Lecture 16 z-Transform

Fundamentals of Digital Signal Processing

Spring, 2012

Wei-Ta Chu

2012/5/1

Definition of the z-Transform

- A finite-length signal

  $x[n] = \sum_{k=0}^{N} x[k]\delta[n-k]$

- The *z*-transform of such a signal is defined by

  $X(z) = \sum_{k=0}^{N} x[k]z^{-k}$

- Assume that *z* represents any complex number, i.e., *z* is the independent (complex) variable of the *z*-transform *X*(z).
- It's more instructive to write in the form

  $X(z) = \sum_{k=0}^{N} x[k](z^{-1})^{k}$

- *X*(*z*) is simply a polynomial of degree *N* in the variable z^{-1}.

Definition of the z-Transform

- All that we have to do to obtain $X(z)$ is to construct a polynomial whose coefficients are the values of the sequence $x[n]$. The k^{th} sequence value is the coefficient of the k^{th} power of z^{-1} in the polynomial $X(z)$.
- Taking z-transform and inverse z-transform

n-Domain (time-Domain)	$\overset{z}{\longleftrightarrow}$	*z-Domain*
$x[n] = \sum_{k=0}^{N} x[k]\delta[n-k]$	$\overset{z}{\longleftrightarrow}$	$X(z) = \sum_{k=0}^{N} x[k]z^{-k}$

- z-transform pair

$$x[n] \overset{z}{\longleftrightarrow} X(z)$$

Example

- Suppose $x[n] = \delta[n - n_0]$

n-Domain	$\overset{z}{\longleftrightarrow}$	*z-Domain*
$x[n] = \delta[n - n_0]$	$\overset{z}{\longleftrightarrow}$	$X(z) = z^{-n_0}$

- Consider the sequence *x*[*n*]

n	**<-1**	**-1**	**0**	**1**	**2**	**3**	**4**	**5**	**>5**
x[*n*]	0	0	2	4	6	4	2	0	0

$$X(z) = 2 + 4z^{-1} + 6z^{-2} + 4z^{-3} + 2z^{-4}$$

Example

- Inverse *z*-Transform
- Consider the *X*(*z*) by the equation

$$X(z) = 1 - 2z^{-1} + 3z^{-3} - z^{-5}$$

$$x[n] = \begin{cases} 0 & n < 0 \\ 1 & n = 0 \\ -2 & n = 1 \\ 0 & n = 2 \\ 3 & n = 3 \\ 0 & n = 4 \\ -1 & n = 5 \\ 0 & n > 5 \end{cases}$$

$$x[n] = \delta[n] - 2\delta[n-1] + 3\delta[n-3] - \delta[n-5]$$

The z-Transform and Linear Systems

- Recall that the general difference equation of an FIR filter is $y[n] = \sum_{k=0}^{M} b_k x[n-k]$
- In convolution sum $y[n] = x[n] * h[n]$
- Remember that the impulse response $h[n]$ is identical to the sequence of difference equation coefficients b_n

n	**<0**	**0**	**1**	**2**	**…**	**M**	**>M**
$h[n]$	0	b_0	b_1	b_2	…	b_M	0

$$h[n] = \sum_{k=0}^{M} b_k \delta[n-k]$$

The z-Transform and Linear Systems

- Let the input to the FIR system be the signal

  $x[n] = z^n \qquad \text{for all } n$

  where *z* is any complex number.
- The corresponding output signal is

$$y[n] = \sum_{k=0}^{M} b_k x[n-k]$$
$$= \sum_{k=0}^{M} b_k z^{n-k}$$
$$= \sum_{k=0}^{M} b_k z^n z^{-k} = \left(\sum_{k=0}^{M} b_k z^{-k}\right) z^n$$

- The term inside the parentheses is a polynomial in z^{-1} whose form depends on the coefficients of the FIR filter. It's called the *system function* of the FIR filter.

The z-Transform and Linear Systems

- The system function is the *z*-transform of the impulse-response sequence.
- We define the *system function* of an FIR filter to be

$$H(z) = \sum_{k=0}^{M} b_k z^{-k} = \sum_{k=0}^{M} h[k] z^{-k}$$

The system function *H*(*z*) is the z-transform of the impulse response.

$$h[n] = \sum_{k=0}^{M} b_k \delta[n-k] \overset{z}{\longleftrightarrow} H(z) = \sum_{k=0}^{M} b_k z^{-k}$$

The z-Transform and Linear Systems

- For FIR filters, if the input is z^n for $-\infty < n < \infty$, then the corresponding output is $y[n] = h[n] * z^n = H(z)z^n$
- That is, the result of convolving the sequence $h[n]$ with the sequence z^n is $H(z)z^n$, where $H(z)$ is the *z*-transform of $h[n]$
- The FIR filter difference equation can be transformed easily into a polynomial in the *z*-domain simply by replacing each "delayed by *k*" (i.e., $x[n-k]$) by z^{-k}

Example

- Consider the FIR filter
  $y[n] = 6x[n] - 5x[n-1] + x[n-2]$
- The *z*-transform system function is
  $H(z) = 6 - 5z^{-1} + z^{-2}$
  $= (3 - z^{-1})(2 - z^{-1}) = 6\frac{(z - \frac{1}{3})(z - \frac{1}{2})}{z^2}$
- The zeros of *H*(*z*) are 1/3 and 1/2
- The filter $w[n] = x[n] - \frac{5}{6}x[n-1] + \frac{1}{6}x[n-2]$ has a system function with the same zeros, but the overall constant is 1 rather than 6. This simply means that
  $$w[n] = y[n]/6$$

Superposition Properties

- The *z*-transform is a linear transformation.
- Consider the sequence $x[n] = ax_1[n] + bx_2[n]$

$$X(z) = \sum_{n=0}^{N}(ax_1[n] + bx_2[n])z^{-n}$$

$$= a\sum_{n=0}^{N} x_1[n]z^{-n} + b\sum_{n=0}^{N} x_2[n]z^{-n}$$ superposition property

$$= aX_1(z) + bX_2(z)$$

The z-transform is linear

$$ax_1[n] + bx_2[n] \overset{z}{\longleftrightarrow} aX_1(z) + bX_2(z)$$

Example

- Recall that any finite-length sequence *x*[*n*] can be represented as a sum of scaled and shifted impulse sequences

  $x[n] = \sum_{k=0}^{N} x[k]\delta[n-k]$

- For a single shifted unit impulse sequence

  $\delta[n-k] \overset{z}{\longleftrightarrow} z^{-k}$

- Because $ax_1[n] + bx_2[n] \overset{z}{\longleftrightarrow} aX_1(z) + bX_2(z)$ we obtain $X(z) = \sum_{k=0}^{N} x[k]z^{-k}$

Time-Delay Property

- The quantity z^{-1} in the z-domain corresponds to a time shift of 1 in the n-domain.
- Consider the length-6 signal $x[n]$

n	<0	0	1	2	3	4	5	>5
$x[n]$	0	3	1	4	1	5	9	0

- The z-transform of $x[n]$

$$X(z) = 3 + z^{-1} + 4z^{-2} + z^{-3} + 5z^{-4} + 9z^{-5}$$

- E.g. the term $4z^{-2}$ indicates that the signal values at n = 2 is $x[2] = 4$.

Time-Delay Property

- Now consider the effect of multiplying the polynomial by z^{-1}

$$Y(z) = z^{-1}X(z)$$
$$= z^{-1}(3 + z^{-1} + 4z^{-2} + z^{-3} + 5z^{-4} + 9z^{-5})$$
$$= 0z^{0} + 3z^{-1} + z^{-2} + 4z^{-3} + z^{-4} + 5z^{-5} + 9z^{-6}$$

- The resulting polynomial $Y(z)$ is the z-transform representation of a signal $y[n]$, which is found by using the polynomial coefficients and exponents in $Y(z)$ to determine the values of $y[n]$ at all time positions.
- The result of $y[n]$:

n	**<0**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**>6**
$y[n]$	0	0	3	1	4	1	5	9	0

Time-Delay Property

- For any finite-length sequence, multiplication of the *z*-transform polynomial by z^{-1} simply subtracts one from each exponent in the polynomial, thereby creating a delay of one.

A delay of one sample multiplies the z-transform by z^{-1}

$$x[n-1] \stackrel{z}{\longleftrightarrow} z^{-1}X(z)$$

A delay of n_0 sample multiplies the z-transform by z^{-n0}

$$x[n-n_0] \stackrel{z}{\longleftrightarrow} z^{-n_0}X(z)$$

A General z-Transform Formula

- Our definition assumes that the sequence is nonzero only in the interval $0 \leq n \leq N$.

$$X(z) = \sum_{n=0}^{N} x[n]z^{-n}$$

- It's possible to extend the definition

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

Unit-Delay Operator

- In the time domain, the unit-delay operator $\mathcal{D}$ is defined by $y[n] = \mathcal{D}\{x[n]\} = x[n-1]$
- It's instructive to find the *z*-transform representation of this system by letting the input to the unit-delay system be the signal $x[n] = z^n$ for all n
- With the z^n input signal, the output of the unit delay is
  $y[n] = \mathcal{D}\{x[n]\}$
  $= \mathcal{D}\{z^n\} = z^{n-1} = z^{-1}z^n = z^{-1}x[n]$
- The input signal is multiplied by z^{-1}, in particular when
  $x[n] = z^n$

Unit-Delay Operator

- We must remember that it holds only for $x[n] = z^n$
- It's common to use z^{-1} interchangeably with the unit-delay operator symbol $\mathcal{D}$, so we can say that for any input $x[n]$ the action of the unit-delay system is represented by the operator z^{-1}

$$y[n] = z^{-1}\{x[n]\} = x[n-1]$$

- If $y[n] = x[n-1]$, then $Y(z) = z^{-1}X(z)$. For any finite-length sequence, z^{-1} multiplies $X(z)$ to produce $Y(z)$ – precise way to represent a unit delay
  - It's not appropriate to write $z^{-1}x[n]$ without the brackets, because this mixes the z-domain and the n-domain

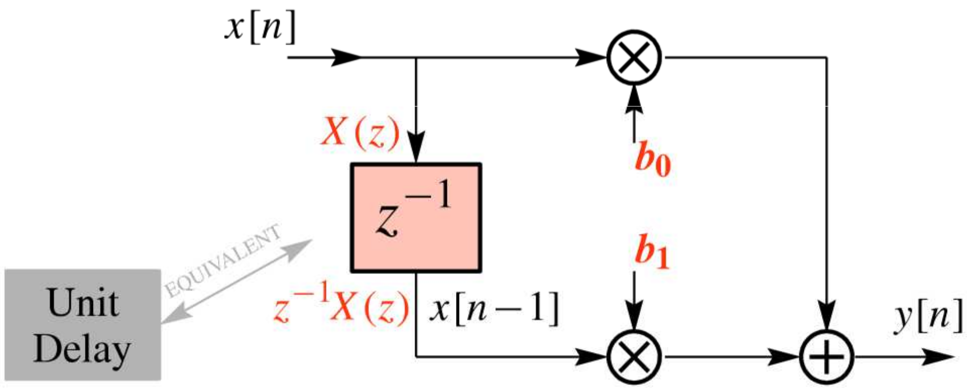
Operator Notation

- Consider a system that calculates the first difference of two successive signal values: $y[n] = x[n] - x[n-1]$
- The z-transform operator that represents the first difference system is $(1-z^{-1})$
  $$y[n] = (1 - z^{-1})\{x[n]\} = x[n] - x[n-1]$$
- The operator "1" leaves $x[n]$ unchanged, and the operator z^{-1} delays $x[n]$ before subtracting it from $x[n]$
- Another example is the system that delays by n_d samples
  $$y[n] = x[n - n_d] \qquad n_d \text{ is an integer}$$
- In this case, the system function is $H(z) = z^{-n_d}$ and the operator is z^{-n_d}

Operator Notation in Block Diagrams

- All the unit delays become z^{-1} operators in the transform domain, and, owing to the superposition property of the *z*-transform, the scalar multipliers and adders are the same as in the time-domain representation.

$$y[n] = b_0 x[n] + b_1 x[n-1]$$



McClellan, Schafer and Yoder, *Signal Processing First*, ISBN 0-13-065562-7.
Pearson Prentice Hall, Inc. Upper Saddle River, NJ 07458. © 2003

Convolution and the z-Transform

- A unit delay of a signal in the *n*-domain is equivalent to multiplication by z^{-1} of the corresponding *z*-transform in the *z*-domain.
- The impulse response of the unit-delay system is

$$h[n] = \delta[n-1]$$

- A delay by one sample is equivalent to the convolution

$$y[n] = x[n] * \delta[n-1] = x[n-1]$$

- The system function of the unit-delay system is the *z*-transform of its impulse response so

$$H(z) = z^{-1}$$

Convolution and the z-Transform

- The unit-delay property states that delay by one sample multiples the z-transform by z^{-1}

  $$Y(z) = z^{-1}X(z)$$

- z-transform of the output is equal to the z-transform of the input multiplied by the system function of the LTI system $Y(z) = H(z)X(z)$

Convolution and the z-Transform

- Recall that the discrete convolution

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

- Apply the superposition property and the general delay property to find the *z*-transform of *y*[*n*]

$$Y(z) = \sum_{k=0}^{M} h[k](z^{-k}X(z))$$

$$= \left(\sum_{k=0}^{M} h[k]z^{-k}\right)X(z) = H(z)X(z)$$

- If *x*[*n*] is a finite-length sequence, *X*(*z*) is a polynomial, so this equation proves that convolution is equivalent to polynomial multiplication.

Example

$$x[n] = \delta[n-1] - \delta[n-2] + \delta[n-3] - \delta[n-4]$$
$$h[n] = \delta[n] + 2\delta[n-1] + 3\delta[n-2] + 4\delta[n-3]$$

- The *z*-transforms of the sequences are:

$$X(z) = 0 + 1z^{-1} - 1z^{-2} + 1z^{-3} - 1z^{-4}$$
$$H(z) = 1 + 2z^{-1} + 3z^{-2} + 4z^{-3}$$

- Both $X(z)$ and $H(z)$ are polynomials in z^{-1}, so we can compute the *z*-transform of the convolution by multiplying these two polynomials

$$Y(z) = H(z)X(z) =$$
$$(0 + 1z^{-1} - 1z^{-2} + 1z^{-3} - 1z^{-4})(1 + 2z^{-1} + 3z^{-2} + 4z^{-3})$$
$$= z^{-1} + z^{-2} + 2z^{-3} + 2z^{-4} - 3z^{-5} + z^{-6} - 4z^{-7}$$

Example

$$Y(z) = H(z)X(z) =$$
$$(0 + 1z^{-1} - 1z^{-2} + 1z^{-3} - 1z^{-4})(1 + 2z^{-1} + 3z^{-2} + 4z^{-3})$$
$$= z^{-1} + z^{-2} + 2z^{-3} + 2z^{-4} - 3z^{-5} + z^{-6} - 4z^{-7}$$

- We can inverse transform *Y*(*z*) to obtain

$$y[n] = \delta[n-1] + \delta[n-2] + 2\delta[n-3] + 2\delta[n-4] - 3\delta[n-5] + \delta[n-6] - 4\delta[n-7]$$

- A few terms about convolution:

$$y[0] = h[0]x[0] = 1(0) = 0$$
$$y[1] = h[0]x[1] + h[1]x[0] = 1(1) + 2(0) = 1$$
$$y[2] = h[0]x[2] + h[1]x[1] + h[2]x[0] = 1(-1) + 2(1) + 3(0) = 1$$
$$y[3] = h[0]x[3] + h[1]x[2] + h[2]x[1] + h[3]x[0] = 1(1) + 2(-1) + 3(1) = 2$$

Example

- The same thing happens in polynomial multiplication because exponents add.
- The final row is the sequence of values of $y[n] = x[n]*h[n]$ or, equivalently, the coefficients of the polynomial $Y(z)$.

z	z^0	z^{-1}	z^{-2}	z^{-3}	z^{-4}	z^{-5}	z^{-6}	z^{-7}
$x[n]$, $X(z)$	0	+1	-1	+1	-1	0	0	0
$h[n]$, $H(z)$	1	2	3	4				
$X(z)$	0	+1	-1	+1	-1	0	0	0
$2z^{-1}X(z)$		0	+2	-2	+2	-2	0	0
$3z^{-2}X(z)$			0	+3	-3	+3	-3	0
$4z^{-3}X(z)$				0	+4	-4	+4	-4
$y[n]$, $Y(z)$	0	+1	+1	+2	+2	-3	+1	-4

Convolution and the z-Transform

- Convolution and polynomial multiplication are essentially the same thing.

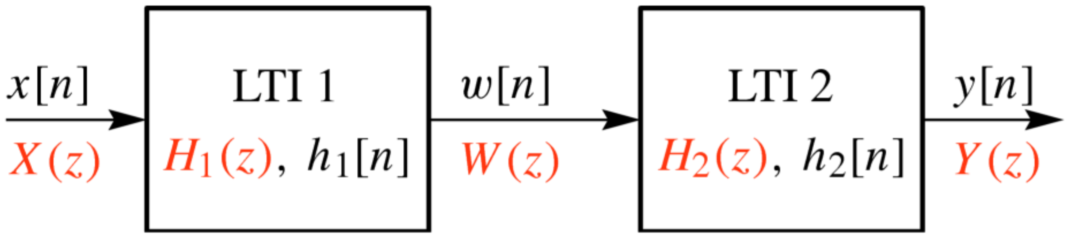
Convolution in the n-domain corresponds to multiplication in the z-domain

$$y[n] = h[n] * x[n] \quad \overset{z}{\longleftrightarrow} \quad Y(z) = H(z)X(z)$$

Cascading Systems

- One of the main applications of the z-transform in system design is its use in creating alternative filters that have exactly the same input-output behavior.
- In cascading systems, the *z*-transform of the overall impulse response is the product of the individual *z*-transforms of the two impulse responses.

$$h[n] = h_1[n] * h_2[n] \;\overset{z}{\longleftrightarrow}\; H(z) = H_1(z)H_2(z)$$

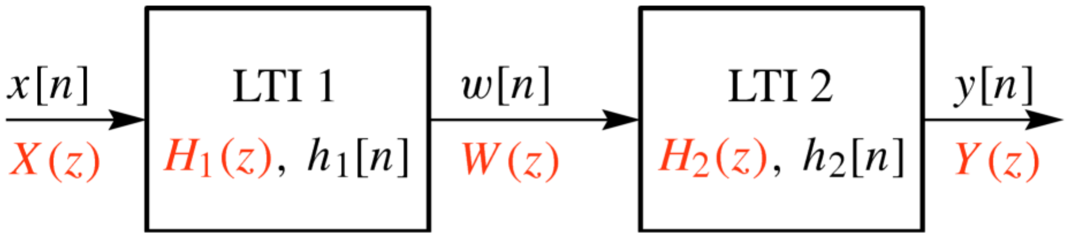


Example

$$w[n] = 3x[n] - x[n-1]$$
$$y[n] = 2w[n] - w[n-1]$$

$$y[n] = 2(3x[n] - x[n-1]) - (3x[n-1] - x[n-2])$$
$$= 6x[n] - 5x[n-1] + x[n-2]$$

- The cascade of the two first-order system is equivalent to a single second-order system



McClellan, Schafer and Yoder, *Signal Processing First*, ISBN 0-13-065562-7.
Pearson Prentice Hall, Inc. Upper Saddle River, NJ 07458. © 2003

Example

- The overall difference equation would be extremely tedious if the systems were higher-order.
- The z-transform simplifies these operations into the multiplication of polynomials.
- The first-order system functions

$$H_1(z) = 3 - z^{-1} \qquad H_2(z) = 2 - z^{-1}$$

- The overall system function is

$$H(z) = (3 - z^{-1})(2 - z^{-1}) = 6 - 5z^{-1} + z^{-2}$$

- The *z*-domain solution is more straightforward than the *n*-domain solution.