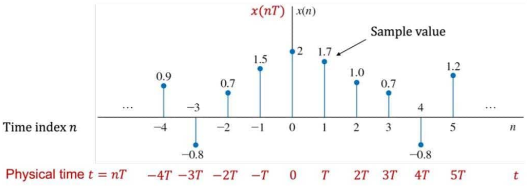
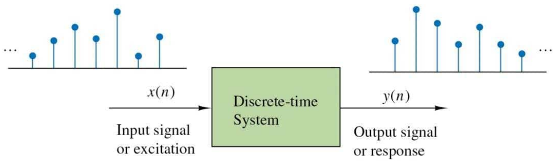
EECE-5666: Digital Signal Processing

Discrete-Time Signals and Systems





0

Representation of Discrete-Time Signals

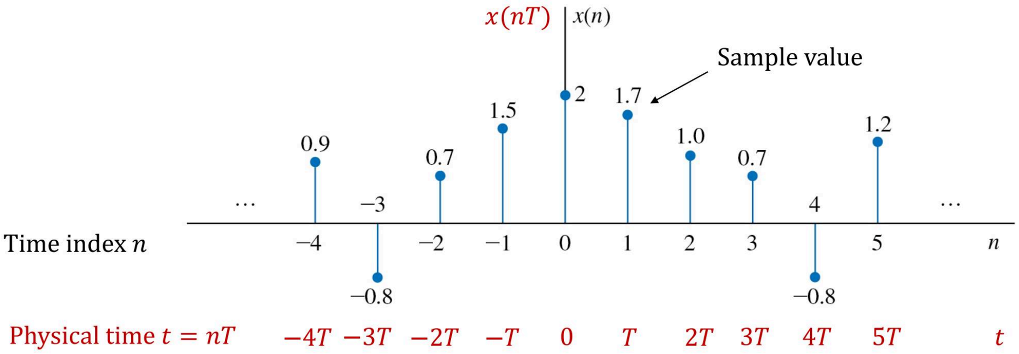
A discrete-time signal $x(n)$ is a sequence of numbers defined for all values of n, $-\infty < n < \infty$

Functional	$x(n) = \begin{cases} \left(\frac{1}{2}\right)^n, n \geq 0 \\ 0, n < 0 \end{cases}$
Tabular	$\begin{array}{c\|cccccccc} n & \cdots & -2 & -1 & 0 & 1 & 2 & 3 & \cdots \\ \hline x(n) & \cdots & 0 & 0 & 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \cdots \end{array}$
Sequence	$x(n) = \left\{ \cdots \ 0 \ \underset{\substack{\uparrow \\ n=0}}{1} \ \frac{1}{2} \ \frac{1}{4} \ \frac{1}{8} \ \cdots \right\}$

A discrete-time signal is not defined for non-integer values of n.
For example, the value of $x(3/2)$ is unspecified, not zero!

1

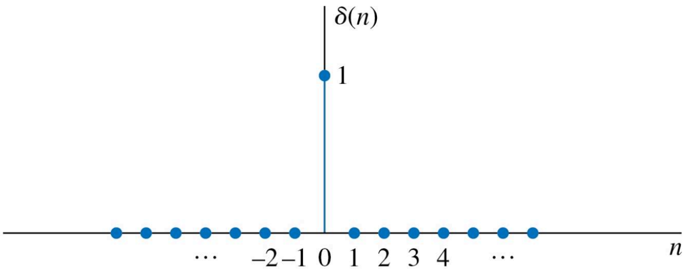
Graphical Representation of a Discrete-Time Signal



EECE-5666: Digital Signal Processing Signals and Systems 2

2

The Unit Sample or Impulse Sequence

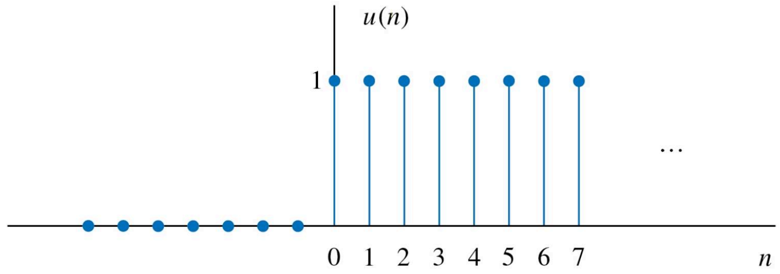


$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$

EECE-5666: Digital Signal Processing Signals and Systems 3

3

Unit Step Sequence



$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

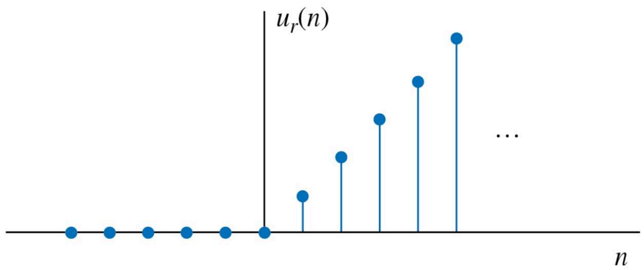
$$\delta(n) = u(n) - u(n-1)$$

$$u(n) = \sum_{k=0}^{\infty} \delta(n-k)$$

EECE-5666: Digital Signal Processing Signals and Systems 4

4

Unit Ramp Sequence



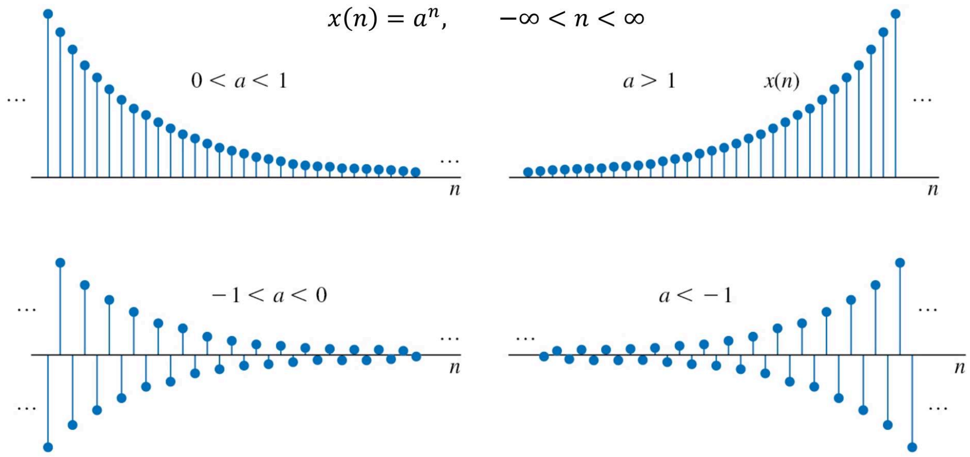
$$u_r(n) = nu(n)$$

EECE-5666: Digital Signal Processing Signals and Systems 5

5

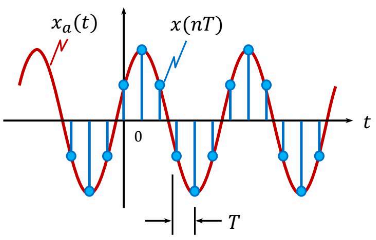
Graphical Representation of Exponential Signal

$$x(n) = a^n, \qquad -\infty < n < \infty$$



6

Continuous-Time to Discrete-Time Sinusoidal Signals



$$x_a(t) = A\cos(\Omega t + \theta) = A\cos(2\pi F t + \theta)$$

$$x(n) = x_a(nT), \qquad T = \text{Sampling period}$$

$$F_s = \frac{1}{T} = \text{Sampling frequency}$$

$$x(n) = x_a(nT) = A\cos(2\pi F n T + \theta) = A\cos\left(2\pi \frac{F}{F_s} n + \theta\right)$$

$f = \frac{F}{F_s}$ Normalized cyclic frequency

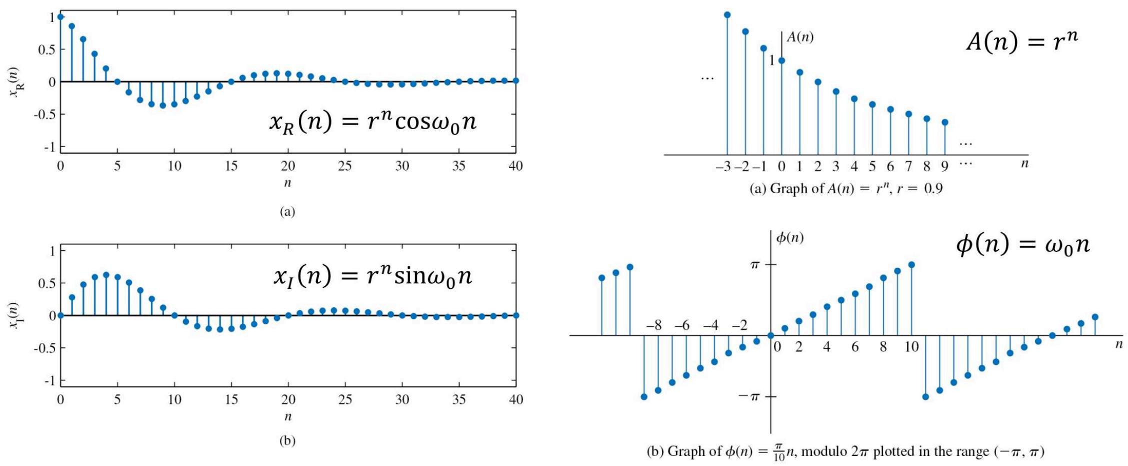
$\omega = 2\pi f$ Normalized angular frequency

$$x(n) = A\cos(2\pi f n + \theta) = A\cos(\omega n + \theta)$$

7

Representation of Complex Exponential Sequences

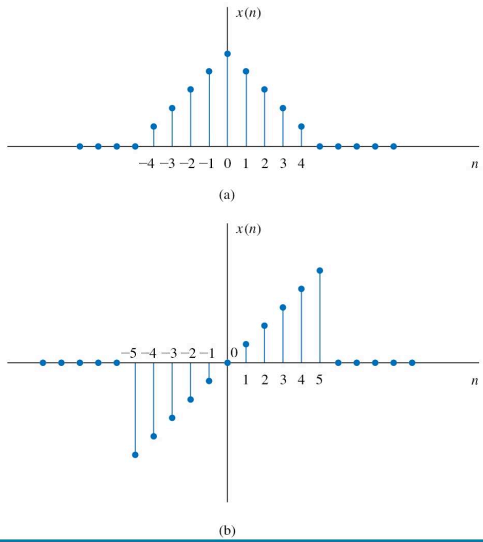
$$x(n) = a^n = \left(re^{j\omega_0}\right)^n = r^n e^{j\omega_0 n} = r^n \cos\omega_0 n + jr^n \sin\omega_0 n$$



Some Properties of Sequences

Property	Definition	Comments
Duration	The number of samples from the first non-zero sample $x(n_1)$ to the last non-zero sample $x(n_2)$	Finite/Infinite $L = n_2 - n_1 + 1$
Periodicity	$x(n) = x(n + kN)$ Fundamental period = Smallest N	Periodic/Aperiodic
Symmetries (real $x(n)$)	$x(n) = x(-n)$ (even) $x(n) = -x(-n)$ (odd)	Yes/No
Energy	$E_x = \sum_{n=-\infty}^{\infty} \lvert x(n)\rvert^2$	Finite/Infinite
Power	$P_x = \lim_{L\to\infty} \frac{1}{2L+1} \sum_{n=-L}^{L} \lvert x(n)\rvert^2$	Finite/Infinite

Example of Even and Odd Signals



Even symmetry

$$x(-n) = x(n)$$

Odd symmetry

$$x(-n) = -x(n)$$

10

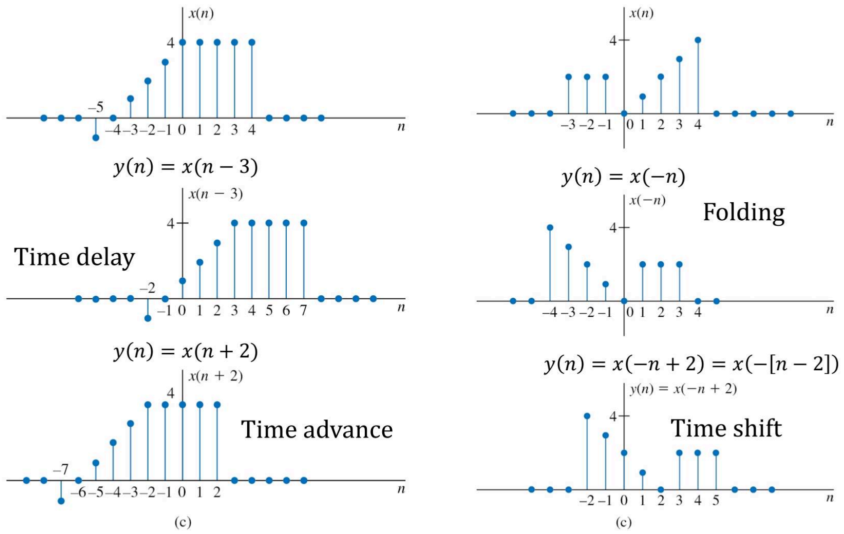
Some Transformations of Sequences

Transformations of the dependent variable	
Scaling	$y(n) = ax(n)$
Addition/Subtraction	$y(n) = x_1(n) \pm x_2(n)$
Multiplication	$y(n) = x_1(n)x_2(n)$
Division	$y(n) = x_1(n)/x_2(n)$

Transformations of the independent variable	
Time Shifting	$y(n) = x(n - n_0)$
Time Reversal	$y(n) = x(-n)$
Decimation	$y(n) = x(nD)$

11

Illustration of Shifting and Folding



Index mapping

$m = f(n)$

Signal sample mapping

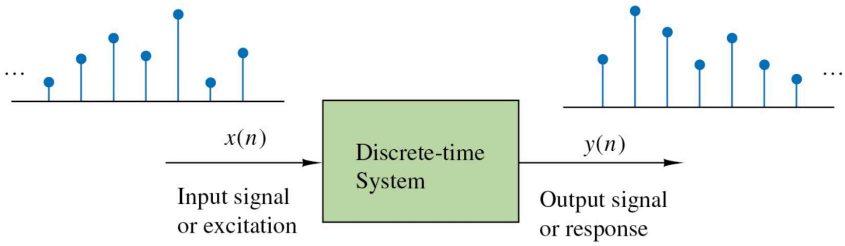
$y(n) = x(f(n))$

It is important to understand these operations by doing some simple examples by hand drawing!

12

Discrete-Time Systems

A discrete-time system is a computational process or algorithm that transforms a sequence $x(n)$, called the input signal, into another sequence $y(n)$, called the output signal



A discrete-time system can be implemented by (a) software on a general purpose computer, (b) software on a special purpose DSP, and (c) dedicated hardware

13

Discrete-Time Systems

- **Examples**
  - Identity system:
    $$y(n) = x(n)$$
  - Unit delay system:
    $$y(n) = x(n-1)$$
  - Unit advance system:
    $$y(n) = x(n+1)$$
  - Moving Average (MA) system:
    $$y(n) = \frac{1}{3}\{x(n+1) + x(n) + x(n-1)\}$$
  - Median filter:
    $$y(n) = \text{Median}\{x(n) + x(n-1) + x(n-2)\}$$
  - Auto-Regressive (AR) system:
    $$y(n) = 0.5y(n-1) + x(n)$$
  - Downsampling system:
    $$y(n) = x(nM)$$

14

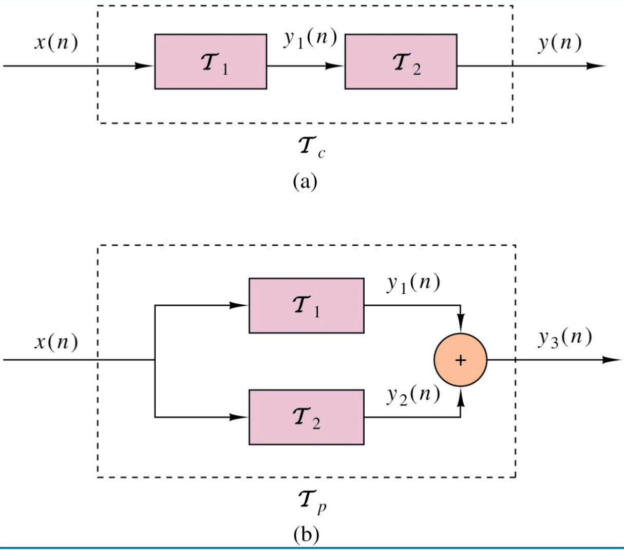
Discrete-Time System Properties

Property	Input Output	Comments
	$x(n) \to y(n)$ $x_k(n) \to y_k(n)$	
Linearity	$\sum_k c_k x_k(n) \to \sum_k c_k y_k(n)$	Imposed for simplicity
Time–Invariance	$x(n-n_0) \to y(n-n_0)$	Imposed for simplicity
Stability	$\|x(n)\| \le M_x < \infty \to \|y(n)\| \le M_x < \infty$	Required (Unstable systems "blow-up")
Causality	$x(n) = 0, n \le n_0 \to y(n) = 0, n \le \mathrm{n}_0$	Useful
Practical Realizability	Finite memory and finite number of computations per output sample	Practical Requirement
Real–Time Operation	The computation of each output sample is completed before the arrival of the next input sample	Real-Time Requirement

Note: Verification depends on system description (black-box or I/O equation) and known properties

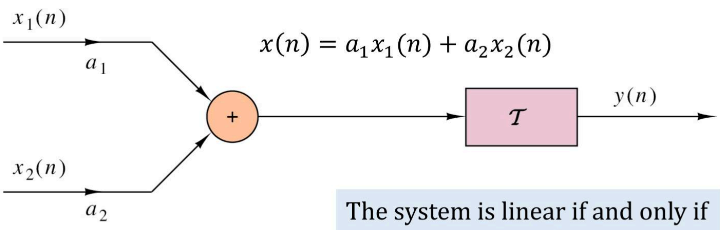
15

Cascade and Parallel Interconnections of Systems



16

Testing for Linearity



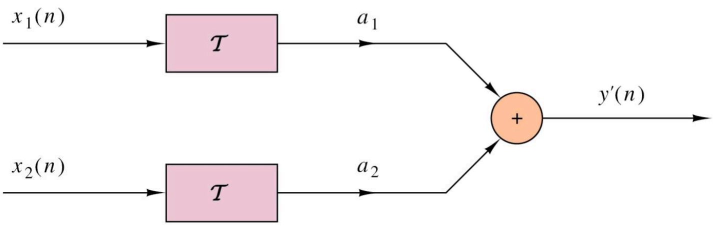
$x(n) = a_1 x_1(n) + a_2 x_2(n)$

Square-Law System

$y(n) = x^2(n)$

$y(n) = a_1^2 x_1^2(n) + a_2^2 x_2^2(n) + 2a_1 a_2 x_1(n) x_2(n)$

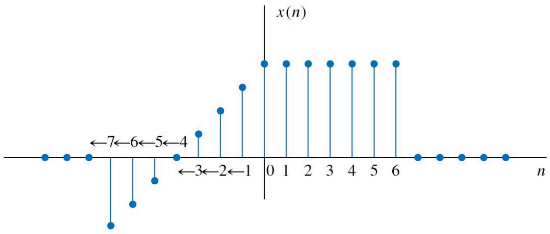
The system is linear if and only if $y(n) = y'(n)$



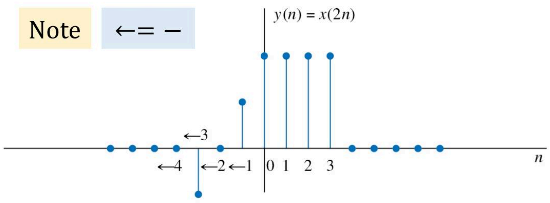
$y'(n) = a_1 x_1^2(n) + a_2 x_2^2(n)$
$\neq y(n)$
$\Rightarrow$ System is nonlinear

17

Down-Sampling System



(a)



(b)

Note ←= −

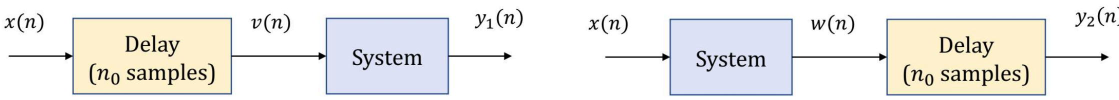
$y(n) = x(nD), \; D = \text{Integer}$

$$x(n) = a_1 x_1(n) + a_2 x_2(n)$$
$$y(n) = a_1 x_1(nD) + a_2 x_2(nD)$$

$$y'(n) = a_1 x_1(nD) + a_2 x_2(nD)$$
$$= y(n)$$
$\Rightarrow$ System is linear

18

Testing for Time Invariance



If $y_1(n) = y_2(n)$ then the system is time−invariant

Square-law system

$y(n) = x^2(n)$

$$v(n) = x(n - n_0)$$
$$y_1(n) = x^2(n - n_o)$$

$$w(n) = x^2(n)$$
$$y_2(n) = w(n - n_o) = x^2(n - n_o)$$
$$= y_1(n)$$
$\Rightarrow$ System is time−invariant

Down-sampler

$y(n) = x(nD)$

$$v(n) = x(n - n_0)$$
$$y_1(n) = x(nD - n_0)$$

$$w(n) = x(nD)$$
$$y_2(n) = w(n - n_o) = x\big((n - n_o)D\big)$$
$$\neq y_1(n)$$
$\Rightarrow$ System is time−varying

19

Examples of Discrete-Time Systems

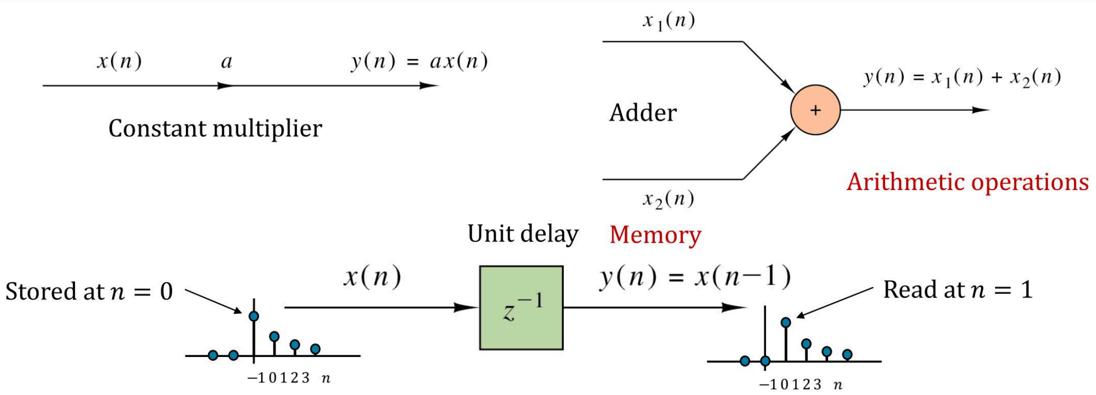
System	Linear	Time-Invariant	Causal	Stable	Comments
$y(n) = x^2(n)$	No	Yes	Yes	Yes	Square-law
$y(n) = x(-n)$	Yes	No	No	Yes	Time-flip
$y(n) = \sum_{k=-\infty}^{n} x(k)$	Yes	Yes	Yes	No	Accumulator
$y(n) = x(n) - x(n-1)$	Yes	Yes	Yes	Yes	First-difference
$y(n) = \frac{1}{2M+1}\sum_{k=-M}^{M} x(n-k)$	Yes	Yes	No	Yes	Moving-Average
$y(n) = \text{round}\{x(n)\}$	No	Yes	Yes	Yes	Quantizer

Note: Proof of a property should be done for every possible input.
A counter-example is sufficient to prove lack of a property.

EECE-5666: Digital Signal Processing Signals and Systems 20

20

Elementary Discrete-Time Systems



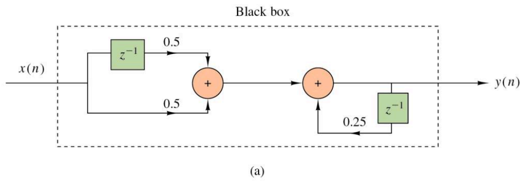
A unit delay is a memory location where we can store a number at a given sampling interval and read it at the next sampling interval.

Every LTI system can be implemented using the elementary building blocks

EECE-5666: Digital Signal Processing Signals and Systems 21

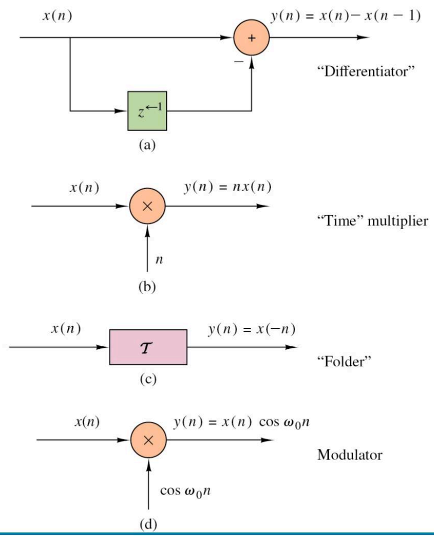
21

Examples of System Realizations



Equivalent block diagram realizations of the system

$y(n) = 0.25y(n-1) + 0.5x(n) + 0.5x(n-1)$



EECE-5666: Digital Signal Processing Signals and Systems 22

22

An Important and Useful Result

- In discrete-time signals and systems we will always encounter the summation of exponential samples, that is

$$\sum_{n=N_1}^{N_2} r^n, \qquad N_1 \le N_2$$

which is equivalent to the integral of exponential signal in continuous time.

- **Basic Result**:

$$\sum_{n=0}^{\infty} r^n = \frac{1}{1-r}, \qquad \text{provided } |r| < 1$$

- **Additional Results**:

$$\sum_{n=N_1}^{\infty} r^n = \frac{r^{N_1}}{1-r}, \qquad \text{provided } |r| < 1$$

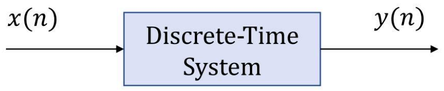
$$\sum_{n=0}^{N-1} r^n = \frac{1-r^N}{1-r}, \qquad \text{no condition on } r$$

$$\sum_{n=N_1}^{N_2-1} r^n = \frac{r^{N_1}-r^{N_2}}{1-r}, \qquad \text{no condition on } r$$

EECE-5666: Digital Signal Processing Signals and Systems 23

23

Why Linearity and Time-Invariance?



To define an arbitrary "black-box" system we must know the output for every input!

To bypass this intractable task we must impose linearity and time-invariance!

Linearity

- Decompose input signal into a linear combination of basic signals
- Choose basic signals so that decomposition and/or response is easy to compute

Time-Invariance

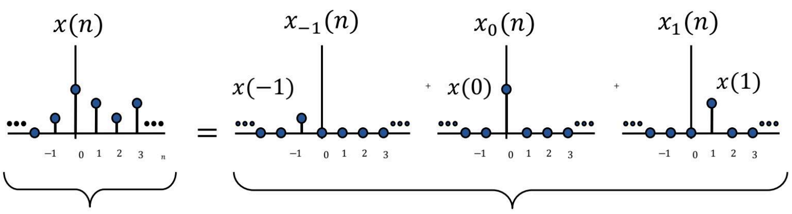
- The shape of response does not depend on the time the input was applied

Useful Decompositions for LTI Systems

- Delayed impulses $\Rightarrow$ Convolution (easy input decomposition)
- Complex exponentials $\Rightarrow$ Fourier analysis (easy response computation)

We shall focus on LTI systems!

Signal Decomposition into Impulses



$$\{x(n)\} = \cdots + x(-1)\{\delta(n+1)\} + x(0)\{\delta(n)\} + x(1)\{\delta(n-1)\} + \cdots$$

Entire sequence as one "entity"

Sequence decomposed into impulse sequences

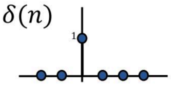
$$\{x(n)\} = \{\cdots, x(-2), x(-1), x(0), x(1), x(2), \cdots\}$$

Sometimes there is confusion because we use the symbol $x(n)$ to denote both the entire sequence $\{x(n)\}$ and a single sample $x(n)$!

$$\{x(n)\} = \sum_{k=-\infty}^{\infty} x(k)\{\delta(n-k)\} \Leftrightarrow x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k), \qquad -\infty < n < \infty$$

Convolution Description of LTI Systems

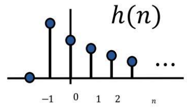
Impulse $\delta(n)$



$\delta(n)$ → LTI System → $h(n)$

$x(n)$ → LTI System → $y(n)$

$h(n)$



Impulse response

Time-invariance ⇒ $\delta(n-k) \xrightarrow{H} h(n-k)$

Homogeneity ⇒ $x(k)\delta(n-k) \xrightarrow{H} x(k)h(n-k)$

$\vdots$

$x(-1)\delta(n+1) \xrightarrow{H} x(-1)h(n+1)$

Number

$x(0)\delta(n) \xrightarrow{H} x(0)h(n)$

Sequence

$x(1)\delta(n-1) \xrightarrow{H} x(1)h(n-1)$

$\vdots$

The response $y(n)$ of an LTI system to any input $x(n)$ can be determined from the convolution summation using the impulse response sequence $h(n)$

Additivity ⇒ $x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k) \xrightarrow{H} y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k), \quad -\infty < n < \infty$

Commutative Property of Convolution

$$y(n) = \sum_k x(k)h(n-k) = x(n) * h(n)$$

Convolution operator

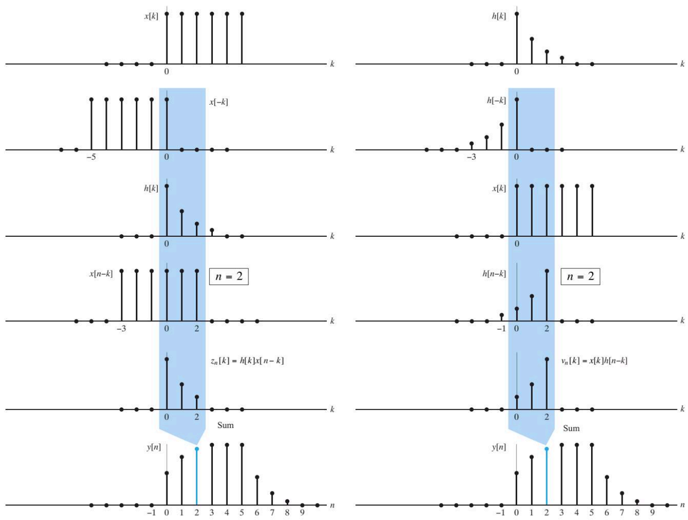
Let $m = n - k \Rightarrow k = n - m$

$$y(n) = \sum_m x(n-m)h(m) = \sum_m h(m)x(n-m) = h(n) * x(n)$$

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k), \quad -\infty < n < \infty$$

$$y(n) = h(n) * x(n) \Leftrightarrow \{y(n)\} = \{h(n)\} * \{x(n)\}$$

Graphical Illustration of Commutative Property



28

Understanding the Convolution Summation

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k), -\infty < n < \infty$$

The input sequence is flipped or reversed in time

Expand summation $\Rightarrow$

$$y(n) = \cdots + h(-1)x(n+1) + h(0)x(n) + h(1)x(n-1) + h(2)x(n-2) + \cdots$$

Compute $y(n)$ for each value of $n = \cdots, -1, 0, 1, 2, \cdots$ $\Rightarrow$

$$y(-1) = \cdots + h(-1)x(0) + h(0)x(-1) + h(1)x(-2) + h(2)x(-3) + \cdots$$

$$y(0) = \cdots + h(-1)x(1) + h(0)x(0) + h(1)x(-1) + h(2)x(-2) + \cdots$$

$$y(1) = \cdots + h(-1)x(2) + h(0)x(1) + h(1)x(0) + h(2)x(-1) + \cdots$$

$$y(2) = \cdots + h(-1)x(3) + h(0)x(2) + h(1)x(1) + h(2)x(0) + \cdots$$

To determine the output sample $y(n)$, the flipped input is shifted so that, the sample $x(n)$ is aligned with the impulse response sample $h(0)$

29

Evaluation of Convolution Sum

1. Assign value to n

4. Multiply

Single sample → $$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k)$$ ← Sequences indexed by k

5. Sum

3. Shift

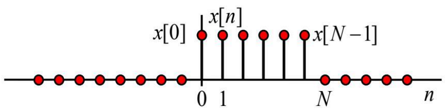
2. Flip

1. Plot the signals $x(n)$ and $h(n)$ as a function of the index k
2. Flip (fold) the signal $x(k)$ about $k = 0$ to obtain $x(-k)$
3. Shift the flipped signal $x(-k)$ to the right by n samples if $n > 0$ or to the left if $n < 0$ to obtain the sequence $x(n-k)$; n is a known fixed integer number
4. Multiply the sequences $h(k)$ and $x(n-k)$ to obtain the sequence $z_n(k) = h(k)x(n-k)$
5. Add all values of $z_n(k)$ to compute $y(n)$ for the given value of n
6. Increase n by 1 and go to step 3

EECE-5666: Digital Signal Processing Signals and Systems 30

30

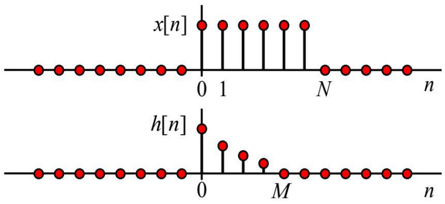
Graphical Illustration of Convolution



EECE-5666: Digital Signal Processing Signals and Systems 31

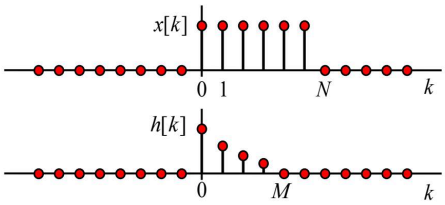
31

Graphical Illustration of Convolution



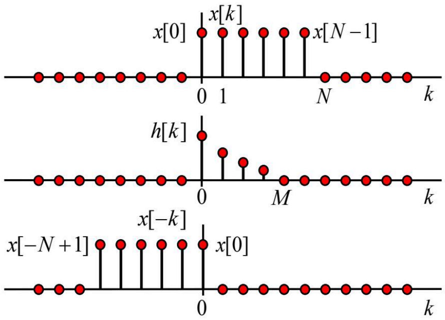
32

Graphical Illustration of Convolution



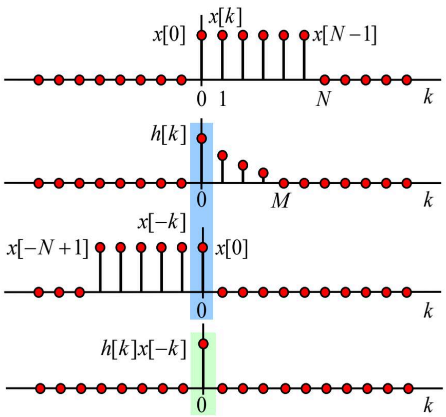
33

Graphical Illustration of Convolution



34

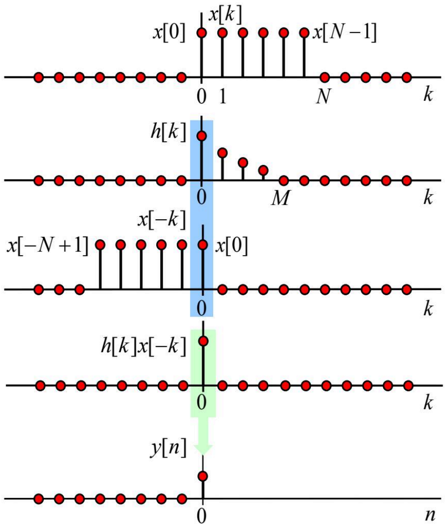
Graphical Illustration of Convolution



No Overlap for $n < 0$

35

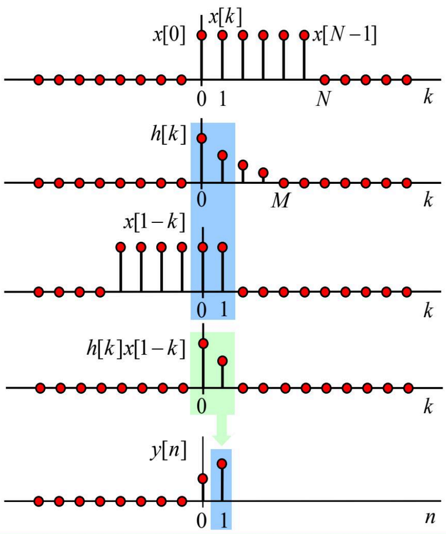
Graphical Illustration of Convolution



Partial Overlap

36

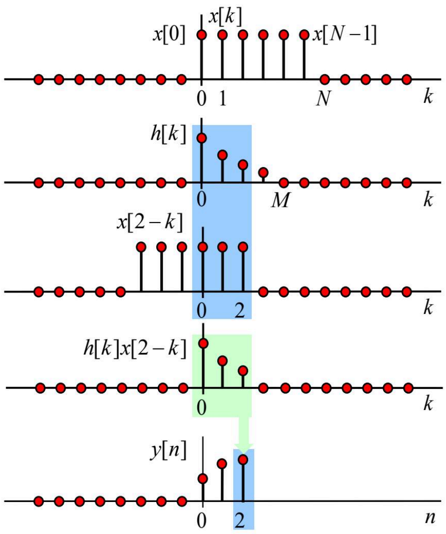
Graphical Illustration of Convolution



Partial Overlap

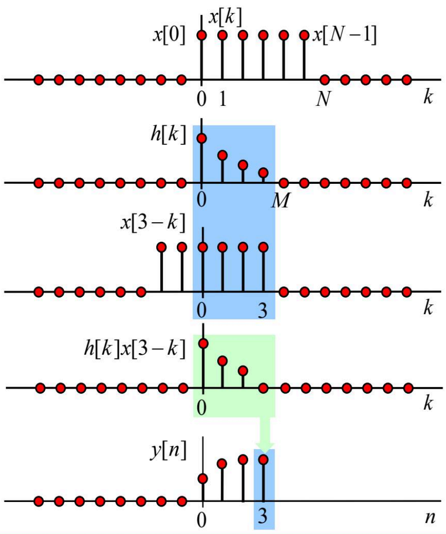
37

Graphical Illustration of Convolution



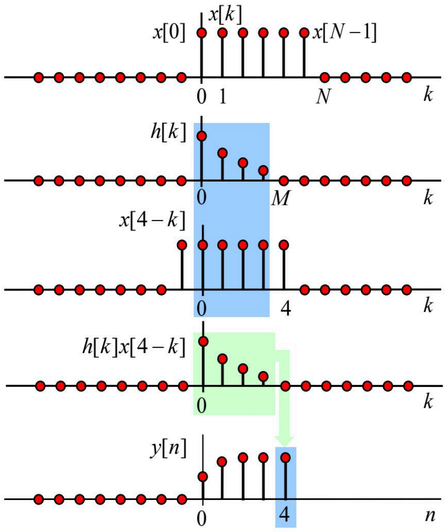
Partial Overlap

Graphical Illustration of Convolution



Full Overlap

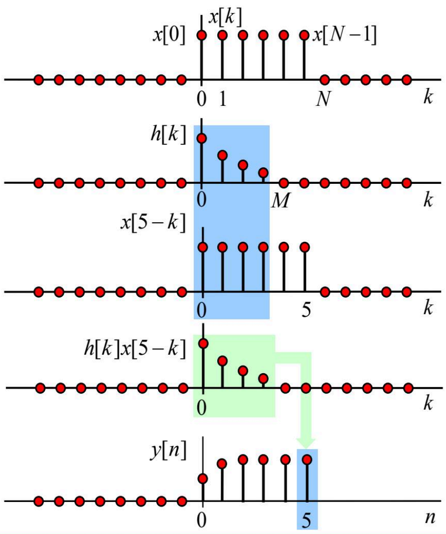
Graphical Illustration of Convolution



Full Overlap

40

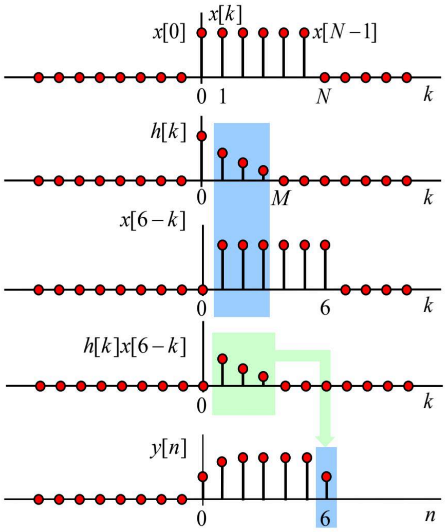
Graphical Illustration of Convolution



Full Overlap

41

Graphical Illustration of Convolution

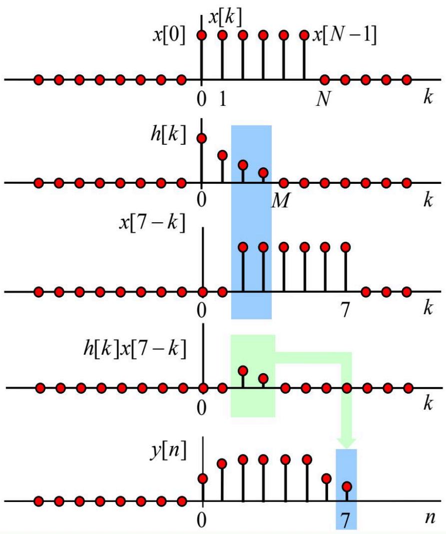


Partial Overlap

EECE-5666: Digital Signal Processing Signals and Systems 42

42

Graphical Illustration of Convolution

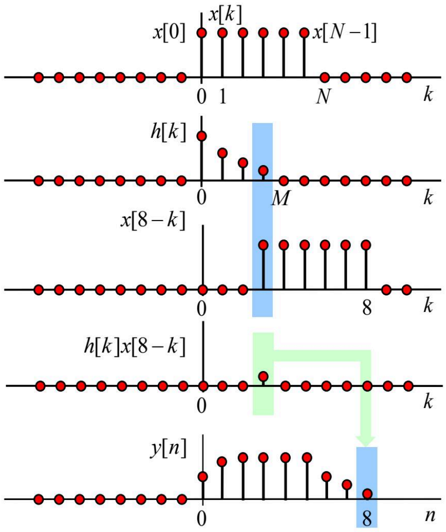


Partial Overlap

EECE-5666: Digital Signal Processing Signals and Systems 43

43

Graphical Illustration of Convolution

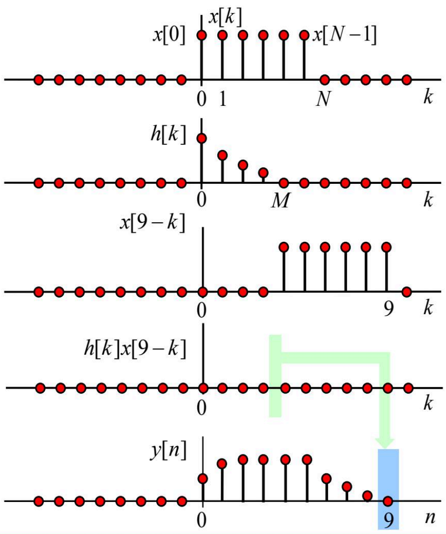


Partial Overlap

EECE-5666: Digital Signal Processing Signals and Systems 44

44

Graphical Illustration of Convolution

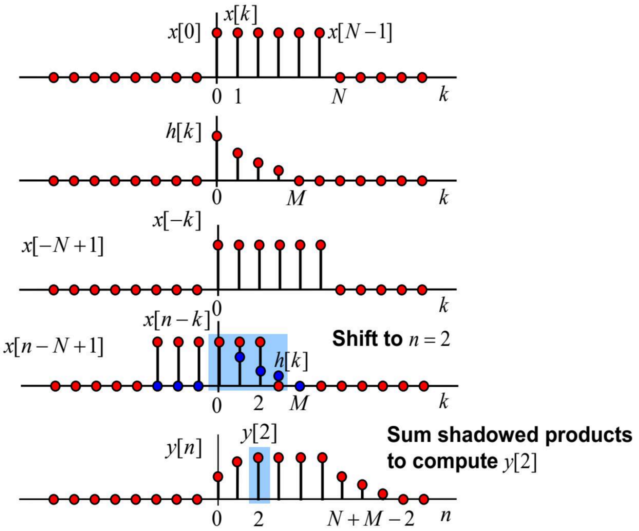


No Overlap
for $n > 0$

EECE-5666: Digital Signal Processing Signals and Systems 45

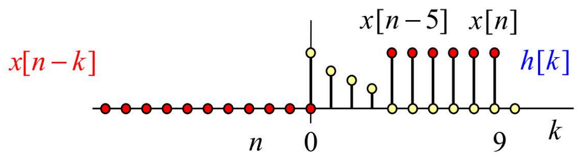
45

Summary of Graphical Illustration



46

Example of Analytical Evaluation of Convolution Sum



$n < 0 \quad \Rightarrow$ **No overlap** $\quad \Rightarrow y[n] = 0$

$0 \le n \le 2 \quad \Rightarrow$ **Partial overlap** $\quad \Rightarrow y[n] = \sum_{k=0}^{n} a^k 1 = \frac{1-a^{n+1}}{1-a}$

$3 \le n \le 5 \quad \Rightarrow$ **Full overlap** $\quad \Rightarrow y[n] = \sum_{k=0}^{3} a^k 1 = \frac{1-a^4}{1-a}$

$6 \le n \le 8 \quad \Rightarrow$ **Partial overlap** $\quad \Rightarrow y[n] = \sum_{k=n-5}^{3} a^k 1 = a^{n-5}\frac{1-a^{11-n}}{1-a}$

$n \ge 9 \quad \Rightarrow$ **No overlap** $\quad \Rightarrow y[n] = 0$

47

Numerical Evaluation of the Convolution Sum

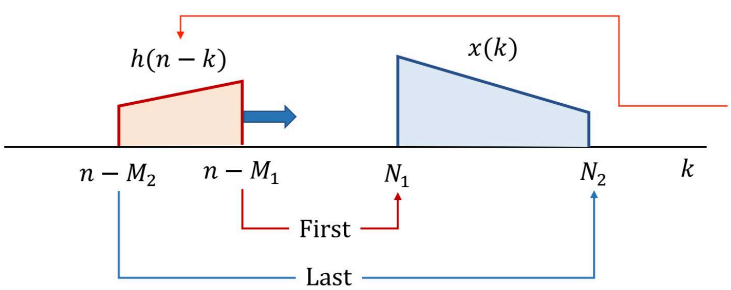
Compute $y[n]=\sum_k h[k]x[n-k]$ **given** $h[n],\ 0 \le n \le M-1$ **and** $y[n],\ 0 \le n \le N-1$

	$y[-1]=h[0]\ 0\ +h[1]\ 0\ +h[2]\ 0$	**No Overlap**
$M=3$ $n=0\rightarrow$	$y[0]=h[0]x[0]\ +h[1]\ 0\ +h[2]\ 0$	**Partial Overlap**
$N=6$	$y[1]=h[0]x[1]\ +h[1]x[0]\ +h[2]\ 0$	
	$y[2]=h[0]x[2]\ +h[1]x[1]\ +h[2]x[0]$	**Full Overlap**
	$y[3]=h[0]x[3]\ +h[1]x[2]\ +h[2]x[1]$	
	$y[4]=h[0]x[4]\ +h[1]x[3]\ +h[2]x[2]$	
$n=N-1\rightarrow$	$y[5]=h[0]x[5]\ +h[1]x[4]+h[2]x[3]$	
	$y[6]=h[0]\ 0\ +h[1]x[5]\ +h[2]x[4]$	**Partial Overlap**
$n=N+M-2\rightarrow$	$y[7]=h[0]\ 0\ +h[1]\ 0\ +h[2]x[5]$	
	$y[8]=h[0]\ 0\ +h[1]\ 0\ +h[2]\ 0$	**No Overlap**

MATLAB **function** `y = conv(x, h)` **computes** $y[n],\ 0 \le n \le N+M-2$
`y = filter(h,1,x)` **computes** $y[n],\ 0 \le n \le N-1$

48

Arbitrary Convolution Limits



$h(k) \rightarrow M_1 \le k \le M_2$
$h(-k) \rightarrow -M_2 \le k \le -M_1$
$h(n-k) \rightarrow n-M_2 \le k \le n-M_1$

First nonzero sample at:

$n-M_1=N_1 \Rightarrow n=M_1+N_1$

Last nonzero sample at:

$n-M_2=N_2 \Rightarrow n=M_2+N_2$

```
y=conv(h,x)
n=(M1+N1:M2+N2)
plot(n,y) or
stem(n,y)
```


49

LTI Systems ⇔ Convolution

Impulse decomposition $$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k)$$

Impulse response $$\delta(n-k) \xrightarrow{H} h_k(n)$$

If system is linear ⇒

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h_k(n)$$ Superposition Sum

If system is time−invariant ⇒

$$h_k(n) = h(n-k)$$

$$\text{LTI} \Rightarrow y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \triangleq x(n) * h(n)$$ Convolution Sum

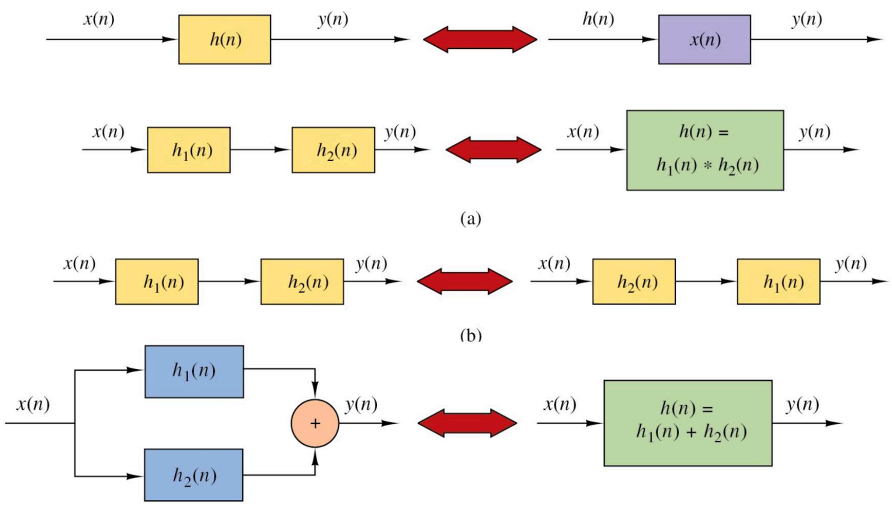
50

Properties of Convolution

Property	Formula
Identity	$x(n) * \delta(n) = x(n)$
Delay	$x(n) * \delta(n-n_0) = x(n-n_0)$
Commutative	$x(n) * h(n) = h(n) * x(n)$
Associative	$[x(n) * h_1(n)] * h_2(n) = x(n) * [h_1(n) * h_2(n)]$
Distributive	$x(n) * [h_1(n) + h_2(n)] = x(n) * h_1(n) + x(n) * h_2(n)$
Area Property	$A_y = \sum_{n=-\infty}^{\infty} y(n) = \sum_{k=-\infty}^{\infty} h(k) \sum_{n=-\infty}^{\infty} x(n-k) = A_h A_x$

51

Implications of Convolution Properties



52

Causality of LTI Systems

An LTI system with impulse response $h(n)$ is causal iff $h(n) = 0$ for $n < 0$

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) =$$

Future Present Past

$$\cdots + h(-2)x(n+2) + h(-1)x(n+1) + h(0)x(n) + h(1)x(n-1) + h(2)x(n-2) + \cdots$$

- Causality is important for real-time applications
- A finite number of “future” samples in $h(n)$ can be tolerated by delaying the output
- Causality is not a problem for non real-time (stored signals) applications

53

Stability of LTI Systems

An LTI system with impulse response $h(n)$ is BIBO stable iff $h(n)$ is absolutely summable, i.e.,

$$S_h = \sum_{n=-\infty}^{\infty} |h(n)| < \infty$$

Sufficient Assume that $|x(n)| \leq M_x < \infty$ (bounded) for all n. Then we have

$$|y(n)| = \left| \sum_{k=-\infty}^{\infty} h(k)x(n-k) \right| \leq \sum_{k=-\infty}^{\infty} |h(k)||x(n-k)| \leq M_x \sum_{k=-\infty}^{\infty} |h(k)| = M_x S_h$$

If $S_h < \infty \Rightarrow |y(n)| \leq M_x S_h = \text{bounded}$

Necessary Consider a bounded input signal defined by $x(n) = \text{sign}(h(n))$

The response $y(n)$ at $n = 0$ to the input $x(n)$ is $y(0) = \sum_{k=-\infty}^{\infty} h(k)x(-k) = \sum_{k=-\infty}^{\infty} |h(k)| = S_h$

A bounded input leads to an unbounded response if $S_h = \infty$

Note that $h(n)$ cannot be periodic and practical systems must be stable

Examples

Determine whether the following LTI systems are stable

$$h(n) = a^n u(n) \Rightarrow S_h = \sum_{n=-\infty}^{\infty} |h(n)| = \sum_{n=0}^{\infty} |a|^n = \frac{1}{1-|a|} \Rightarrow \text{Stable iff } |a| < 1$$

$$h(n) = \cos(\omega_0 n)\, u(n) \Rightarrow S_h = \sum_{n=-\infty}^{\infty} |h(n)| = \sum_{n=0}^{\infty} |\cos(\omega_0 n)| = \infty \Rightarrow \text{Unstable}$$

$$y(n) = \frac{1}{M+1} \sum_{k=0}^{M} x(n-k) \Rightarrow h(n) = \begin{cases} \dfrac{1}{M+1}, & 0 \leq n \leq M \\ 0, & \text{elsewhere} \end{cases}$$

$$S_h = \sum_{n=0}^{M} |h(n)| = 1 \Rightarrow \text{Stable}$$

All FIR systems are stable!

Summary

- Any Linear and Time-Invariant (LTI) system can be completely described by its impulse response sequence defined by

$$\delta(n) \xrightarrow{H} h(n)$$

- The output of any LTI can be determined using the convolution summation

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) = h(n) * x(n)$$

- The impulse response provides the basis for the analysis of an LTI system in the time-domain
  - Causality ($x(n) = 0$ for $n < n_0 \Rightarrow y(n) = 0$ for $n < n\_0$)$\Leftrightarrow h(n) = 0$ for $n < 0$
  - Stability (Bounded Input $\Rightarrow$ Bounded Output)$\Leftrightarrow \sum_{n=-\infty}^{\infty} |h(n)| < \infty$
- Implementation requires only delays (memory), multiplications, and additions

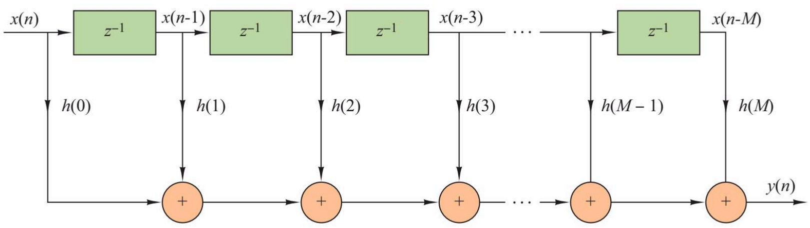
56

FIR and IIR Systems

$$y(n) = \sum_{k=M_1}^{M_2} h(k)x(n-k)$$

- M_1 and M_2 finite $\Rightarrow$ Finite (Duration) Impulse Response (FIR) System — Always stable
- Otherwise $\Rightarrow$ Infinite (Duration) Impulse Response(IIR) System — Not always stable

Causal FIR System



$$y(n) = \sum_{k=0}^{M} h(k)x(n-k) = h(0)x(n) + h(1)x(n-1) + \cdots + h(M)x(n-M)$$

Are there any practically realizable IIR systems?

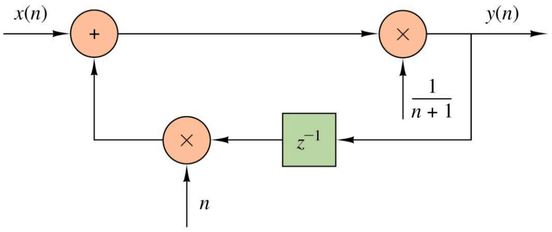
57

Growing Memory Signal Averaging System

Linear, time-varying, non-recursive

$$y(n) = \frac{1}{n+1}\sum_{k=0}^{n} x(k), \qquad n \geq 0$$

$$y(n) = \frac{1}{n+1}\sum_{k=0}^{n-1} x(k) + \frac{1}{n+1}x(n)$$



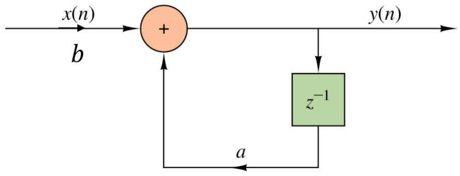
$$y(n) = \frac{n}{n+1}y(n-1) + \frac{1}{n+1}x(n)$$

Linear, time-varying, recursive

$$y(n) = a(n)y(n-1) + b(n)x(n)$$

Time-invariant

$$y(n) = ay(n-1) + bx(n)$$

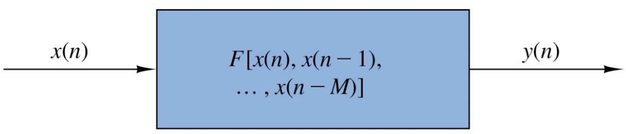


- Is the system linear?
- Can the system be represented by a convolution sum?
- Is the system stable?

58

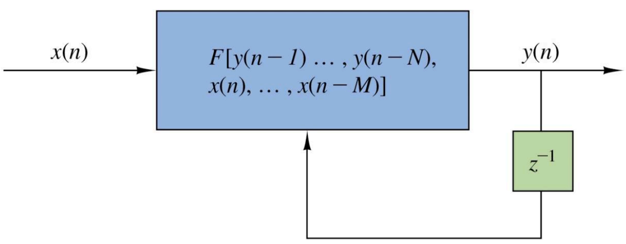
Basic Forms for Casual and Realizable Systems

Non-recursive systems



(a)

Recursive systems



Unit delay required to ensure realizability

(b)

59

Impulse Response of First-Order System

First-order recursive system

$y(n) = ay(n-1) + bx(n)$

Definition of impulse response

$x(n) = \delta(n) \xrightarrow{H} y(n) = h(n)$

The input starts at $n = 0 \Rightarrow$

$y(0) = ay(-1) + bx(0) = b$ Initial condition: $y(-1) = 0$

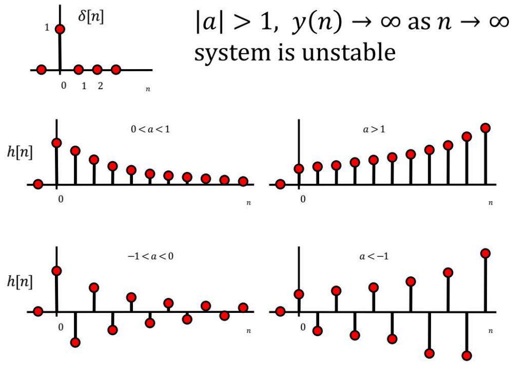
$y(1) = ay(0) + bx(1) = ba$

$y(2) = ay(1) + bx(2) = ba^2$

$\vdots$

$y(n) = ay(n-1) + bx(n) = ba^n$

$h(n) = ba^n u(n)$



$|a| > 1,\ y(n) \to \infty$ as $n \to \infty \Rightarrow$ system is unstable

Definition of recursive systems requires:

Input-output equation:
$y(n) = ay(n-1) + bx(n), \quad n \geq 0$
Initial condition: $y(-1) = 0$

EECE-5666: Digital Signal Processing Signals and Systems 60

60

Response to a Suddenly Applied Input

$y(0) = ay(-1) + bx(0)$
$y(1) = ay(0) + bx(1) = a^2y(-1) + bax(0) + bx(1)$
$y(2) = ay(1) + bx(2) = a^3y(-1) + ba^2x(0) + bax(1) + bx(2)$
$\vdots$
$y(n) = ay(n-1) + bx(n) = a^{n+1}y(-1) + ba^n x(0) + ba^{n-1}x(1) + \cdots + bx(n)$

$h(n) = ba^n u(n) \Rightarrow$
$y(n) = a^{n+1}y(-1) + h(n)x(0) + h(n-1)x(1) + \cdots + h(0)x(n)$

$$y(n) = \sum_{k=0}^{n} h(k)x(n-k) + a^{n+1}y(-1) = y_{zs}(n) + y_{zi}(n)$$

Zero-state response Zero-input response

- Initially-at-rest system: If $y(-1) = 0 \Rightarrow$ $y(n) = \sum_{k=0}^{n} h(k)x(n-k) \Rightarrow$ LTI
- Stability: Iff $|a| < 1 \Rightarrow$ $\sum_{n=-\infty}^{\infty}|h(n)| = |b|\sum_{n=-\infty}^{\infty}|a|^n = \frac{|b|}{1-|a|}$
- If $y(-1) \neq 0 \Rightarrow$ System is linear in a more general sense (Affine)

EECE-5666: Digital Signal Processing Signals and Systems 61

61

Example of Unit Step Response

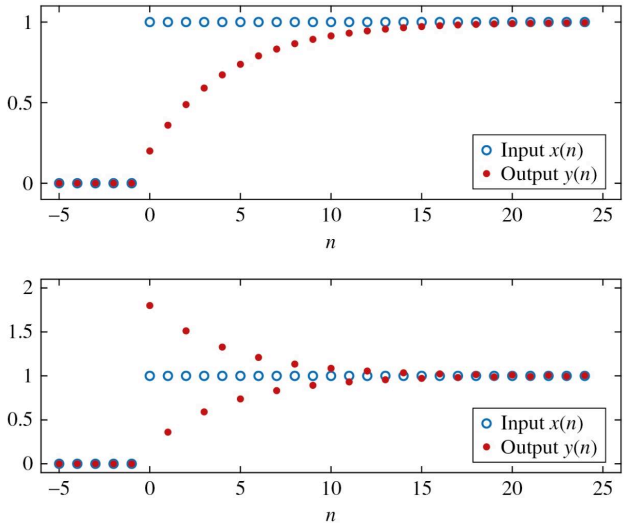


Illustration of the transient and steady-state unit-step response of a first-order recursive system with a = 0.8 (top) and a = −0.8 (bottom).

EECE-5666: Digital Signal Processing Signals and Systems 62

62

Response to a Suddenly Applied Complex Exponential

$y(n) = ay(n-1) + bx(n), \quad n \geq 0 \Rightarrow$ If $x(n) = \mathrm{e}^{\mathrm{j}\omega_0 n}u(n) \Rightarrow$

$$y(n) = a^{n+1}y(-1) + \sum_{k=0}^{n} a^k x(n-k) \qquad y(n) = a^{n+1}y(-1) + \mathrm{e}^{\mathrm{j}\omega_0 n}\sum_{k=0}^{n}\left(a\mathrm{e}^{-\mathrm{j}\omega_0}\right)^k \Rightarrow$$

Fourier Transform

$$y(n) = \overbrace{a^{n+1}y(-1) + \frac{-a^{n+1}\mathrm{e}^{-\mathrm{j}\omega_0(n+1)}}{1-a\mathrm{e}^{-\mathrm{j}\omega_0}}e^{j\omega_0 n}}^{\text{Transient response } y_{\mathrm{tr}}(n)} + \overbrace{\frac{1}{1-a\mathrm{e}^{-\mathrm{j}\omega_0}}\mathrm{e}^{\mathrm{j}\omega_0}}^{\text{Steady-state response } y_{\mathrm{ss}}(n)}$$

Main interest in DSP applications

Zero-input response $y_{\mathrm{zi}}(n)$ Zero-state response $y_{\mathrm{zs}}(n)$

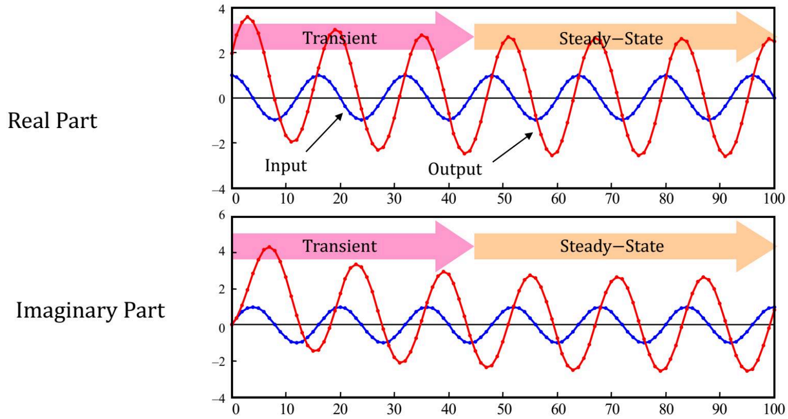
Convolution sum
z-Transform

Difference Equation, One-Sided z-Transform

EECE-5666: Digital Signal Processing Signals and Systems 63

63

Transient versus Steady-State Response



- Steady−state response: "Tracks" the input signal
- Transient−response: "Reveals" properties of the system, but eventually "dies out" for a BIBO stable system

General Recursive Systems

Consider the linear constant-coefficient difference equation

$$y(n) = -\sum_{k=1}^{N} \underbrace{a_k}_{\text{Feedback coefficients}} y(n-k) + \sum_{k=0}^{M} \underbrace{b_k}_{\text{Feed-forward coefficients}} x(n-k)$$

- Prove that the system is LTI
- Determine analytically the impulse response of the system
- Given an analytical expression for the input $x(n)$ find an analytical expression for the output $y(n)$ of the system
- Given the coefficients $\{a_k, b_k\}$ determine if the system is stable

To answer these questions, we need the z-transform

Computation of Difference Equations

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

Important sign convention!

$$y = \text{filter}(b, a, x)$$
$$b = [b_0 \; b_1 \cdots b_M]$$
$$a = [1 \;\; a_1 \;\; \cdots \;\; a_N]$$
$$x = [x(0) \;\; x(1) \;\; \cdots \;\; x(L-1)]$$
$$y = [y(1) \;\; y(1) \;\; \cdots \;\; y(L-1)]$$

- The case for non-zero initial conditions is discussed in a subsequent lecture
- y = filter(h,1,x) computes the convolution in the range of vector x

66

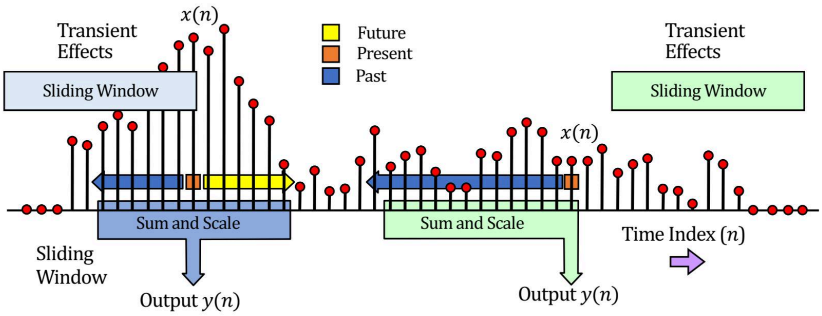
Moving Average Systems

Non-Causal Moving Average

$$y(n) = \frac{1}{2L+1} \sum_{k=-L}^{L} x(n-k)$$

Causal Moving Average

$$y(n) = \frac{1}{2L+1} \sum_{k=0}^{2L} x(n-k)$$



67

Moving Average Smoothing

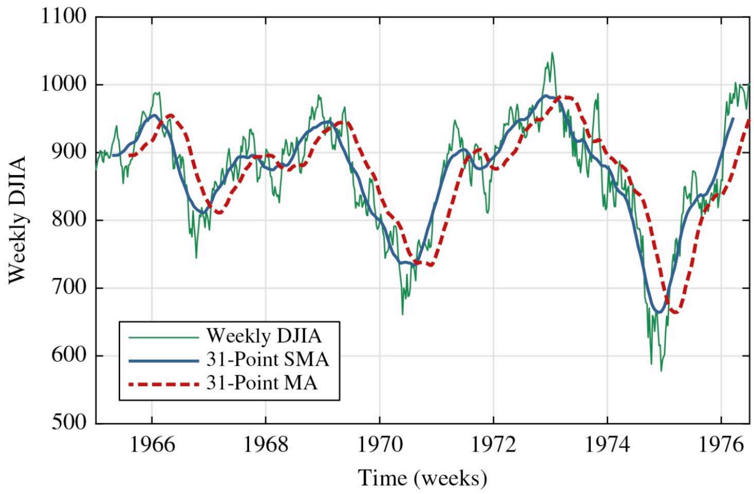
Causal Moving Average (MA)

$$y(n) = \frac{1}{M}\sum_{k=0}^{M-1} x(n-k)$$

$$M = 2L + 1$$

Symmetric or centered MA

$$y(n) = \frac{1}{2L+1}\sum_{k=-L}^{L} x(n-k)$$



68

Exponential Smoothing

$y(n) = ay(n-1) + bx(n), n \geq 0$

$h(n) = ba^n u(n), \quad 0 < a < 1$

$\sum_n y(n) = \sum_n h(n) \sum_n x(n)$

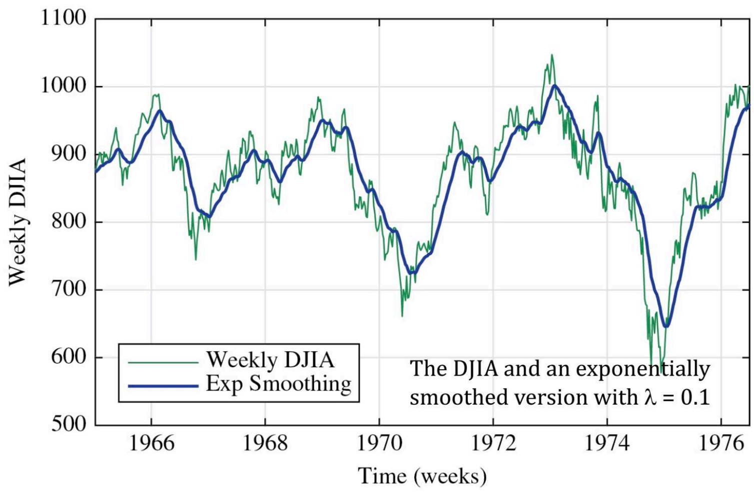
Area-based gain normalization

$$\sum_n h(n) = \frac{b}{1-a} = 1 \Rightarrow b = 1 - a$$

$y(n) = ay(n-1) + (1-a)x(n)$

Alternative form ($\lambda = 1 - a$)

$y(n) = \lambda y(n-1) + (1-\lambda)x(n)$



69

Summary

- Any LTI system with zero initial conditions can be described by a convolution

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k)$$

- The impulse response sequence $h(n)$ is defined by $\delta(n) \xrightarrow{H} h(n)$
- To compute $h(n)$, set the initial conditions to zero, $x(0) = 1, x(n) = 0$ for $n > 0$, and compute $\mathrm{y}(n) = h(n)$
- An LTI system is stable iff $\sum_{n=-\infty}^{\infty} |h(n)| =$ finite and causal iff $h(n) = 0$ for $n < 0$

- Consider a system described by the linear constant-coefficient difference equation

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

- In general, if at least one $a_k \neq 0$, the convolution summation can compute the output of the system only approximately

EECE-5666: Digital Signal Processing

Discrete-Time Signals and Systems