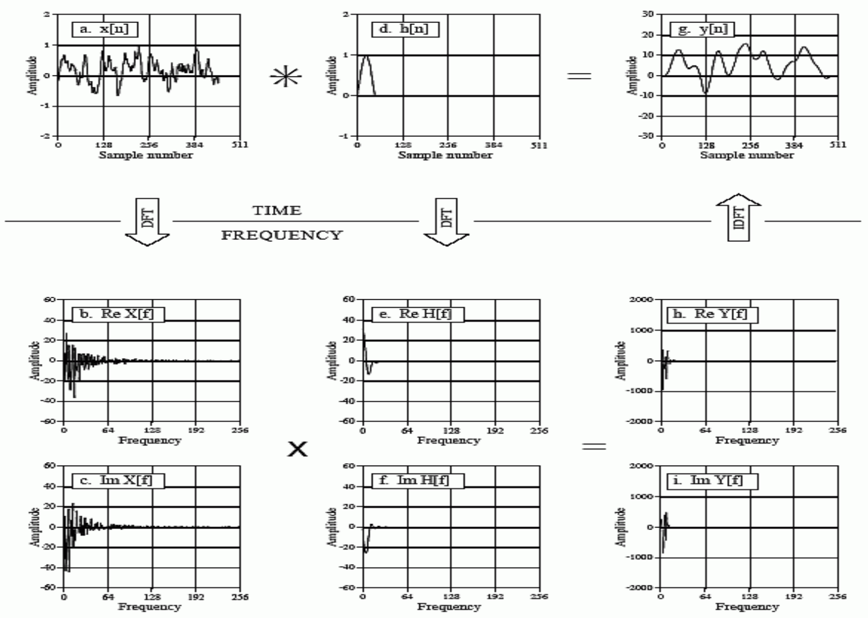


FIGURE 9-8
Frequency domain convolution. In the *time domain*, $x[n]$ is convolved with $h[n]$ resulting in $y[n]$, as is shown in Figs. (a), (d), and (g). This same procedure to be accomplished in the *frequency domain*. The DFT is used to find the frequency spectrum of the input signal, (b) & (c), and the system's frequency response, (e) & (f). Multiplying these two frequency domain signals results in the frequency spectrum of the output signal, (h) & (i). The Inverse DFT is then used to find the output signal, (g).

378

The Z-Transform

- **Analog filters** are designed using the **Laplace transform**, **Recursive digital filters** are developed with a technique called the **z-transform.**
- The Laplace transform deals with differential equations, the s-domain, and the s-plane. Correspondingly, the z-transform deals with difference equations, the z-domain, and the z-plane.
- The **s-plane** is arranged in a **rectangular coordinate** system, while the z-plane uses a polar format.

379

The Z-Transform

- Recursive digital filters are often designed by starting with one of the classic analog filters, such as the Butterworth, Chebyshev, or elliptic. A series of mathematical conversions are then used to obtain the desired digital filter. The z-transform provides the framework for this mathematics.

380

From Laplace Transform to Z-Transform

$$X(s) = \int_{t=-\infty}^{\infty} x(t)\, e^{-st}\, dt \quad \Rightarrow \quad X(\sigma, \omega) = \int_{t=-\infty}^{\infty} x(t)\, e^{-\sigma t} e^{-j\omega t}\, dt$$

Continuous

Exponential Forms :

$$X(\sigma, \omega) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-\sigma n} e^{-j\omega n}$$

Discrete

$$y[n] = e^{-\sigma n} \quad \text{or} \quad y[n] = r^{-n}$$

$$\Downarrow$$

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

where: $\sigma = ln(r)$

381

From Laplace Transform to Z-Transform

a. Decreasing

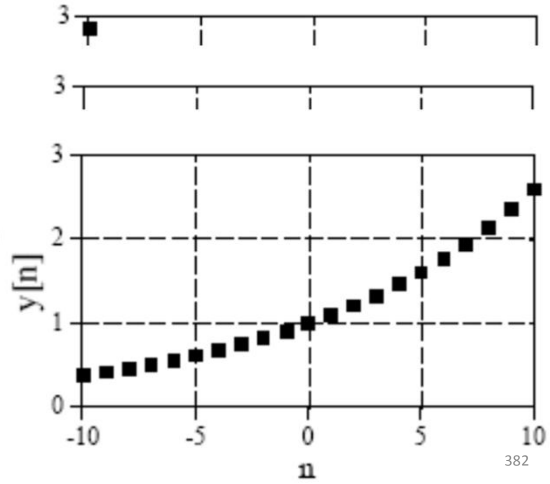
b. Constant

c. Increasing

$$y[n] = e^{-\sigma n},\ \sigma = -0.095$$

or

$$y[n] = r^{-n},\ \ r = 0.9$$



382

From Laplace Transform to Z-Transform

$$X(\sigma,\omega) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-\sigma n} e^{-j\omega n}$$

$$z = re^{j\omega}$$

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

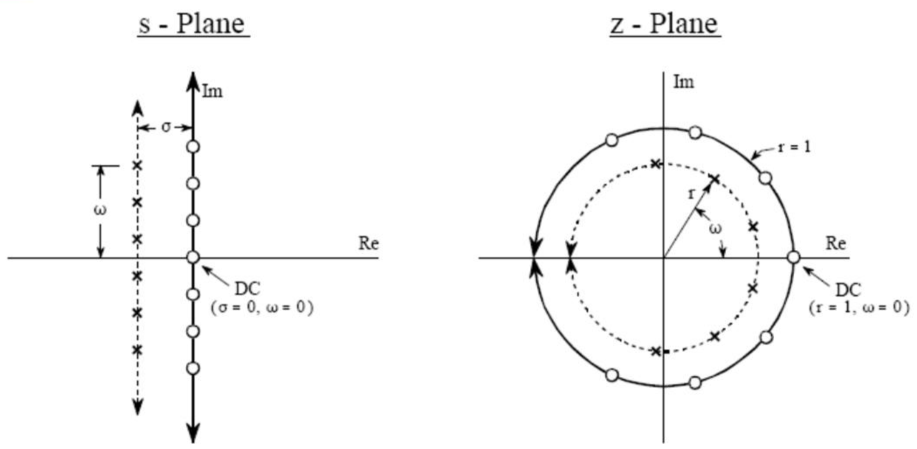
383

The Z-Transform

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

$$z = re^{j\omega}$$

The Relationship between the s-plane and the z-plane



The Analysis of Recursive Systems

A recursive filter is described by a difference equation:

$$y[n] = a_0x[n] + a_1x[n-1] + a_2x[n-2] + \cdots + b_1y[n-1] + b_2y[n-2] + b_3y[n-3] + \cdots$$

The "*a*" and "*b*" terms are the recursion coefficients.

The **system's transfer function, H[z] = Y[z] / X[z]**

$$H[z] = \frac{a_0 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3} + \cdots}{1 - b_1z^{-1} - b_2z^{-2} - b_3z^{-3} - \cdots}$$

386

The recursion coefficients of a Digital Filter

$a_0 = 0.389$	
$a_1 = -1.558$	$b_1 = 2.161$
$a_2 = 2.338$	$b_2 = -2.033$
$a_3 = -1.558$	$b_3 = 0.878$
$a_4 = 0.389$	$b_4 = -0.161$

The system's transfer function

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

387

Transfer function in pole-zero form

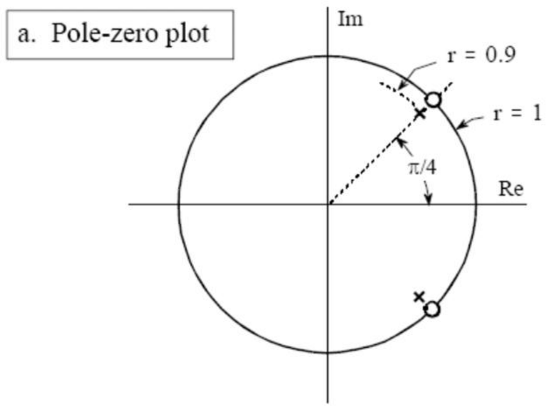
Just as with the s-domain, an important feature of the z-domain is that the transfer function can be expressed as **poles** and **zeros**. This provides the second general form of the z-domain:

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

Each of the poles (p) and zeros (z) is a complex number.

388

A Notch filter Design



(1) specify the pole-zero placement in the z-plane

389

A Notch filter Design

In polar form:

$$z_1 = 1.00e^{j(\pi/4)}$$
$$z_2 = 1.00e^{j(-\pi/4)}$$
$$p_1 = 0.90e^{j(\pi/4)}$$
$$p_2 = 0.90e^{j(-\pi/4)}$$

In rectangular form:

$$z_1 = 0.7071 + j\,0.7071$$
$$z_2 = 0.7071 - j\,0.7071$$
$$p_1 = 0.6364 + j\,0.6364$$
$$p_2 = 0.6364 - j\,0.6364$$

(2) Write down the transfer function in pole – zero form

$$H(z) = \frac{[z-(0.7071+j\,0.7071)]\,[z-(0.7071-j\,0.7071)]}{[z-(0.6364+j\,0.6364)]\,[z-(0.6364-j\,0.6364)]}$$

390

A Notch filter Design

(3) Rearrange the transfer function into the form of Z Polynomial

$$H(z) = \frac{z^2 - 0.7071z + j0.7071z - 0.7071z + 0.7071^2 - j0.7071^2 - j0.7071z + j0.7071^2 - j^2 0.7071^2}{z^2 - 0.6364z + j0.6364z - 0.6364z + 0.6364^2 - j0.6364^2 - j0.6364z + j0.6364^2 - j^2 0.6364^2}$$

$$H[z] = \frac{1.000 - 1.414z + 1.000z^2}{0.810 - 1.273z + 1.000z^2}$$

$$H[z] = \frac{1.000 - 1.414z^{-1} + 1.000z^{-2}}{1.000 - 1.273z^{-1} + 0.810z^{-2}}$$

391

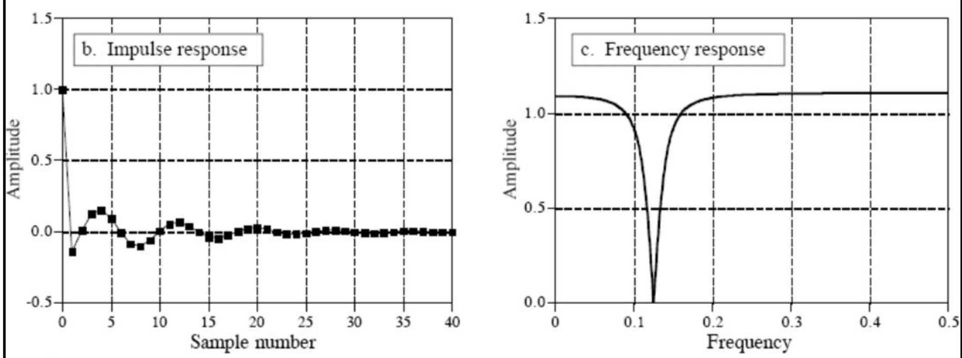
A Notch filter Design

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

$$a_0 = 1.000$$
$$a_1 = -1.414 \qquad b_1 = 1.273$$
$$a_2 = 1.000 \qquad b_2 = -0.810$$

392

The resulting impulse and frequency response of a Notch Filter.



393

How do we find the frequency response?

- ✓ Mathematical

  [**Find H(z) at r=1,** replace each z with **e** $^{j\pi}$ $\rightarrow$ $H(\omega)$]

- ✓ Computational (programming)

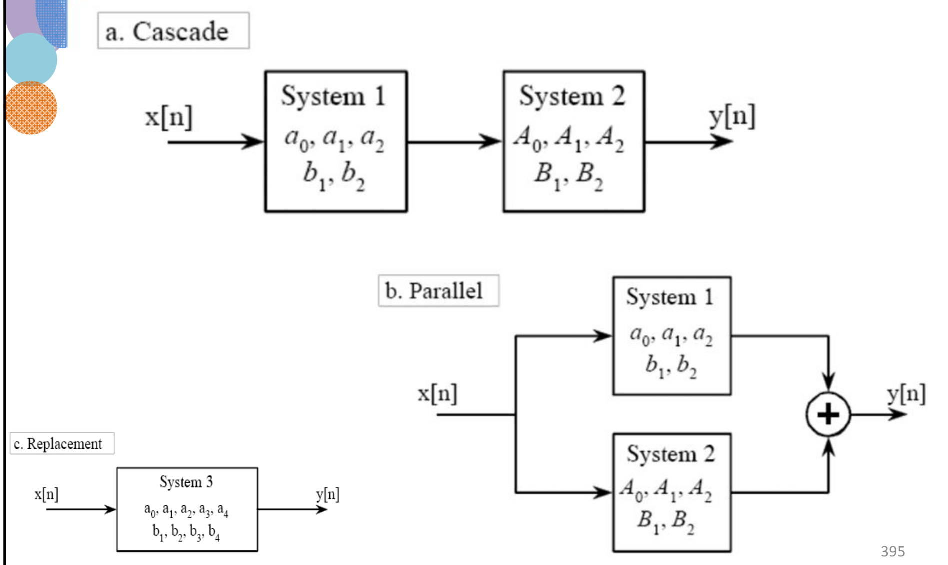
  [**Define x equally spaced frequencies between ω = 0 and ω =** π]

- ✓ Find the frequency response from the recursion coefficients that are actually used to implement the filter.

  [Find the impulse response of the filter by passing an impulse through the system, then take the FFT of the impulse response to find the system's frequency response.]

394

Cascade and Parallel Stages



395

Cascade and Parallel Stages

$$H[z] = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} - b_2 z^{-2}} \times \frac{A_0 + A_1 z^{-1} + A_2 z^{-2}}{1 - B_1 z^{-1} - B_2 z^{-2}}$$

$$H[z] = \frac{a_0 A_0 + (a_0 A_1 + a_1 A_0) z^{-1} + (a_0 A_2 + a_1 A_1 + a_2 A_0) z^{-2} + (a_1 A_2 + a_2 A_1) z^{-3} + (a_2 A_2) z^{-4}}{1 - (b_1 + B_1) z^{-1} - (b_2 + B_2 - b_1 B_1) z^{-2} - (-b_1 B_2 - b_2 B_1) z^{-3} - (-b_2 B_2) z^{-4}}$$

$$\begin{aligned} \mathrm{a}_0 &= a_0 A_0 \\ \mathrm{a}_1 &= a_0 A_1 + a_1 A_0 \\ \mathrm{a}_2 &= a_0 A_2 + a_1 A_1 + a_2 A_0 \\ \mathrm{a}_3 &= a_1 A_2 + a_2 A_1 \\ \mathrm{a}_4 &= a_2 A_2 \end{aligned}$$

$$\begin{aligned} \mathrm{b}_1 &= b_1 + B_1 \\ \mathrm{b}_2 &= b_2 + B_2 - b_1 B_1 \\ \mathrm{b}_3 &= -b_1 B_2 - b_2 B_1 \\ \mathrm{b}_4 &= -b_2 B_2 \end{aligned}$$

396

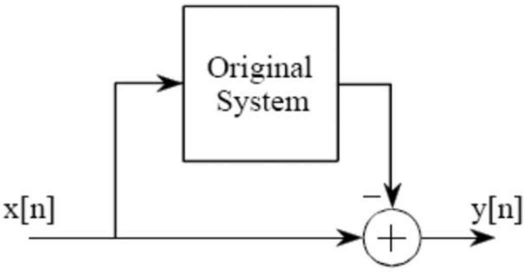
In specific cases, it is possible to derive simpler equations directly relating the pole-zero positions to the recursion coefficients. For example, a system containing two poles and two zeros, called as **biquad**, has the following relations:

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

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

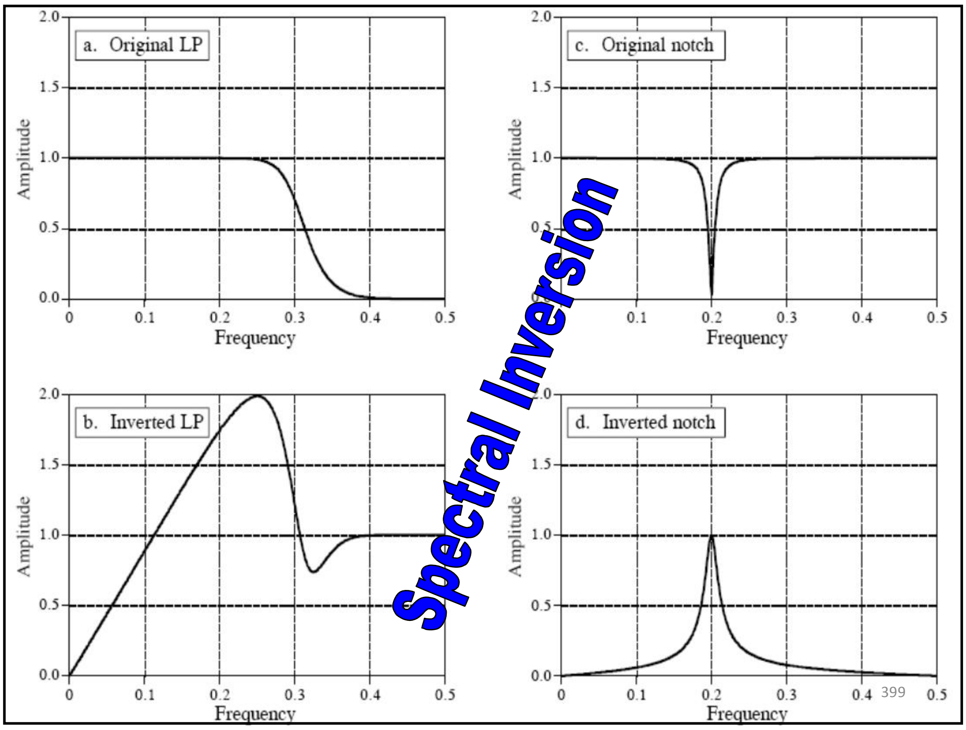
397

Spectral Inversion



$$\begin{aligned}
\mathrm{a}_0 &= 1 - a_0 \\
\mathrm{a}_1 &= -a_1 - b_1 \\
\mathrm{a}_2 &= -a_2 - b_2 \\
\mathrm{a}_3 &= -a_3 - b_3 \\
&\vdots
\end{aligned}$$

398



399

The Hilbert transformer.

Any system that has the frequency response: Magnitude = 1 and phase = 90 degrees, for all frequencies.

Hilbert transformers can be analog or discrete (hardware or software)

Gain Changes

Since the gain must be specified at a frequency in the *passband*, the procedure depends on the type of filter being used. Low-pass filters have their gain measured at a frequency of *zero*, while high-pass filters use a frequency of 0.5, the maximum frequency allowable.

$$y[n] = a_0 x[n] + a_1 x[n-1] + a_2 x[n-2] + \cdots + b_1 y[n-1] + b_2 y[n-2] + b_3 y[n-3] + \cdots$$

$$G = a_0 + a_1 + a_2 + a_3 + \cdots + b_1 G + b_2 G + b_3 G + b_4 G \cdots$$

$$G = \frac{a_0 + a_1 + a_2 + a_3 \cdots}{1 - (b_1 + b_2 + b_3 \cdots)}$$

To make a filter have a gain of *one* at DC, calculate the existing gain by using this relation, and then divide all the "a" coefficients by *G*.

Gain Changes

The gain at a frequency of 0.5 is found in a similar way: we force the input and output signals to operate at this frequency, and see how the system responds. At a frequency of 0.5, the samples in the input signal alternate between -1 and 1. That is, successive samples are: 1, -1, 1, -1, 1, -1, 1, etc. The corresponding output signal also alternates in sign, with an amplitude equal to the gain of the system: *G*, -*G*, *G*, -*G*, *G*, -*G*, etc. Plugging these signals into the recursion equation:

$$G = a_0 - a_1 + a_2 - a_3 + \cdots - b_1 G + b_2 G - b_3 G + b_4 G \cdots$$

$$G = \frac{a_0 - a_1 + a_2 - a_3 + a_4 \cdots}{1 - (-b_1 + b_2 - b_3 + b_4 \cdots)}$$