Digital Audio and Speech Processing

(Sayısal Ses ve Konuşma İşleme)

Prof. Dr. Nizamettin AYDIN

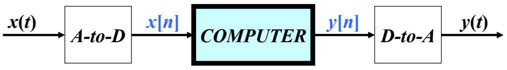
naydin@yildiz.edu.tr
nizamettinaydin@gmail.com
http://www3.yildiz.edu.tr/~naydin

FIR Filtering

LECTURE OBJECTIVES

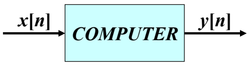
- INTRODUCE FILTERING IDEA
  - Weighted Average
  - Running Average
- FINITE IMPULSE RESPONSE FILTERS
  - **FIR** Filters
- INFINITE IMPULSE RESPONSE FILTERS
  - **IIR** Filters
- Show how to compute the output $y[n]$ from the input signal, $x[n]$

DIGITAL FILTERING



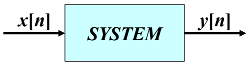
- CONCENTRATE on the COMPUTER
  - PROCESSING ALGORITHMS
  - SOFTWARE (MATLAB)
  - HARDWARE: DSP chips, VLSI
- DSP: DIGITAL SIGNAL PROCESSING

DISCRETE-TIME SYSTEM



- OPERATE on $x[n]$ to get $y[n]$
- WANT a **GENERAL** CLASS of SYSTEMS
  - **ANALYZE** the SYSTEM
    - TOOLS: TIME-DOMAIN & FREQUENCY-DOMAIN
  - **SYNTHESIZE** the SYSTEM

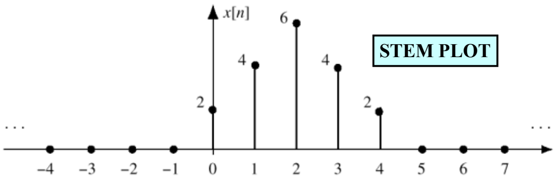
D-T SYSTEM EXAMPLES



- EXAMPLES:
  - POINTWISE OPERATORS
    - SQUARING: $y[n] = (x[n])^2$
  - RUNNING AVERAGE
    - RULE: "the output at time n is the average of three consecutive input values"

DISCRETE-TIME SIGNAL

- $x[n]$ is a LIST of NUMBERS
  - INDEXED by "n"



3-PT AVERAGE SYSTEM

- ADD 3 CONSECUTIVE NUMBERS
  - – Do this for each “*n*”

the following input–output equation

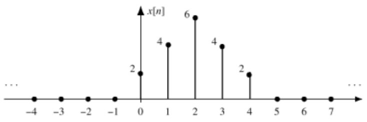
$$y[n] = \tfrac{1}{3}(x[n] + x[n+1] + x[n+2])$$

n	$n < -2$	-2	-1	0	1	2	3	4	5	$n > 5$
$x[n]$	0	0	0	2	4	**6**	**4**	**2**	0	0
$y[n]$	0	$\frac{2}{3}$	2	4	$\frac{14}{3}$	**4**	2	$\frac{2}{3}$	0	0

n=0 $y[0] = \tfrac{1}{3}(x[0] + x[1] + x[2])$

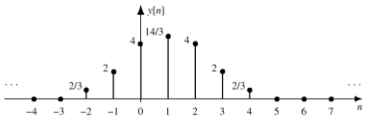
n=1 $y[1] = \tfrac{1}{3}(x[1] + x[2] + x[3])$

3-PT AVERAGE SYSTEM



INPUT SIGNAL

$y[n] = \tfrac{1}{3}(x[n] + x[n+1] + x[n+2])$



OUTPUT SIGNAL

PAST, PRESENT, FUTURE

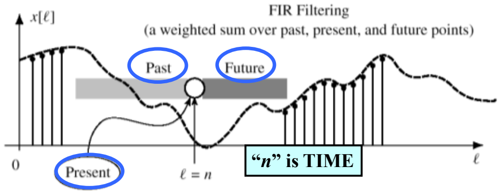


Figure 5.4 The running-average filter calculation at time index n uses values within a sliding window (shaded). Dark shading indicates the future ($\ell > n$); light shading, the past ($\ell < n$).

ANOTHER 3-pt AVERAGER

- Uses “PAST” VALUES of $x[n]$
  - – IMPORTANT IF “*n*” represents REAL TIME
    - WHEN $x[n]$ & $y[n]$ ARE STREAMS

$$y[n] = \tfrac{1}{3}(x[n] + x[n-1] + x[n-2])$$

n	$n < -2$	-2	-1	0	1	2	3	4	5	6	7	$n > 7$
$x[n]$	0	0	0	2	4	**6**	**4**	**2**	0	0	0	0
$y[n]$	0	0	0	$\frac{2}{3}$	2	4	$\frac{14}{3}$	**4**	2	$\frac{2}{3}$	0	0

GENERAL FIR FILTER

- FILTER COEFFICIENTS $\{b_k\}$
  - – DEFINE THE FILTER

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

  - – For example, $b_k = \{3, -1, 2, 1\}$

$$y[n] = \sum_{k=0}^{3} b_k x[n-k]$$
$$= 3x[n] - x[n-1] + 2x[n-2] + x[n-3]$$

GENERAL FIR FILTER

- FILTER COEFFICIENTS $\{b_k\}$

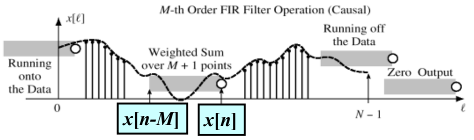
$$y[n] = \sum_{k=0}^{M} b_k x[n-k]$$

- FILTER **ORDER** is M
- FILTER **LENGTH** is $L = M+1$
  - – NUMBER of FILTER COEFFS is L

GENERAL FIR FILTER

- SLIDE a WINDOW across $x[n]$

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



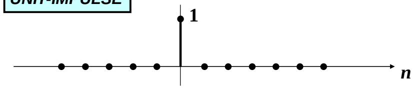
FILTERED STOCK SIGNAL

SPECIAL INPUT SIGNALS

- $x[n]$ = SINUSOID **FREQUENCY RESPONSE (LATER)**
- $x[n]$ has only one NON-ZERO VALUE

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

UNIT-IMPULSE



UNIT IMPULSE SIGNAL $\delta[n]$

n	...	−2	−1	0	1	2	3	4	5	6	...
$\delta[n]$	0	0	0	1	0	0	0	0	0	0	0
$\delta[n-3]$	0	0	0	0	0	0	1	0	0	0	0

$n = 3$

δ[n] is NON-ZERO When its argument is equal to ZERO

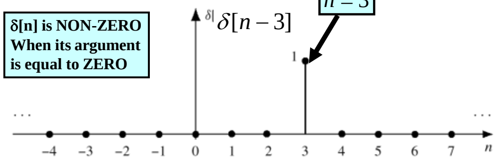
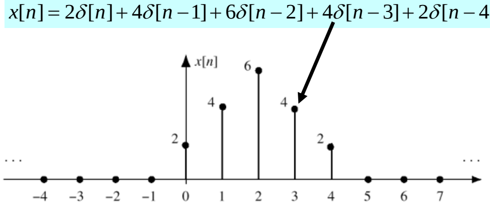


Figure 5.7 Shifted impulse sequence, $\delta[n-3]$.

MATH FORMULA for $x[n]$

- Use SHIFTED IMPULSES to write $x[n]$

$$x[n] = 2\delta[n] + 4\delta[n-1] + 6\delta[n-2] + 4\delta[n-3] + 2\delta[n-4]$$



SUM of SHIFTED IMPULSES

n	...	−2	−1	0	1	2	3	4	5	6	...
$2\delta[n]$	0	0	0	2	0	0	0	0	0	0	0
$4\delta[n-1]$	0	0	0	0	4	0	0	0	0	0	0
$6\delta[n-2]$	0	0	0	0	0	6	0	0	0	0	0
$4\delta[n-3]$	0	0	0	0	0	0	4	0	0	0	0
$2\delta[n-4]$	0	0	0	0	0	0	0	2	0	0	0
$x[n]$	0	0	0	2	4	6	4	2	0	0	0

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

$$= \ldots + x[-1]\delta[n+1] + x[0]\delta[n] + x[1]\delta[n-1] + \ldots$$

4-pt AVERAGER

- CAUSAL SYSTEM: USE PAST VALUES

$$y[n]=\tfrac{1}{4}(x[n]+x[n-1]+x[n-2]+x[n-3])$$

- INPUT = UNIT IMPULSE SIGNAL = $\delta[n]$

$$x[n]=\delta[n]$$
$$y[n]=\tfrac{1}{4}\delta[n]+\tfrac{1}{4}\delta[n-1]+\tfrac{1}{4}\delta[n-2]+\tfrac{1}{4}\delta[n-3]$$

- OUTPUT is called “IMPULSE RESPONSE”

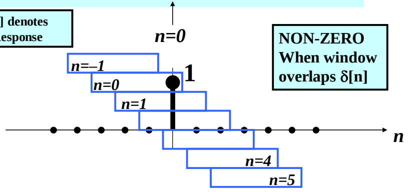
$$h[n]=\{\ldots,0,0,\tfrac{1}{4},\tfrac{1}{4},\tfrac{1}{4},\tfrac{1}{4},0,0,\ldots\}$$

4-pt Avg Impulse Response

$$y[n]=\tfrac{1}{4}(x[n]+x[n-1]+x[n-2]+x[n-3])$$

$\delta[n]$ “READS OUT” the FILTER COEFFICIENTS

$$h[n]=\{\ldots,0,0,\tfrac{1}{4},\tfrac{1}{4},\tfrac{1}{4},\tfrac{1}{4},0,0,\ldots\}$$



“h” in *h*[n] denotes Impulse Response

NON-ZERO When window overlaps δ[n]

FIR IMPULSE RESPONSE

- Convolution = Filter Definition
  - Filter Coeffs = Impulse Response

n	$n<0$	0	1	2	3	…	M	$M+1$	$n>M+1$
$x[n]=\delta[n]$	0	1	0	0	0	0	0	0	0
$y[n]=h[n]$	0	b_0	b_1	b_2	b_3	…	b_M	0	0

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

CONVOLUTIO N

FILTERING EXAMPLE

- 7-point AVERAGER
  - Removes cosine
    - By making its amplitude (A) smaller

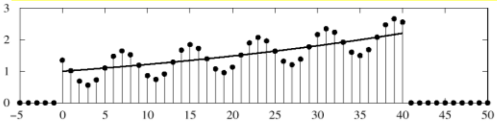
$$y_7[n]=\sum_{k=0}^{6}\left(\tfrac{1}{7}\right)x[n-k]$$

- 3-point AVERAGER
  - Changes A slightly

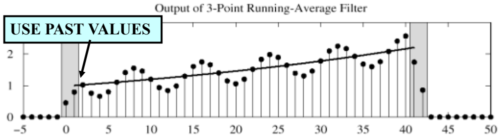
$$y_3[n]=\sum_{k=0}^{2}\left(\tfrac{1}{3}\right)x[n-k]$$

3-pt AVG EXAMPLE

$$\text{Input}: x[n]=(1.02)^n+\cos(2\pi n/8+\pi/4) \quad \text{for } 0\le n\le 40$$

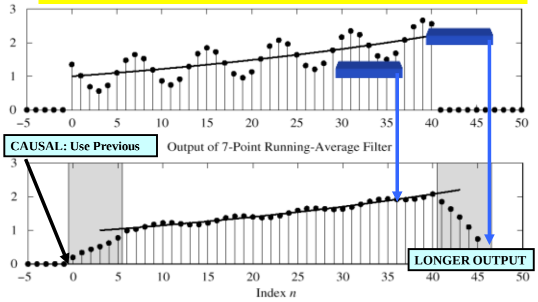


Output of 3-Point Running-Average Filter



7-pt FIR EXAMPLE (AVG)

$$\text{Input}: x[n]=(1.02)^n+\cos(2\pi n/8+\pi/4) \quad \text{for } 0\le n\le 40$$



Output of 7-Point Running-Average Filter

Linearity & Time-Invariance

- GENERAL PROPERTIES of FILTERS
  - LINEARITY
  - TIME-INVARIANCE
  - ==> **CONVOLUTION**

LTI SYSTEMS

- BLOCK DIAGRAM REPRESENTATION
  - Components for Hardware
  - Connect Simple Filters Together to Build More Complicated Systems

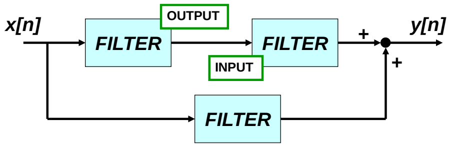
Convolution

- IMPULSE RESPONSE, $h[n]$
  - FIR case: same as $\{b_k\}$
- CONVOLUTION
  - GENERAL: $y[n] = h[n] * x[n]$
  - GENERAL CLASS of SYSTEMS
  - LINEAR and TIME-INVARIANT
- ALL **LTI** systems have $\boldsymbol{h[n]}$ & use convolution

DIGITAL FILTERING

- CONCENTRATE on the FILTER (DSP)
- DISCRETE-TIME SIGNALS
  - FUNCTIONS of n, the "time index"
  - INPUT $x[n]$
  - OUTPUT $y[n]$

BUILDING BLOCKS



- BUILD UP COMPLICATED FILTERS
  - FROM SIMPLE MODULES
  - Ex: FILTER MODULE MIGHT BE 3-pt FIR

GENERAL FIR FILTER

- FILTER COEFFICIENTS $\{b_k\}$
  - DEFINE THE FILTER

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

  - For example, $b_k = \{3, -1, 2, 1\}$

$$y[n] = \sum_{k=0}^{3} b_k x[n-k]$$
$$= 3x[n] - x[n-1] + 2x[n-2] + x[n-3]$$

MATLAB/OCTAVE for FIR FILTER

- `yy = conv(bb,xx)`
  - VECTOR `bb` contains Filter Coefficients
- FILTER COEFFICIENTS $\{b_k\}$

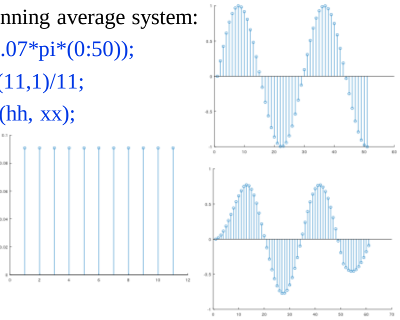
`conv2()` *for images*

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

Convolution Example

- 11-point running average system:

```
>> xx = sin(0.07*pi*(0:50));
>> hh = ones(11,1)/11;
>> yy = conv(hh, xx);
>> figure
>> stem(xx)
>> figure
>> stem(hh)
>> figure
>> stem(yy)
```

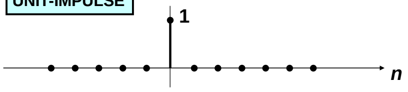


SPECIAL INPUT SIGNALS

- $x[n]$ = SINUSOID **FREQUENCY RESPONSE**
- $x[n]$ has only one NON-ZERO VALUE

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

UNIT-IMPULSE



FIR IMPULSE RESPONSE

- Convolution = Filter Definition
  - Filter Coeffs = Impulse Response

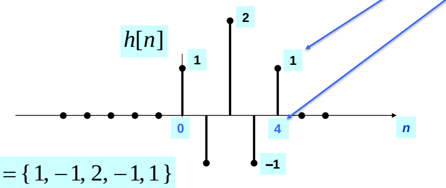
n	$n<0$	0	1	2	3	…	M	$M+1$	$n>M+1$
$x[n]=\delta[n]$	0	1	0	0	0	0	0	0	0
$y[n]=h[n]$	0	b_0	b_1	b_2	b_3	…	b_M	0	0

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

MATH FORMULA for *h*[*n*]

- Use SHIFTED IMPULSES to write $h[n]$

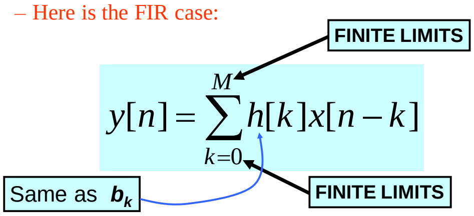
$$h[n] = \delta[n] - \delta[n-1] + 2\delta[n-2] - \delta[n-3] + \delta[n-4]$$



$$b_k = \{1, -1, 2, -1, 1\}$$

LTI: Convolution Sum

- **Output = Convolution of *x*[*n*] & *h*[*n*]**
  - NOTATION: $y[n] = h[n] * x[n]$
  - Here is the FIR case:



FINITE LIMITS

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

Same as b_k

FINITE LIMITS

CONVOLUTION Example

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

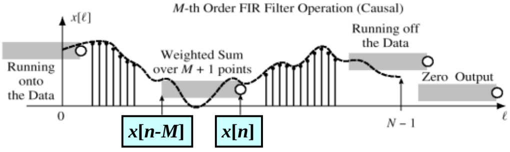
n	–1	0	1	2	3	4	5	6	7
$x[n]$	0	1	1	1	1	1	1	1	...
$h[n]$	0	1	–1	2	–1	1	0	0	0
$h[0]x[n]$	0	1	1	1	1	1	1	1	1
$h[1]x[n-1]$	0	0	–1	–1	–1	–1	–1	–1	–1
$h[2]x[n-2]$	0	0	0	2	2	2	2	2	2
$h[3]x[n-3]$	0	0	0	0	–1	–1	–1	–1	–1
$h[4]x[n-4]$	0	0	0	0	0	1	1	1	1
$y[n]$	0	1	0	2	1	2	2	2	...

GENERAL FIR FILTER

- SLIDE a Length-L WINDOW over x[n]

$$\boxed{y[n] = \sum_{k=0}^{M} b_k\, x[n-k]}$$

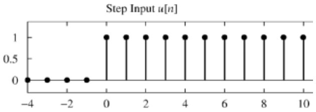
M-th Order FIR Filter Operation (Causal)



POP QUIZ

- FIR Filter is “FIRST DIFFERENCE”
  - ***y*[*n*] = *x*[*n*] - *x*[n-1]**
- INPUT is “UNIT STEP”

$$u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$$

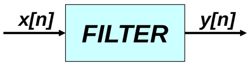


- Find ***y*[*n*]**

$$y[n] = u[n] - u[n-1]$$

$$y[n] = u[n] - u[n-1] = \delta[n]$$

HARDWARE STRUCTURES



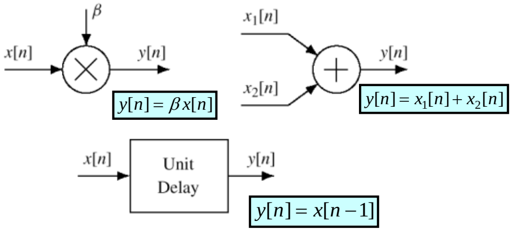
$$\boxed{y[n] = \sum_{k=0}^{M} b_k x[n-k]}$$

- INTERNAL STRUCTURE of “FILTER”
  - WHAT COMPONENTS ARE NEEDED?
  - HOW DO WE “HOOK” THEM TOGETHER?
- SIGNAL FLOW GRAPH NOTATION

HARDWARE ATOMS

- Add, Multiply & Store

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



$$\boxed{y[n] = \beta x[n]}$$

$$\boxed{y[n] = x_1[n] + x_2[n]}$$

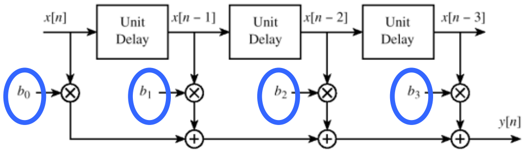
$$\boxed{y[n] = x[n-1]}$$

FIR STRUCTURE

- Direct Form

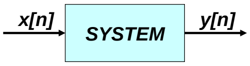
$$\boxed{y[n] = \sum_{k=0}^{M} b_k x[n-k]}$$

SIGNAL FLOW GRAPH



Block-diagram structure for the *M*th order FIR filter.

SYSTEM PROPERTIES

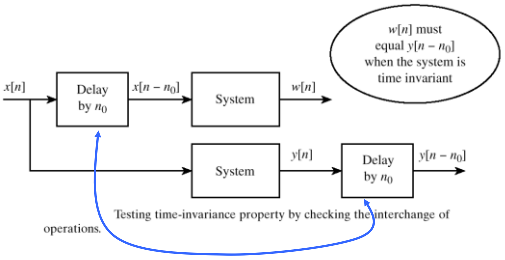


- MATHEMATICAL DESCRIPTION
- TIME-INVARIANCE
- LINEARITY
- CAUSALITY
  - “No output prior to input”

TIME-INVARIANCE

- IDEA:
  - “Time-Shifting the input will cause the same time-shift in the output”
- EQUIVALENTLY,
  - We can prove that
    - The time origin (n=0) is picked arbitrary

TESTING Time-Invariance

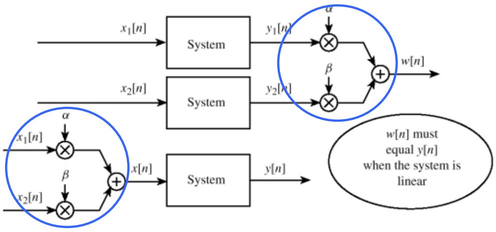


Testing time-invariance property by checking the interchange of operations.

LINEAR SYSTEM

- LINEARITY = Two Properties
- SCALING
  - “Doubling $x[n]$ will double $y[n]$”
- SUPERPOSITION:
  - “Adding two inputs gives an output that is the sum of the individual outputs”

TESTING LINEARITY



Testing linearity by checking the interchange of operations.

LTI SYSTEMS

- LTI: Linear & Time-Invariant
- COMPLETELY CHARACTERIZED by:
  - **IMPULSE RESPONSE** $h[n]$
  - **CONVOLUTION**: $y[n] = x[n]*h[n]$
    - The “rule”defining the system can ALWAYS be re-written as convolution
- FIR Example: $h[n]$ is same as b_k

POP QUIZ

- FIR Filter is “FIRST DIFFERENCE”
  - **$y[n] = x[n] - x[n-1]$**
- Write output as a convolution
  - Need impulse response

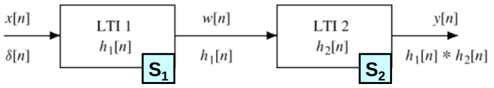
$$h[n]=\delta[n]-\delta[n-1]$$

  - Then, another way to compute the output:

$$y[n]=\left(\delta[n]-\delta[n-1]\right)*x[n]$$

CASCADE SYSTEMS

- Does the order of S_1 & S_2 matter?
  - NO, LTI SYSTEMS can be rearranged !!!
  - WHAT ARE THE FILTER COEFFS? $\{b_k\}$



A Cascade of Two LTI Systems.

CASCADE EQUIVALENT

- Find “overall” $h[n]$ for a cascade ?

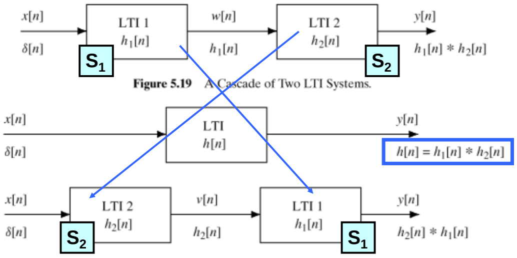


Figure 5.19 A Cascade of Two LTI Systems.

Switching the order of cascaded LTI systems.

Frequency Response of FIR Filters

- SINUSOIDAL INPUT SIGNAL
  - DETERMINE the FIR FILTER OUTPUT
- FREQUENCY RESPONSE of FIR
  - PLOTTING vs. Frequency
  - MAGNITUDE vs. Freq
  - PHASE vs. Freq

MAG

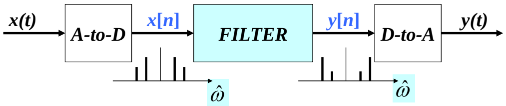
PHASE

$$H(e^{j\hat{\omega}}) = |H(e^{j\hat{\omega}})| e^{j\angle H(e^{j\hat{\omega}})}$$

DOMAINS: Time & Frequency

- **Time-Domain: “*n*” = time**
  - $x[n]$ discrete-time signal
  - $x(t)$ continuous-time signal
- **Frequency Domain (sum of sinusoids)**
  - Spectrum vs. f (Hz)
    - ANALOG vs. DIGITAL
  - Spectrum vs. omega-hat
- Move back and forth QUICKLY

DIGITAL “FILTERING”



- CONCENTRATE on the **SPECTRUM**
- SINUSOIDAL INPUT
  - INPUT $x[n]$ = SUM of SINUSOIDS
  - Then, OUTPUT $y[n]$ = SUM of SINUSOIDS

SINUSOIDAL RESPONSE

- INPUT: x[n] = SINUSOID
- OUTPUT: y[n] will also be a SINUSOID
  - Different Amplitude and Phase
  - **SAME** Frequency
- AMPLITUDE & PHASE CHANGE
  - Called the **FREQUENCY RESPONSE**

COMPLEX EXPONENTIAL

$$x[n] = Ae^{j\varphi}e^{j\hat{\omega}n} \qquad -\infty < n < \infty$$

x[n] is the input signal—a complex exponential

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

FIR DIFFERENCE EQUATION

COMPLEX EXP OUTPUT

- Use the FIR “Difference Equation”

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] = \sum_{k=0}^{M} b_k Ae^{j\varphi}e^{j\hat{\omega}(n-k)}$$

$$= \left(\sum_{k=0}^{M} b_k e^{j\hat{\omega}(-k)}\right)Ae^{j\varphi}e^{j\hat{\omega}n}$$

$$= H(\hat{\omega})Ae^{j\varphi}e^{j\hat{\omega}n}$$

FREQUENCY RESPONSE

- At each frequency, we can DEFINE

$$\boxed{H(e^{j\hat{\omega}}) = \sum_{k=0}^{M} b_k e^{-j\hat{\omega}k}} \longleftarrow$$

FREQUENCY RESPONSE

- Complex-valued formula
  - Has MAGNITUDE vs. frequency
  - And PHASE vs. frequency
- Notation: $H(e^{j\hat{\omega}})$ in place of $H(\hat{\omega})$

EXAMPLE 1

$$\{b_k\} = \{1,\ 2,\ 1\}$$

$$H(e^{j\hat{\omega}}) = 1 + 2e^{-j\hat{\omega}} + e^{-j2\hat{\omega}}$$

$$= e^{-j\hat{\omega}}(e^{j\hat{\omega}} + 2 + e^{-j\hat{\omega}})$$

$$= e^{-j\hat{\omega}}(2 + 2\cos\hat{\omega})$$

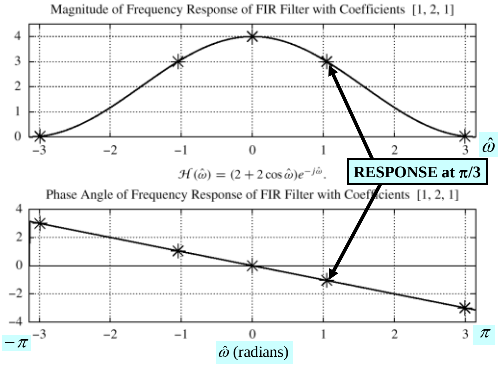
EXPLOIT SYMMETRY

Since $(2 + 2\cos\hat{\omega}) \geq 0$

Magnitude is $\left|H(e^{j\hat{\omega}})\right| = (2 + 2\cos\hat{\omega})$

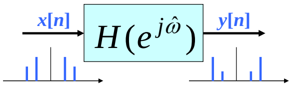
and Phase is $\angle H(e^{j\hat{\omega}}) = -\hat{\omega}$

PLOT of FREQ RESPONSE



EXAMPLE 2

Find $y[n]$ when $H(e^{j\hat{\omega}})$ is known and $x[n] = 2e^{j\pi/4}e^{j(\pi/3)n}$



$$H(e^{j\hat{\omega}}) = (2 + 2\cos\hat{\omega})e^{-j\hat{\omega}}$$

EXAMPLE 2 (answer)

Find $y[n]$ when $x[n]=2e^{j\pi/4}e^{j(\pi/3)n}$

One Step - evaluate $H(e^{j\hat{\omega}})$ at $\hat{\omega}=\pi/3$

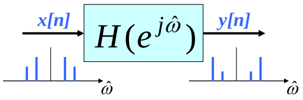
$$H(e^{j\hat{\omega}})=(2+2\cos\hat{\omega})e^{-j\hat{\omega}}$$

$$H(e^{j\hat{\omega}})=3e^{-j\pi/3} \quad @\,\hat{\omega}=\pi/3$$

$$y[n]=\left(3e^{-j\pi/3}\right)\times 2e^{j\pi/4}e^{j(\pi/3)n}=6e^{-j\pi/12}e^{j(\pi/3)n}$$

EXAMPLE: COSINE INPUT

Find $y[n]$ when $H(e^{j\hat{\omega}})$ is known and $x[n]=2\cos(\tfrac{\pi}{3}n+\tfrac{\pi}{4})$



$$H(e^{j\hat{\omega}})=(2+2\cos\hat{\omega})e^{-j\hat{\omega}}$$

EX: COSINE INPUT

Find $y[n]$ when $x[n]=2\cos(\tfrac{\pi}{3}n+\tfrac{\pi}{4})$

$$2\cos(\tfrac{\pi}{3}n+\tfrac{\pi}{4})=e^{j(\pi n/3+\pi/4)}+e^{-j(\pi n/3+\pi/4)}$$

$$\Rightarrow x[n]=x_1[n]+x_2[n]$$

Use Linearity

$$y_1[n]=H(e^{j\pi/3})e^{j(\pi n/3+\pi/4)}$$

$$y_2[n]=H(e^{-j\pi/3})e^{-j(\pi n/3+\pi/4)}$$

$$\Rightarrow y[n]=y_1[n]+y_2[n]$$

EX: COSINE INPUT (ans-2)

Find $y[n]$ when $x[n]=2\cos(\tfrac{\pi}{3}n+\tfrac{\pi}{4})$

$$H(e^{j\hat{\omega}})=(2+2\cos\hat{\omega})e^{-j\hat{\omega}}$$

$$y_1[n]=H(e^{j\pi/3})e^{j(\pi n/3+\pi/4)}=3e^{-j(\pi/3)}e^{j(\pi n/3+\pi/4)}$$

$$y_2[n]=H(e^{-j\pi/3})e^{-j(\pi n/3+\pi/4)}=3e^{j(\pi/3)}e^{-j(\pi n/3+\pi/4)}$$

$$y[n]=3e^{j(\pi n/3-\pi/12)}+3e^{-j(\pi n/3-\pi/12)}$$

$$\Rightarrow y[n]=6\cos(\tfrac{\pi}{3}n-\tfrac{\pi}{12})$$

MATLAB/OCTAVE: FREQUENCY RESPONSE

- `HH = freqz(bb,1,ww)`
  - VECTOR `bb` contains Filter Coefficients
- FILTER COEFFICIENTS $\{b_k\}$

$$\boxed{H(e^{j\hat{\omega}})=\sum_{k=0}^{M}b_k e^{-j\hat{\omega}k}}$$

Time & Frequency Relation

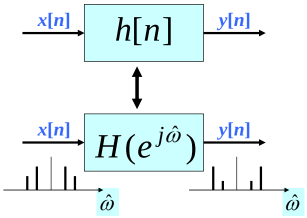
- Get Frequency Response from $h[n]$
  - Here is the FIR case:

$$\boxed{H(e^{j\hat{\omega}})=\sum_{k=0}^{M}b_k e^{-j\hat{\omega}k}=\sum_{k=0}^{M}h[k]e^{-j\hat{\omega}k}}$$

IMPULSE RESPONSE

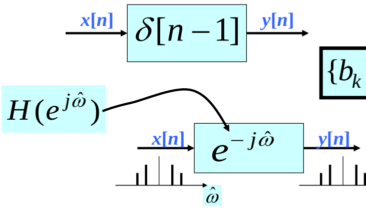
BLOCK DIAGRAMS

- Equivalent Representations



UNIT-DELAY SYSTEM

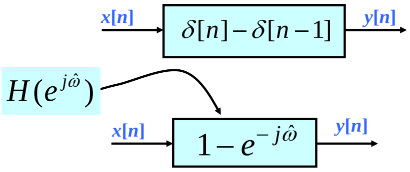
Find $h[n]$ and $H(e^{j\hat{\omega}})$ for $y[n] = x[n-1]$



$\boxed{\{b_k\} = \{\,0,\ 1\,\}}$

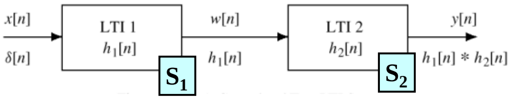
FIRST DIFFERENCE SYSTEM

Find $h[n]$ and $H(e^{j\hat{\omega}})$ for the Difference Equation: $\quad y[n] = x[n] - x[n-1]$



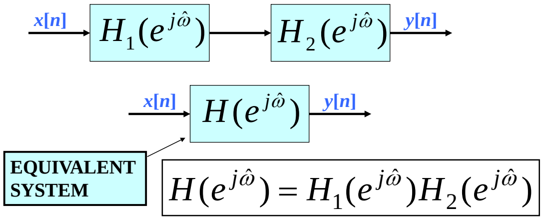
CASCADE SYSTEMS

- Does the order of S_1 & S_2 matter?
  - NO, LTI SYSTEMS can be rearranged !!!
  - WHAT ARE THE FILTER COEFFS? $\{b_k\}$
  - WHAT is the overall FREQUENCY RESPONSE ?



CASCADE EQUIVALENT

- MULTIPLY the Frequency Responses



EQUIVALENT SYSTEM

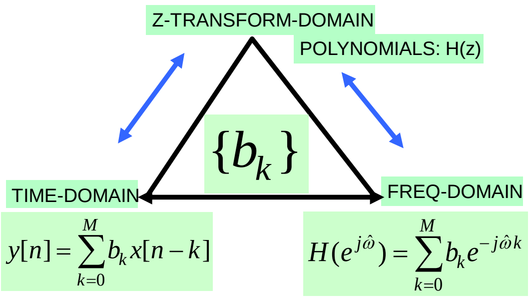
$$H(e^{j\hat{\omega}}) = H_1(e^{j\hat{\omega}})H_2(e^{j\hat{\omega}})$$

Z Transforms: Introduction

- INTRODUCE the Z-TRANSFORM
  - Give Mathematical Definition
  - Show how the H(z) POLYNOMIAL simplifies analysis
    - **CONVOLUTION** is SIMPLIFIED !
- Z-Transform can be applied to
  - FIR Filter: $h[n] \rightarrow H(z)$
  - Signals: $x[n] \rightarrow X(z)$

$$H(z) = \sum_n h[n] z^{-n}$$

THREE DOMAINS



Z-TRANSFORM-DOMAIN

POLYNOMIALS: H(z)

$\{b_k\}$

TIME-DOMAIN

FREQ-DOMAIN

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

$H(e^{j\hat{\omega}})=\sum_{k=0}^{M} b_k e^{-j\hat{\omega}k}$

Three main reasons for Z-Transform

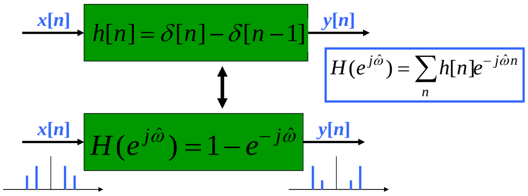
- Offers compact and convenient notation for describing digital signals and systems
- Widely used by DSP designers, and in the DSP literature
- Pole-zero description of a processor is a great help in visualizing its stability and frequency response characteristic

TRANSFORM CONCEPT

- Move to a new domain where
  - – OPERATIONS are EASIER & FAMILIAR
  - – Use POLYNOMIALS
- TRANSFORM both ways
  - – $x[n] \rightarrow X(z)$ (into the z domain)
  - – $X(z) \rightarrow x[n]$ (back to the time domain)

“TRANSFORM” EXAMPLE

- Equivalent Representations



$x[n]$

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

$y[n]$

$H(e^{j\hat{\omega}})=\sum_n h[n]e^{-j\hat{\omega}n}$

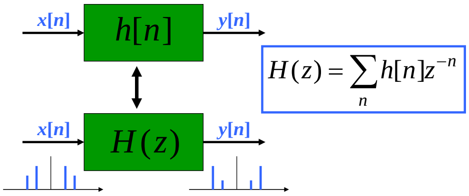
$x[n]$

$H(e^{j\hat{\omega}})=1-e^{-j\hat{\omega}}$

$y[n]$

Z-TRANSFORM IDEA

- **POLYNOMIAL** REPRESENTATION



$x[n]$

$h[n]$

$y[n]$

$H(z)=\sum_n h[n]z^{-n}$

$x[n]$

$H(z)$

$y[n]$

Z-Transform DEFINITION

- POLYNOMIAL Representation of LTI SYSTEM: $H(z)=\sum_n h[n]z^{-n}$
- EXAMPLE:

APPLIES to Any SIGNAL

$\{h[n]\}=\{2,0,-3,0,2\}$

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

POLYNOMIAL in z^{-1}

Z-Transform EXAMPLE-1

- ANY SIGNAL has a z-Transform:

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

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

$$X(z)=?$$

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

Z-Transform EXAMPLE-2

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

EXPONENT GIVES TIME LOCATION

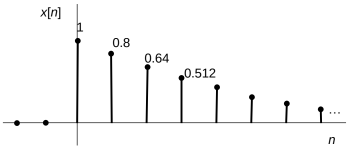
$$x[n]=\begin{cases}0 & n<0\\ 1 & n=0\\ -2 & n=1\\ 0 & n=2\\ 3 & n=3\\ 0 & n=4\\ -1 & n=5\\ 0 & n>5\end{cases}$$

$$x[n]=?$$

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

Z-Transform EXAMPLE-3

- Find the Z-Transform of the exponentially decaying signal shown in the following figure, expressing it as compact as possible.



Z-Transform EXAMPLE-3

- The geometric series formula

$$\frac{1}{1-x}=\sum_{n=0}^{\infty}x^n=1+x+x^2+x^3+\cdots,$$

$$e^x=\sum_{n=0}^{\infty}\frac{x^n}{n!}=1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\cdots,$$

Z-Transform EXAMPLE-3

- The Z-Transform of the signal:

$$\begin{aligned}X(z)&=\sum_{n=0}^{\infty}x[n]z^{-n}\\&=1+0.8z^{-1}+0.64z^{-2}+0.512z^{-3}+\cdots\\&=1+0.8(z^{-1})+0.64(z^{-1})^2+0.512(z^{-1})^3+\cdots\\&=1+(0.8z^{-1})+(0.8z^{-1})^2+(0.8z^{-1})^3+\cdots\\&=\frac{1}{1-0.8z^{-1}}=\frac{z}{z-0.8}\end{aligned}$$

Z-Transform EXAMPLE-4

- Find and sketch, the signal corresponding to the Z-Transform:

$$X(z)=\frac{1}{z+1.2}$$

Z-Transform EXAMPLE-4

- Recasting X(z) as a power series in z^{-1}, we obtain:

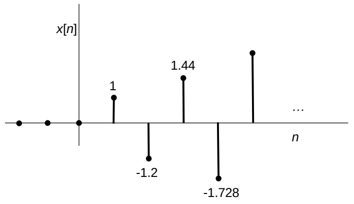
$$X(z)=\frac{1}{(z+1.2)}=\frac{z^{-1}}{(1+1.2z^{-1})}=z^{-1}(1+1.2z^{-1})^{-1}$$
$$=z^{-1}\{1+(-1.2z^{-1})+(-1.2z^{-1})^2+(-1.2z^{-1})^3+\cdots\}$$
$$=z^{-1}-1.2z^{-2}+1.44z^{-3}-1.728z^{-4}+\cdots$$

- Succesive values of $x[n]$, starting at n=0, are therefore:

  0, 1, -1.2, 1.44, -1.728, $\cdots$

Z-Transform EXAMPLE-4

- $x[n]$ is shown in the following figure:



Z-Transform of FIR Filter

- CALLED the **SYSTEM FUNCTION**
  - $h[n]$ is same as $\{b_k\}$

SYSTEM FUNCTION

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

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

FIR DIFFERENCE EQUATION

CONVOLUTION

Z-Transform of FIR Filter

- Get $H(z)$ DIRECTLY from the $\{b_k\}$
- Example 7.3 in the book:

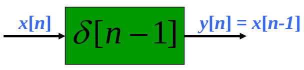
$$y[n]=6x[n]-5x[n-1]+x[n-2]$$

$$\{b_k\}=\{6,-5,1\}$$

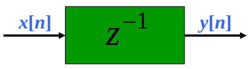
$$H(z)=\sum b_k z^{-1}=6-5z^{-1}+z^{-2}$$

Ex. DELAY SYSTEM

- UNIT DELAY: find $h[n]$ and $H(z)$



$$H(z)=\sum\delta[n-1]z^{-n}\;=z^{-1}$$



DELAY EXAMPLE

- UNIT DELAY: find $y[n]$ via polynomials
  - – x[n] = {3,1,4,1,5,9,0,0,0,...}

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

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

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

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

DELAY PROPERTY

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

$$x[n-1] \quad \Longleftrightarrow \quad z^{-1}X(z)$$

Time delay of n_0 *samples multiplies the z-transform by* z^{-n_0}

$$x[n-n_0] \quad \Longleftrightarrow \quad z^{-n_0}X(z)$$

GENERAL I/O PROBLEM

- Input is $x[n]$, find $y[n]$ (for FIR, $h[n]$)
- How to combine $X(z)$ and $H(z)$?

Example

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

$$\text{and} \quad h[n] = \delta[n] + 2\delta[n-1] + 3\delta[n-2] + 4\delta[n-3]$$

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

$$\text{and} \quad H(z) = 1 + 2z^{-1} + 3z^{-2} + 4z^{-3}$$

FIR Filter = CONVOLUTION

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

CONVOLUTION

```
x[n], X(z)     0   +1   -1   +1   -1
h[n], H(z)     1    2    3    4
--------------------------------------------
               0   +1   -1   +1   -1
                    0   +2   -2   +2   -2
                         0   +3   -3   +3   -3
                              0   +4   -4   +4   -4
--------------------------------------------
 y[n], Y(z)    0   +1   +1   +2   +2   -3   +1   -4
```

CONVOLUTION PROPERTY

- PROOF:

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

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

MULTIPLY Z-TRANSFORMS

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

CONVOLUTION EXAMPLE

- **MULTIPLY** the z-TRANSFORMS:

Example

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

$$\text{and} \quad h[n] = \delta[n] + 2\delta[n-1] + 3\delta[n-2] + 4\delta[n-3]$$

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

$$\text{and} \quad H(z) = 1 + 2z^{-1} + 3z^{-2} + 4z^{-3}$$

MULTIPLY $H(z)X(z)$

CONVOLUTION EXAMPLE

- Finite-Length input $x[n]$
- FIR Filter (L=4)

MULTIPLY Z-TRANSFORMS

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

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

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

$$+ \quad (-2+3-4)z^{-5} + (-3+4)z^{-6} + (-4)z^{-7}$$

$$= z^{-1} + z^{-2} + 2z^{-3} + 2z^{-4} - 3z^{-5} + z^{-6} - 4z^{-7}$$

$y[n]$ = ?

CASCADE SYSTEMS

- Does the order of S_1 & S_2 matter?
  - NO, LTI SYSTEMS can be rearranged !!!
  - Remember: $h_1[n] * h_2[n]$
  - How to combine $H_1(z)$ and $H_2(z)$?

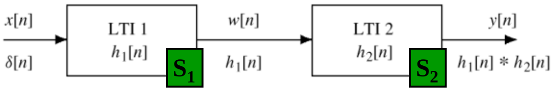
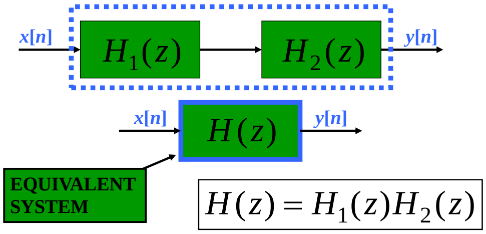


Figure 5.19 A Cascade of Two LTI Systems.

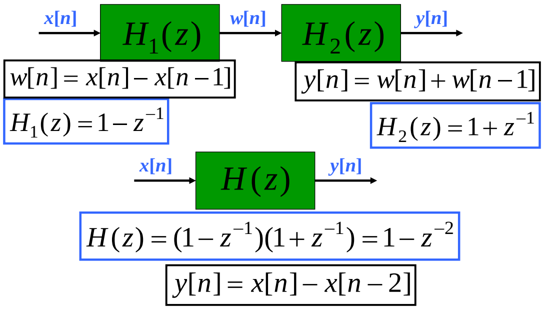
CASCADE EQUIVALENT

- Multiply the System Functions



$$H(z) = H_1(z)H_2(z)$$

CASCADE EXAMPLE



$$w[n] = x[n] - x[n-1]$$

$$H_1(z) = 1 - z^{-1}$$

$$y[n] = w[n] + w[n-1]$$

$$H_2(z) = 1 + z^{-1}$$

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

$$y[n] = x[n] - x[n-2]$$