Lecture I213E – Class 5

Discrete Signal Processing

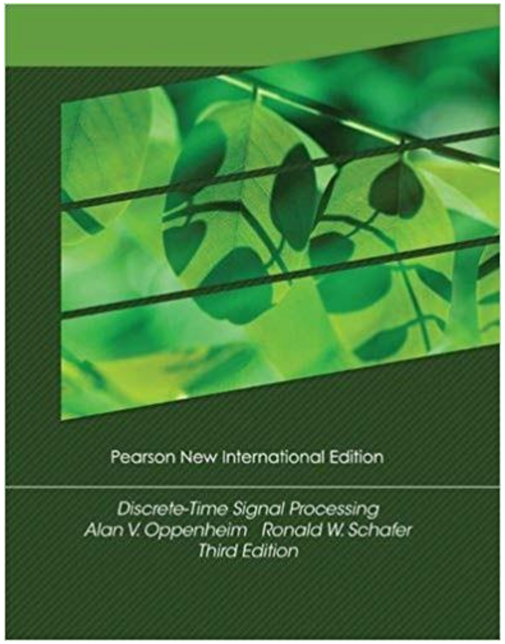
Sakriani Sakti

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY
1 9 9 0

Course Materials

- ## Materials

→ Lecture notes will be uploaded before each lecture
https://jstorage-2018.jaist.ac.jp/s/PGXRrC7iFmN2FWo
Pass: dsp-i213e-2022
(Slide Courtesy of Prof. Nak Young Chong)



- ## References

→ Chi-Tsong Chen:
Linear System Theory and Design, 4th Ed.,
Oxford University Press, 2013.

→ Alan V. Oppenheim and Ronald W. Schafer:
Discrete-Time Signal Processing, 3rd Ed.,
Pearson New International Ed., 2013.

Related Courses & Prerequisite

- **Related Courses**
  - → I212 Analysis for Information Science
  - → I114 Fundamental Mathematics for Information Science

- **Prerequisite**
  - → None

Evaluation

- **Viewpoint of evaluation**

  → Students are able to understand:

  - Basic principles in modeling and analysis of linear time-invariant systems
  - Applications of mathematical methods and tools to different signal processing problems.

- **Evaluation method**

  → Homework, term project, midterm exam, and final exam

- **Evaluation criteria**

  → Homework/labs (30%), term project (30%) midterm exam (15%), and final exam (25%)

Contact

- ## Lecturer

  → Sakriani Sakti

- ## TA

  Tutorial hours & Term project

  → WANG Lijun (s2010026)

  → TANG Bowen (s2110411)

  Homework

  → PUTRI Fanda Yuliana (s2110425)

- ## Contact Email

  → dsp-i213e-2022@ml.jaist.ac.jp

Schedule

- **December 8th, 2022 – February 9th, 2023**

- **Lecture Course Term 2-2**
  - → Tuesday 9:00 — 10:40
  - → Thursday 10:50 — 12:30

- **Tutorial Hours**
  - → Tuesday 13:30-15:10

Schedule

Dec

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Jan

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	~~5~~	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	~~24~~	25	26	27	28
29	30	31				

Feb

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

Lecture:
Tuesday 9:00 — 10:40
Thursday 10:50 — 12:30

Tutorial:
Tuesday 13:30 — 15:10

Course review &
term project evaluation
(on tutorial hours)

Midterm & final exam
Thursday 10:50 — 12:30

Syllabus

Class	Date	Lecture Course Tue 9:00 — 10:40 / Thr 10:50 — 12:30	Tutorial Hours Tue 13:30 — 15:10
1	12/08	Introduction to Linear Systems with Applications to Signal Processing	
2	12/13	State Space Description	○
3	12/15	Linear Algebra	
4	12/20	Quantitative Analysis (State Space Solutions) and Qualitative Analysis (Stability)	○
5	12/22	Discrete-time Signals and Systems	
X	01/05		
6	01/10	Discrete-time Fourier Analysis	
7	01/10*	Review of Discrete-time Linear Time-Invariant Signals and Systems (on Tutorial Hours)	
	01/12	Midterm Exam	
8	01/17	Sampling and Reconstruction of Analog Signals	○
9	01/19	z-Transform	
X	01/24		○
10	01/26	Discrete Fourier Transform	
11	01/31	FFT Algorithms	○
12	01/02	Implementation of Digital Filters	
13	02/07	Digital Signal Processors and Design of Digital Filters	
14	02/07*	Review of the Course and Term Project Evaluation (on Tutorial Hours)	
	02/09	Final exam	

Class 5
Discrete-time
Signals and Systems

Discrete-time Signals and Systems

- *A number of important types of signals and their operations*
- *Linear and shift-invariant systems*
- *The convolution and the difference equation representations*

- *The emphasis is on the representations and implementation of signals and systems using* MATLAB

Signals

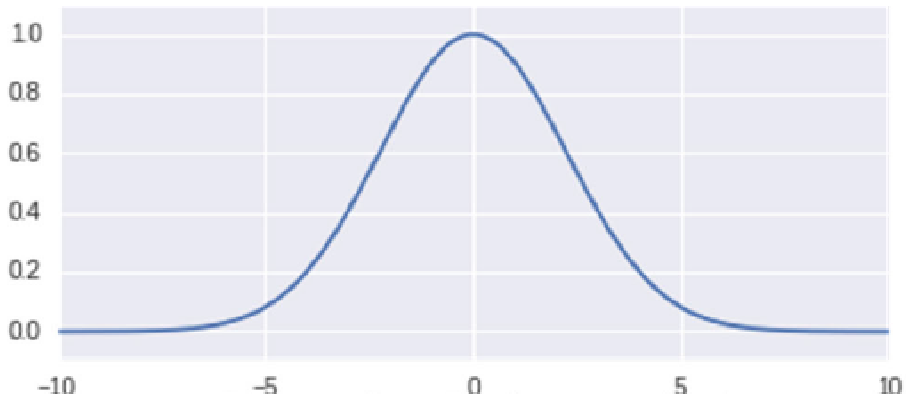
Classification of Signals

- **Classification**

1. Continuous-time vs. Discrete-time Signals
2. Multichannel and Multidimensional Signals
3. Deterministic vs. Non-deterministic (Random) Signals
4. Causality: Causal, Anti-Causal, and Non-causal Signals
5. Symmetric: Even and Odd Signals
6. Periodicity: Periodic and Aperiodic Signals
7. Energy and Power Signals

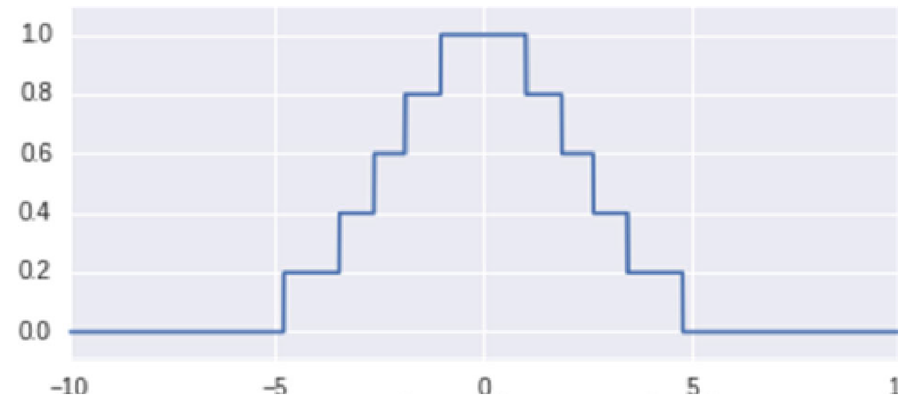
Continuous-time & Discrete-time Signals

Continuous-time Continuous-valued

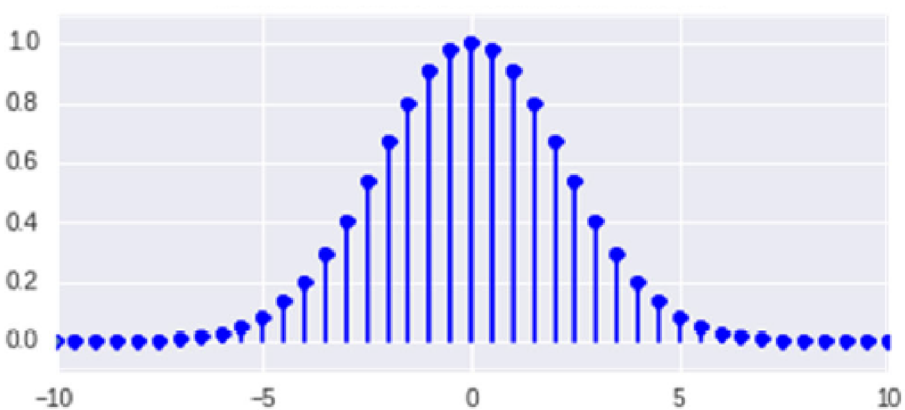


→ **Analog signals**

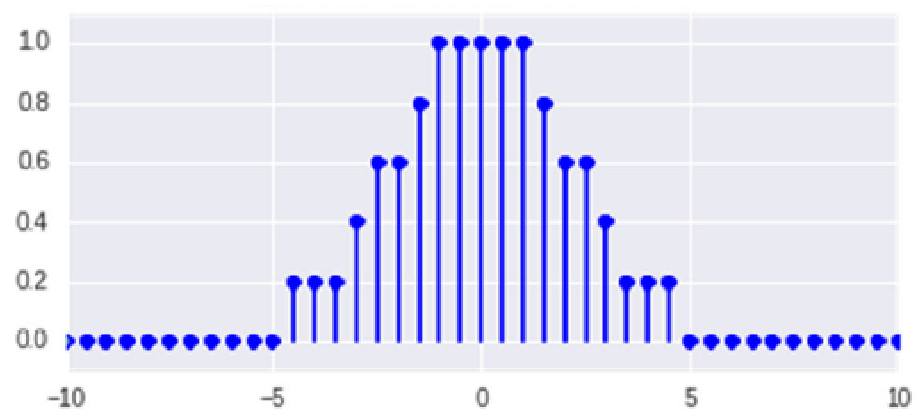
Continuous-time Discrete-valued



Discrete-time Continuous-valued



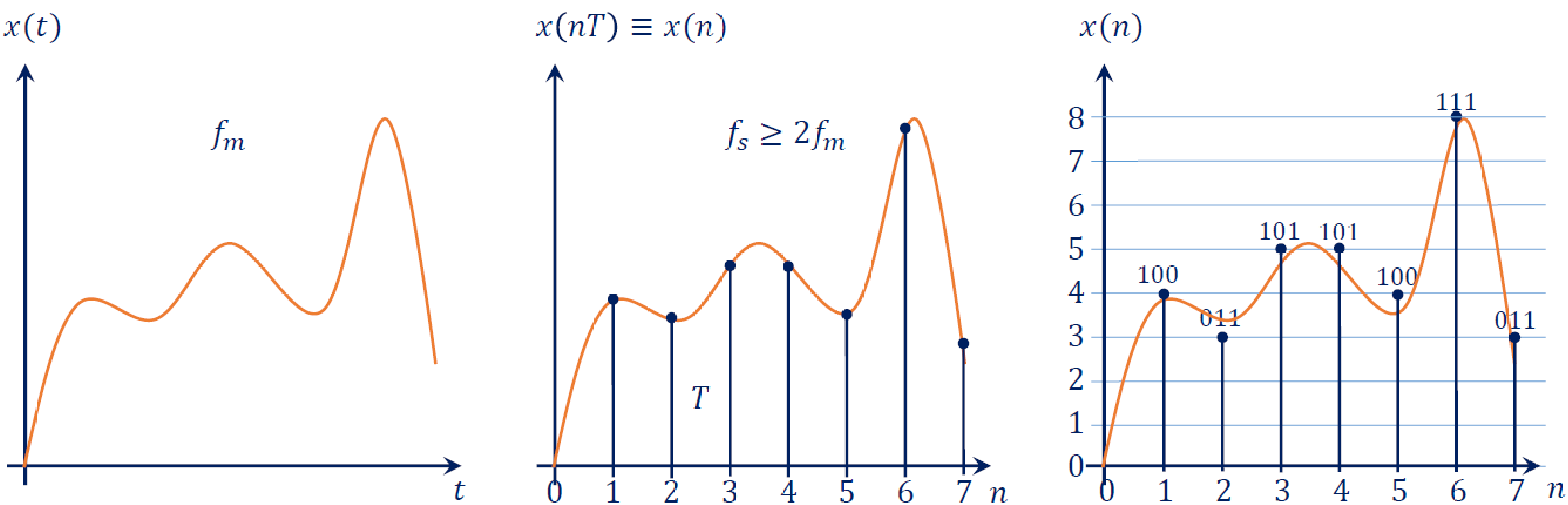
Discrete-time Discrete-valued



→ **Digital signals**

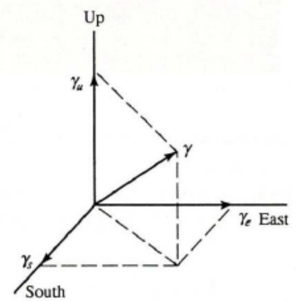
Continuous-time & Discrete-time Signals

- **Mathematical Description**

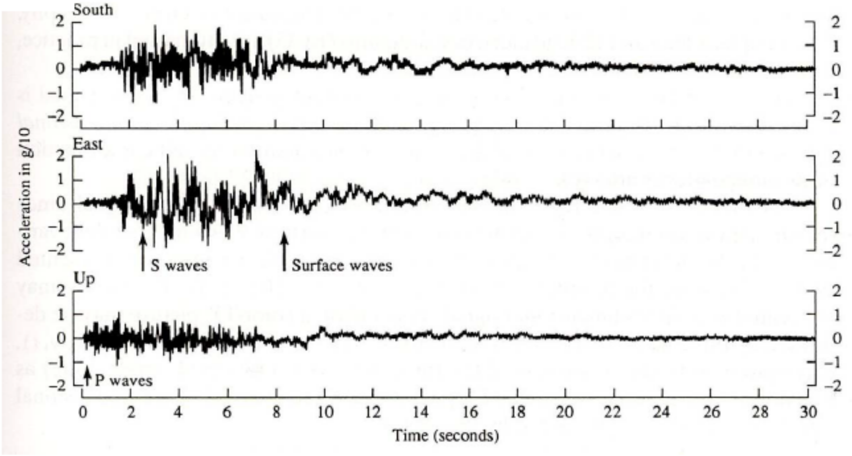


Multichannel & Multidimensional Signals

- **Multichannel Signals**
  - → Signals which are generated by multiple sources or multiple sensors are called multichannel signals
  - → Example:



$$S_3(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \end{bmatrix}$$



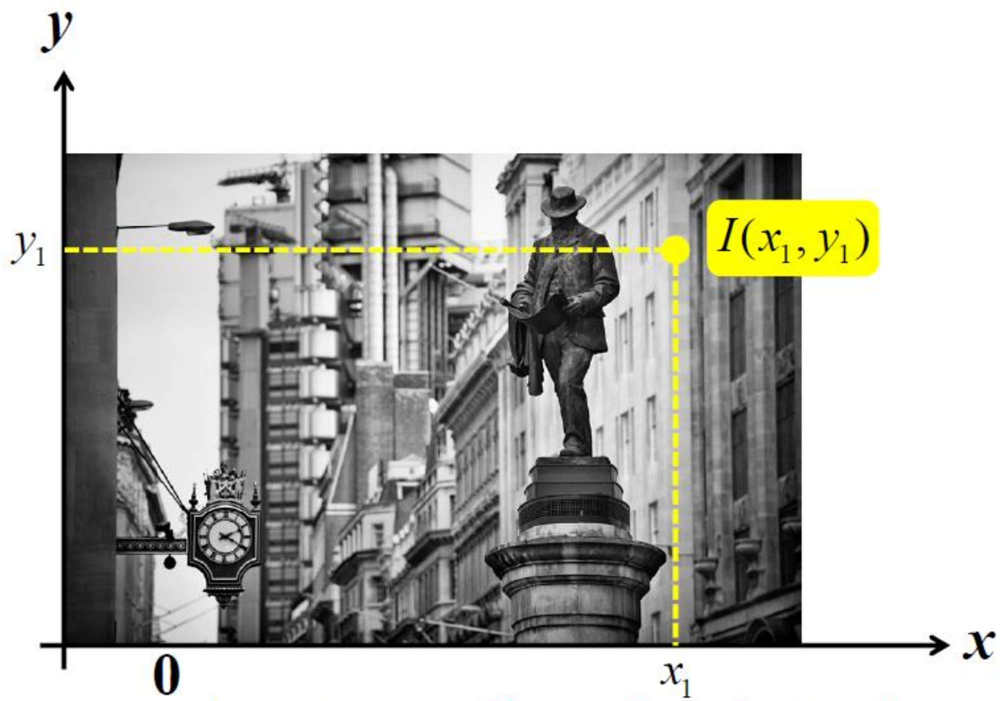
ground acceleration due to an earthquake

Multichannel & Multidimensional Signals

▪ Multidimensional Signals

→ A signal is called multidimensional signal if it is a function of M independent variables (generated by as a single source)

→ Example:



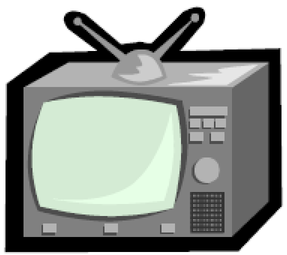
example of a two-dimensional signal: the intensity or brightness $I(x, y)$ at each point is a function of two-independent variables.

Multichannel & Multidimensional Signals

▪ Multichannel & multidimensional Signals

→ Example:

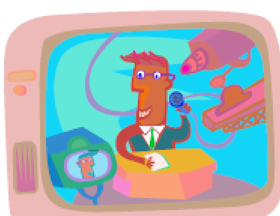
A black-and-white TV picture:



$$I(x, y, t)$$

a three-dimensional signal

A color TV picture:



$$I(x, y, t) = \begin{bmatrix} I_r(x, y, t) \\ I_g(x, y, t) \\ I_b(x, y, t) \end{bmatrix}$$

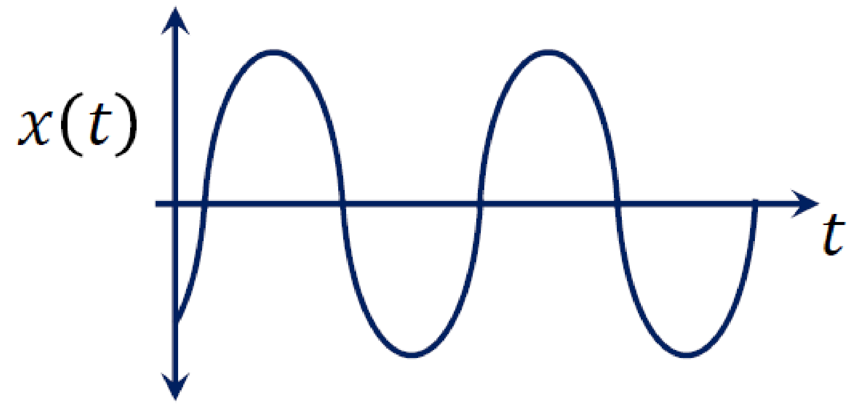
a three-channel, three-dimensional signal

Deterministic & Random Signals

Deterministic Signals

→ Can be uniquely described by an explicit mathematical expression, a table of data, or a well-defined rule

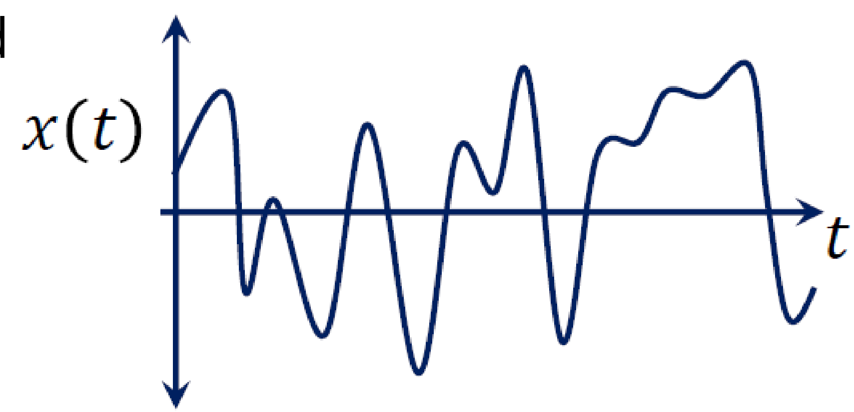
→ All past, present, and future values of the signal are known precisely, without any uncertainty



Random Signals

→ Evolves in time in an unpredictable manner.

→ Random signals can only be expressed as a function of probability based variables. For example, speech or noise signals.



Causal, Anti-causal & Non-causal Signals

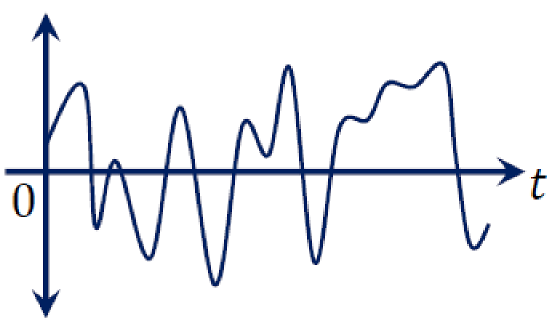
- Term causality is usually used to characterize systems. However, on the same lines signal can also be classified as causal, non-causal or anti-causal signals.

Signal $x(t)$ is a

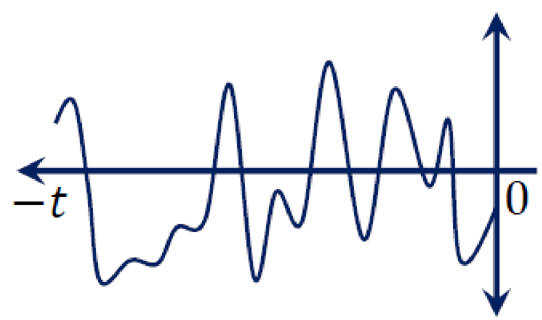
(a) Causal signal, if $|x(t)| > 0$ for $t \geq 0$.

(b) Anti-causal signal if $|x(t)| > 0$ for $t < 0$.

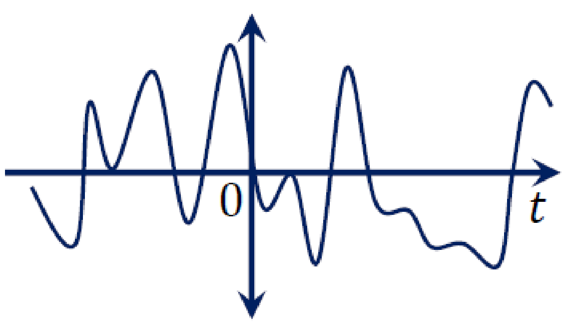
(c) Non-causal signal if $|x(t)| > 0$ for all $t \geq 0$ and $t < 0$.



Causal Signal



Anti-Causal Signal

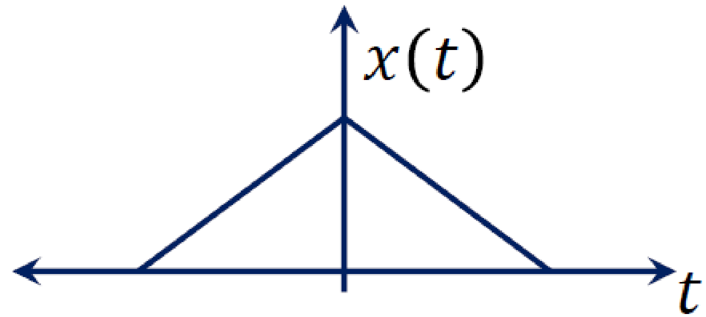


Non-Causal Signal

Symmetric: Even & Odd Signals

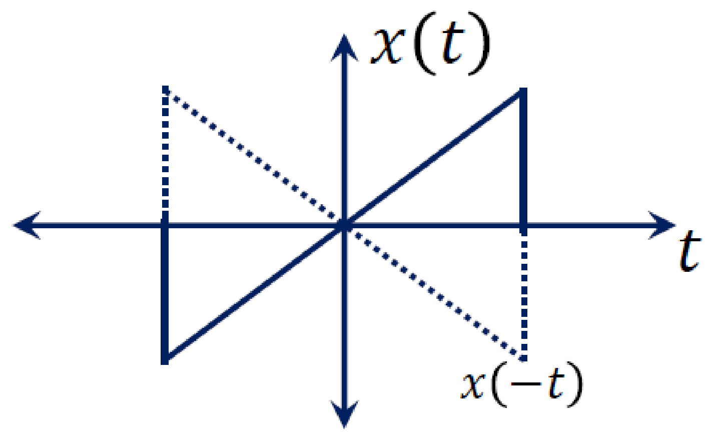
- **Even Signal**: A signal $x(t)$ is said to be an even signal, if $x(t) = x(-t)$

  For Discrete-Time even signals
  $$x(n) = x(-n)$$



- **Odd Signal**: The signal $x(t)$ is termed as an odd signal, if $x(t) = -x(-t)$

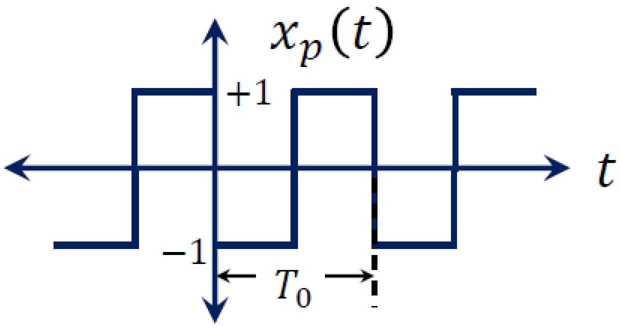
  For Discrete-Time odd signals
  $$x(n) = -x(-n)$$



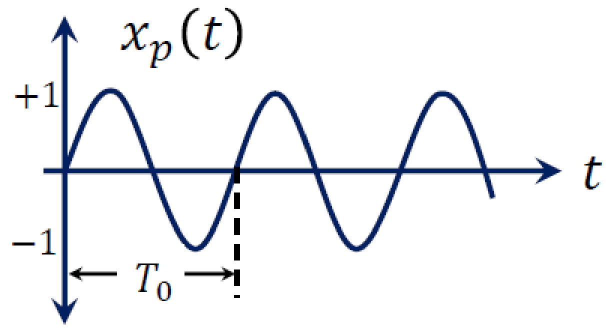
Periodicity: Periodic & Aperiodic Signals

A signal $x(t)$ is said to be a periodic if it repeats after a finite time interval T_0. Mathematically, $x(t) = x(t + T_0)$. T_0 is called fundamental time period and $f_0 = \frac{1}{T_0}$ is called its fundamental frequency.

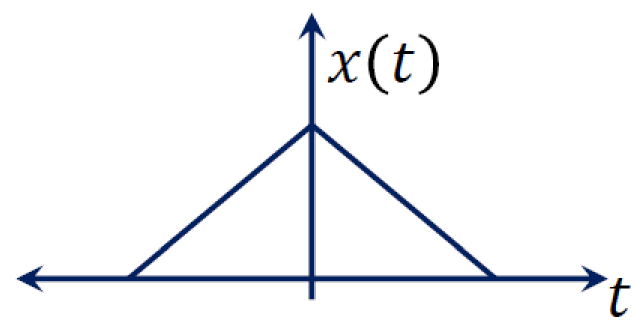
An **aperiodic** or **non-periodic** signal repeats after infinity time, i.e. $T_0 \to \infty$.



Periodic



Periodic



Aperiodic

Energy & Power Signals

Energy Signals

The energy in a **Continuous-Time** signal $x(t)$ is given as

$$E_x = \int_{-\infty}^{\infty} |x^2(t)|dt$$

If $0 < E_x < \infty$, the signal is termed as an Energy Signal.

$\boxed{\text{Signals that are both deterministic and nonperiodic are energy signals.}}$

Power Signals

If the energy of a signal is infinite then we consider its average power given as

$$P_x = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} |x^2(t)|dt$$

If $0 < P_x < \infty$, the signal is termed as a Power Signal.

$\boxed{\text{Periodic and random signals are power signals.}}$

Energy & Power Signals

- Instantaneous signal power

$$P_x(t) = |x(t)|^2, \qquad P_x[k] = |x[k]|^2$$

- Signal energy

$$\varepsilon_x(t_0, t_1) = \int_{t_0}^{t_1} |x(t)|^2 \, dt, \qquad \varepsilon_x(k_0, k_1) = \sum_{k_0}^{k_1} |x[k]|^2$$

- Average signal power

$$P_x(t_0, t_1) = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} |x(t)|^2 \, dt, \quad P_x(k_0, k_1) = \frac{1}{k_1 - k_0 + 1} \sum_{k_0}^{k_1} |x[k]|^2$$

Energy & Power Signals

An energy signal has zero power.

$$P_{x,\infty} = \lim_{T\to\infty} \frac{1}{2T} \underbrace{\int_{-T}^{T} |x(t)|^2 \, dt}_{\to \varepsilon_{x,\infty} < \infty}$$

$$= 0$$

A power signal has infinite energy.

$$\varepsilon_{x,\infty} = \lim_{T\to\infty} \int_{-T}^{T} |x(t)|^2 \, dt$$

$$= \lim_{T\to\infty} 2T \underbrace{\frac{1}{2T} \int_{-T}^{T} |x(t)|^2 \, dt}_{\to P_{x,\infty} > 0}$$

$$= \infty$$

Energy & Power Signals

- **Example:**

$$x(k) = \left(\frac{1}{4}\right)^k u(k)$$

$$\varepsilon_x = \sum_{k=-\infty}^{\infty} |x(k)|^2 = \sum_{k=0}^{\infty} \left| \left(\frac{1}{4}\right)^k \right|^2 = \sum_{k=0}^{\infty} \left(\frac{1}{16}\right)^k$$

$$= \frac{1}{1-\frac{1}{16}} = \frac{16}{15}$$

$$P_x = 0$$

$$x(k) = e^{j10k} u(k)$$

$$P_x = \lim_{K\to\infty} \left(\frac{1}{2K+1} \sum_{k=-K}^{K} |x(k)|^2 \right)$$

$$= \lim_{K\to\infty} \left(\frac{1}{2K+1} \sum_{k=0}^{K} 1_k^2 \right)$$

$$= \lim_{K\to\infty} \left(\frac{K+1}{2K+1} \right) = \frac{1}{2}$$

$$\varepsilon_x = \infty$$

Energy & Power Signals

Geometry Series

The geometric series *A one-sided exponential sequence*

$$\sum_{k=0}^{\infty} \alpha^k \to \frac{1}{1-\alpha}, \quad for\ |\alpha| < 1$$

an arbitrary constant

$$\sum_{k=0}^{N-1} \alpha^k = \frac{1-\alpha^N}{1-\alpha}, \forall \alpha$$

Absolute Value of Complex Exponential

$$e^{xi} = \cos(x) + i\sin(x)$$

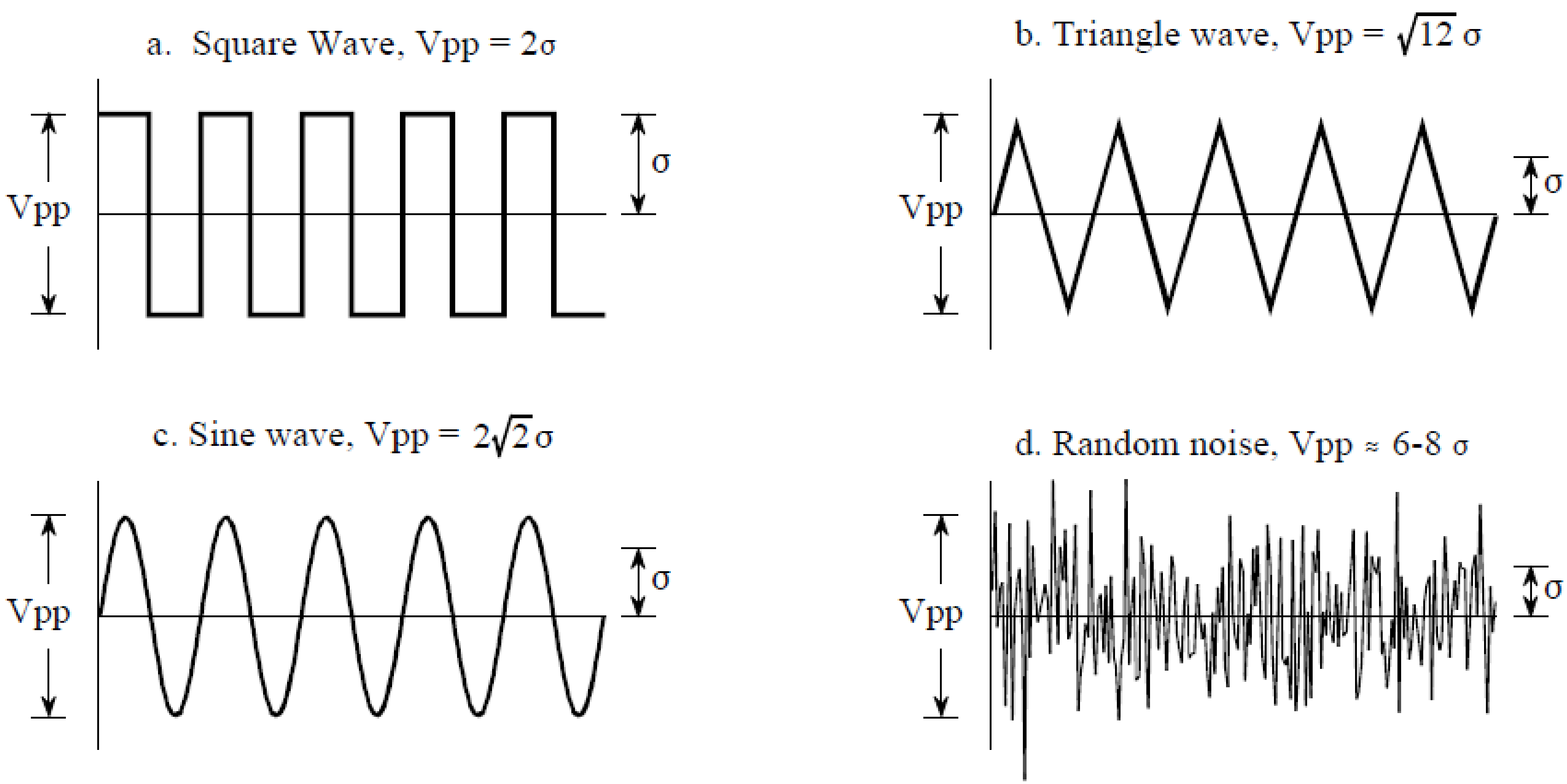
$$|a+bi| = \sqrt{a^2+b^2})$$

$$\sqrt{\cos^2(x)+\sin^2(x)} = 1$$

Statistic and Probability of Random Signals

Deterministic & Random Signals

- **Peak-to-peak Value**
  - → Random signal has no exact peak-to-peak value



Statistics: Mean & Standard Deviation

- **Statistic**
  - → The science of interpreting numerical data, such as acquired signals.
- **Mean (μ)**
  - → Average value of a signal

$$\mu = \frac{1}{N} \sum_{i=0}^{N-1} x_i$$

- **Standard Deviation (σ)**
  - → How far the i-th sample deviates (differs/fluctuates) from the mean: $|x_i - \mu|$
  - → **Standard Deviation:**

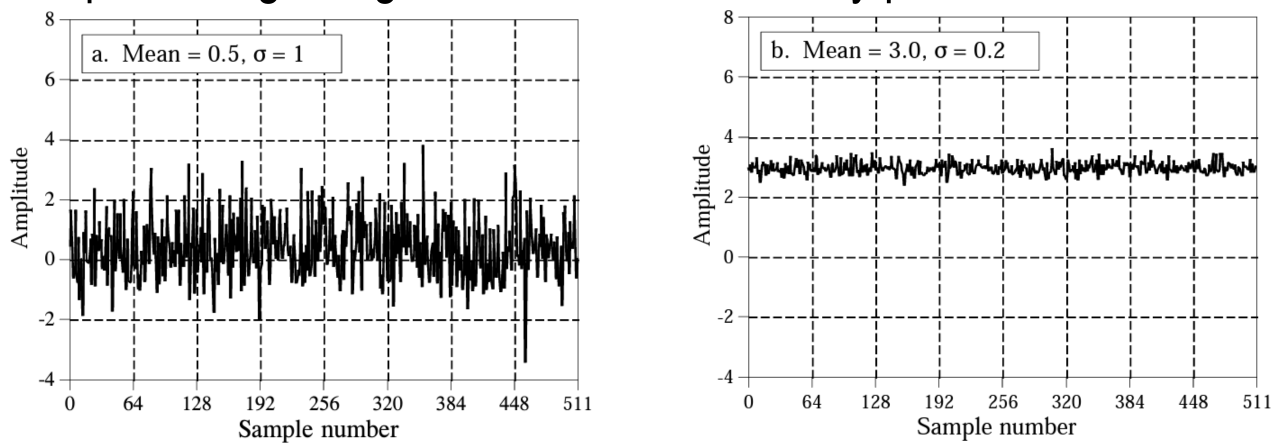
$$\sigma = \sqrt{(x_0 - \mu)^2 + (x_1 - \mu)^2 + \cdots + (x_{N-1} - \mu)^2 / (N-1)}$$

  - → **Variance**: The power of fluctuation

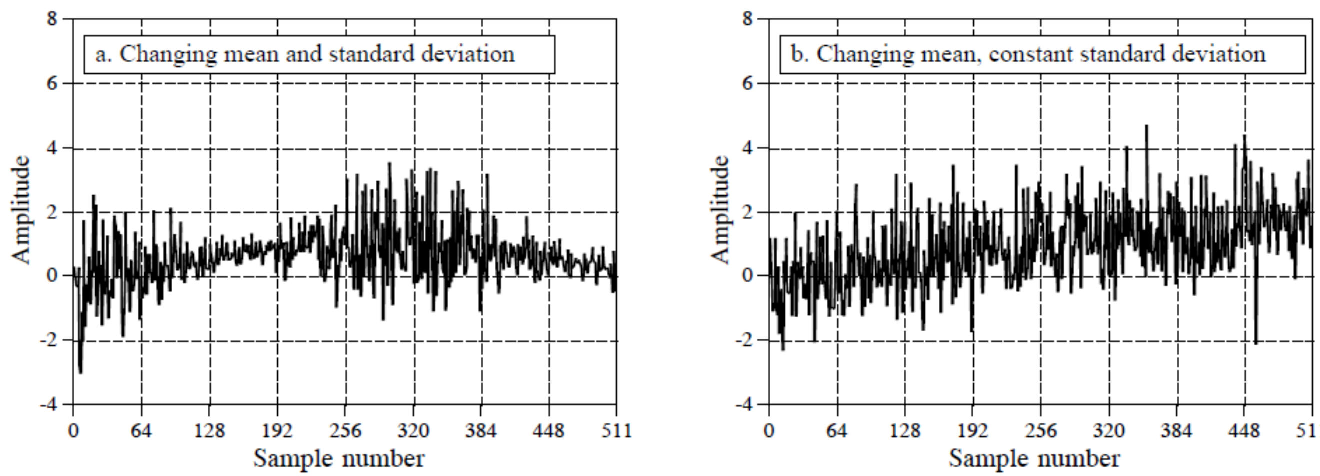
$$\sigma^2 = \frac{1}{N-1} \sum_{i=0}^{N-1} (x_i - \mu)^2$$

Statistics: Mean & Standard Deviation

- **Example:**
  - → Examples of signals generated from stationary processes.

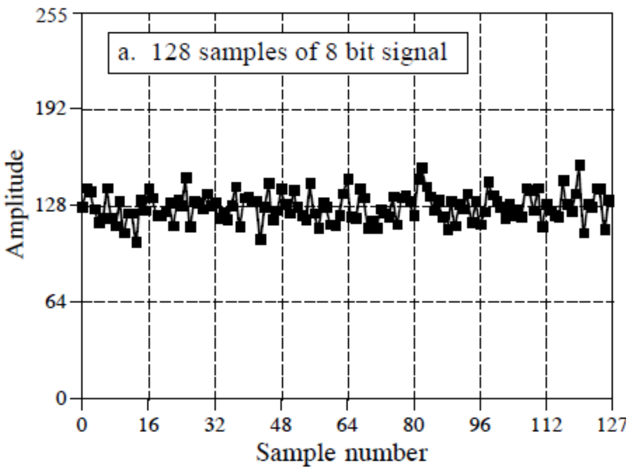


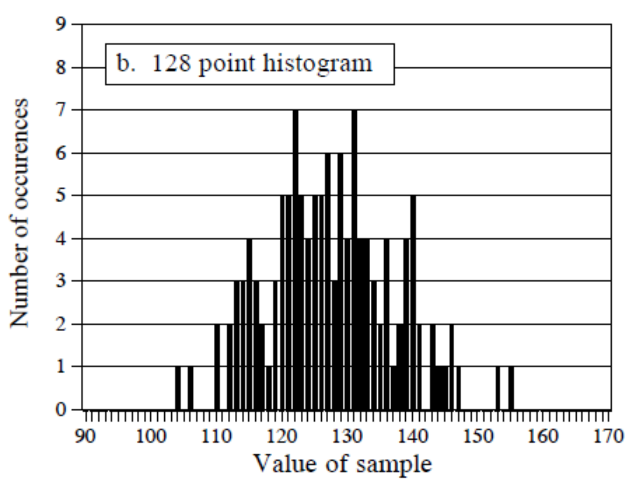
  - → Examples of signals generated from nonstationary processes.

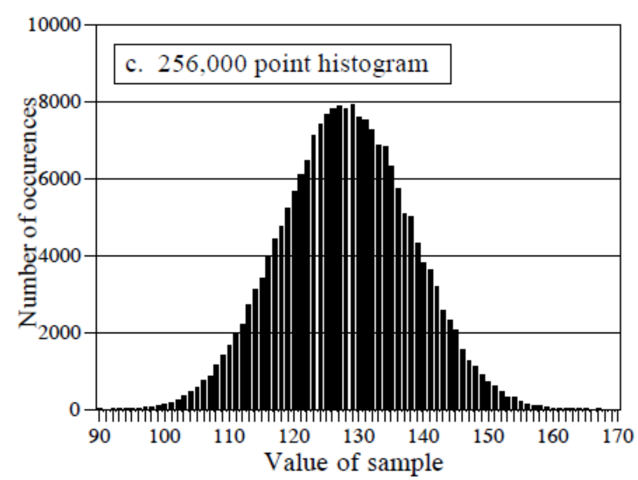


Probability: Histogram, Pmf & Pdf

- **Probability**
  - → To understand the processes that generate signals.
- **Histogram**
  - → the number of samples there are in the signal that have each of these possible values.







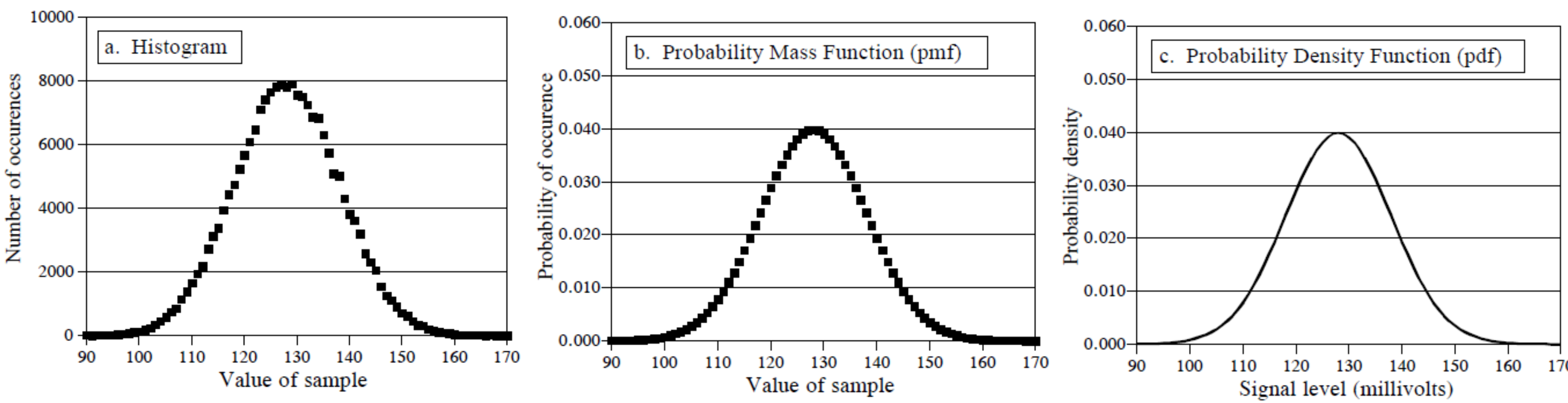
Probability: Histogram, Pmf & Pdf

Histogram, Pmf & Pdf

→ **Histogram:** Calculate the occurence from a finite number of samples

→ **Probability Mass Function (Pmf):** The probabilities of the underlying process

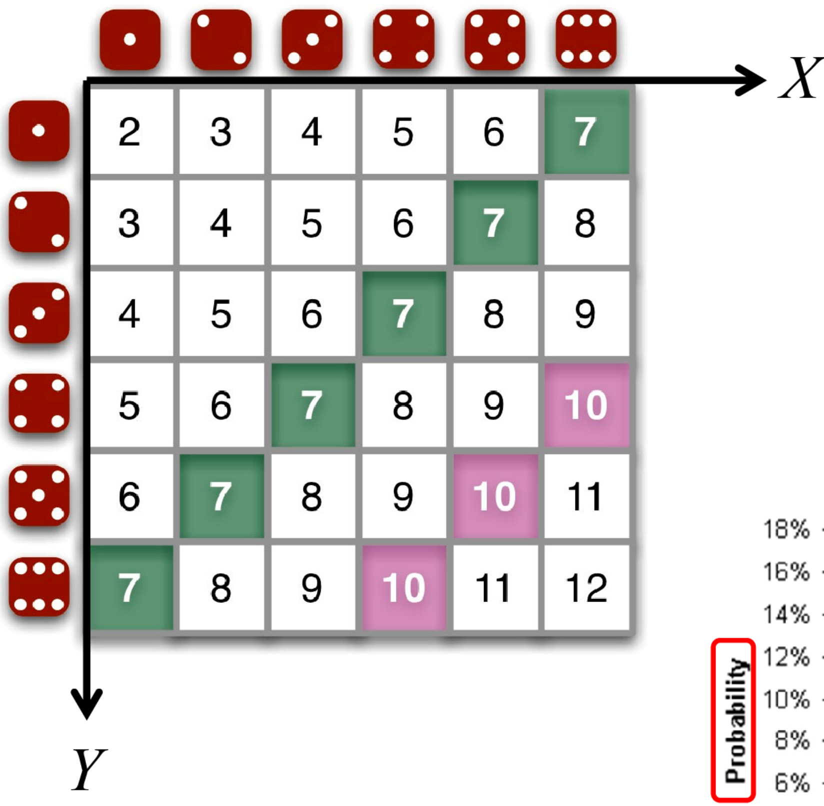
→ **Probability Density Function (Pdf):** Similar to the pmf, but it is used with continuous signals.



Probability Mass Function

Probability Mass Function Calculation

X = *first roll*, Y= *second roll*



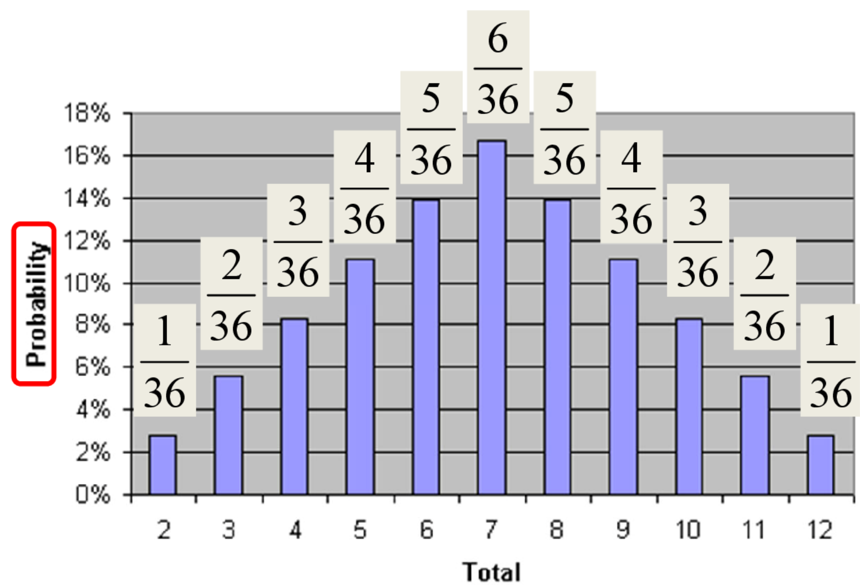
$$Z = X + Y$$

$$p_Z(2) = p(Z = 2) = 1/36$$

$$p_Z(3) = p(Z = 3) = 2/36$$

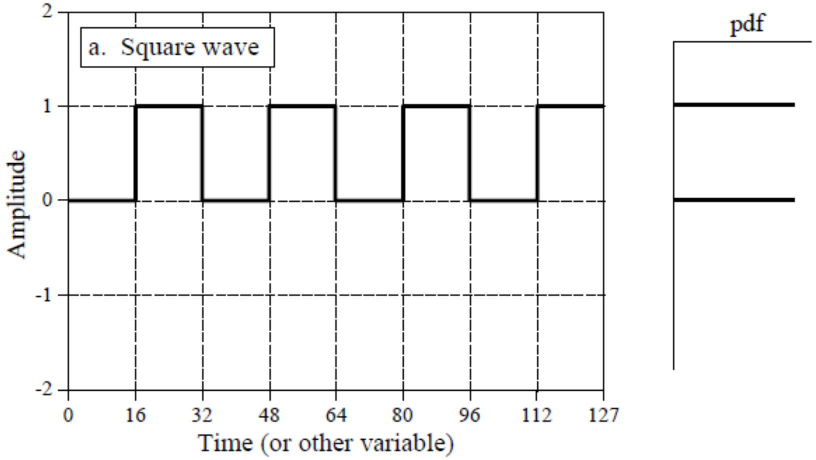
$$p_Z(4) = p(Z = 4) = 3/36$$

$$\vdots$$

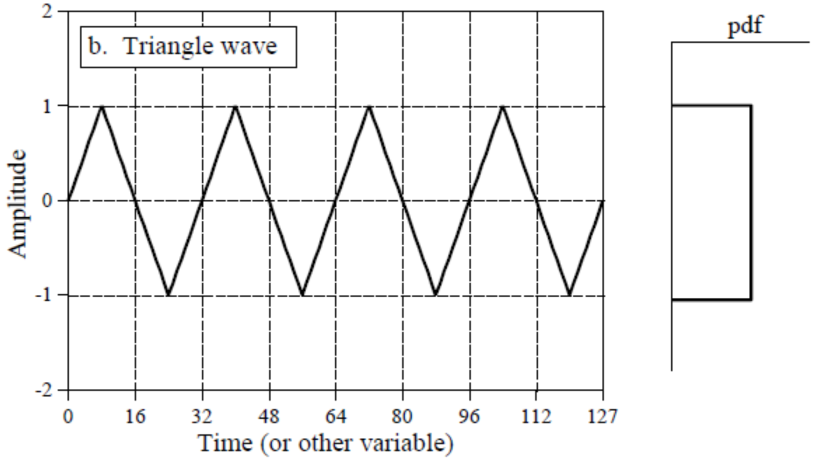


Probability Density Function

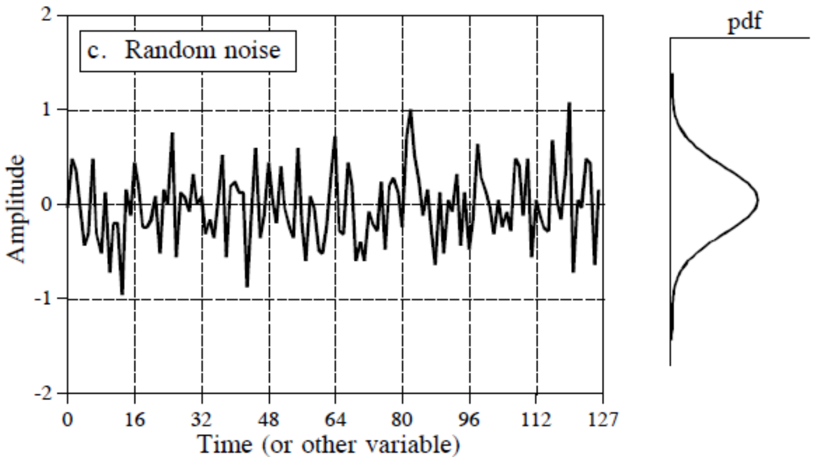
- **Examples**



Pdf: only having two possible values



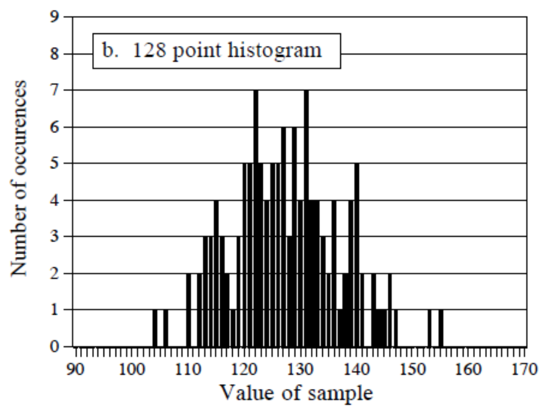
Pdf: uniform distribution



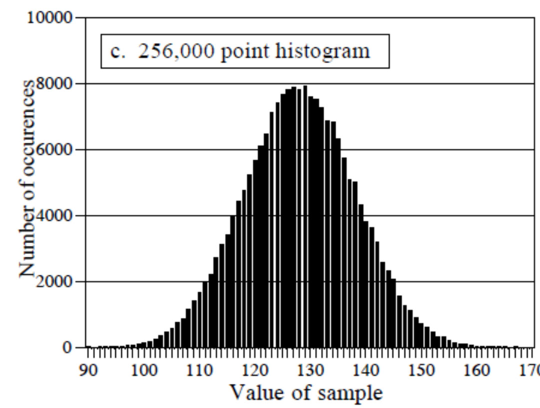
Pdf: Gaussian distribution

Normal/Gaussian Distribution (Bell Shaped)

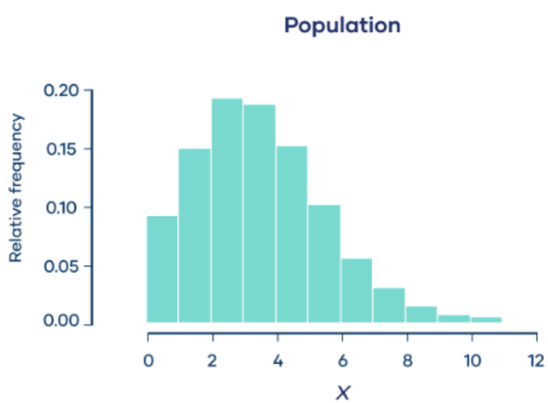
- **Central Limit Theorem**
  - → If we take sufficiently large samples from a population, the samples' means will be normally distributed, even if the population isn't normally distributed.

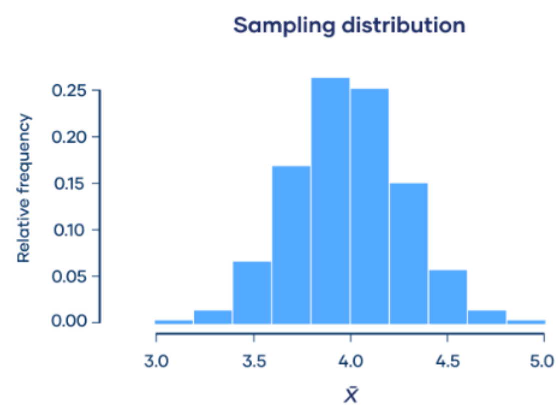


Large samples →



  - → Example: Take samples from a population that isn't normally distributed

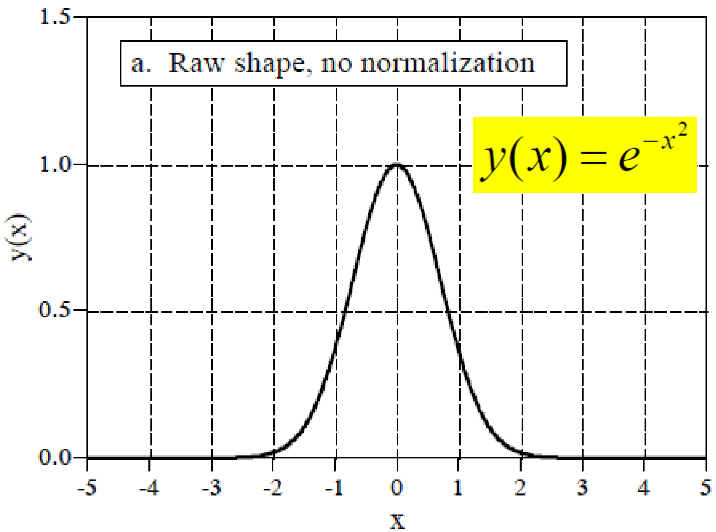


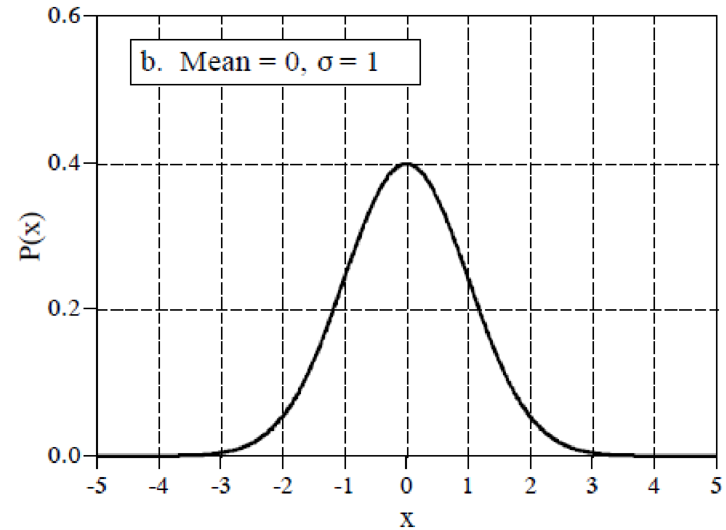


Normal/Gaussian Distribution (Bell Shaped)

▪ Normal/Gaussian Distribution

→ Signals formed from random processes usually have a bell-shaped probability density (distribution) function.





→ Name after the great German mathematician, Karl Friedrich Gauss (1777-1855)

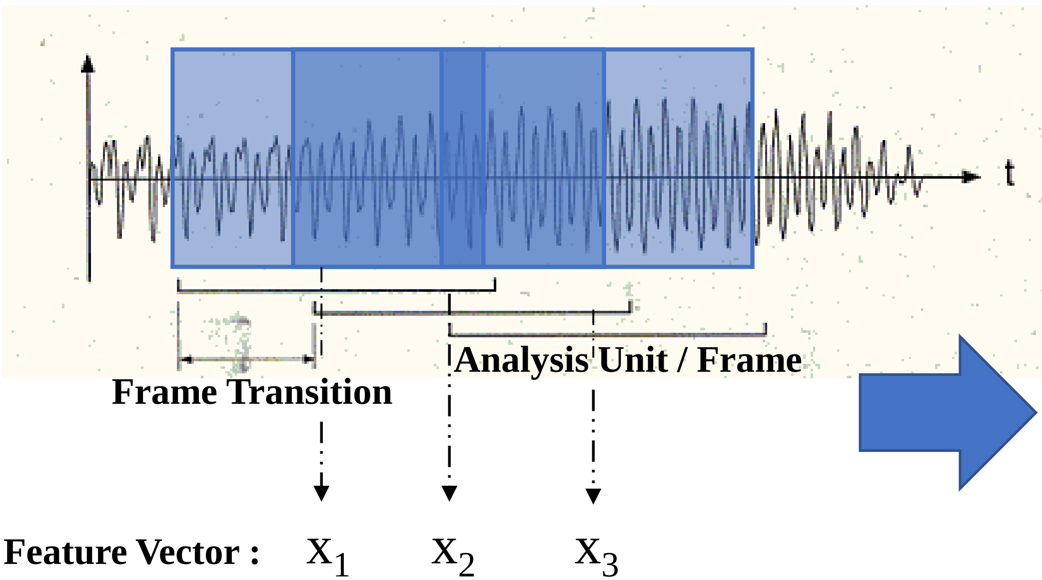
$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}$$

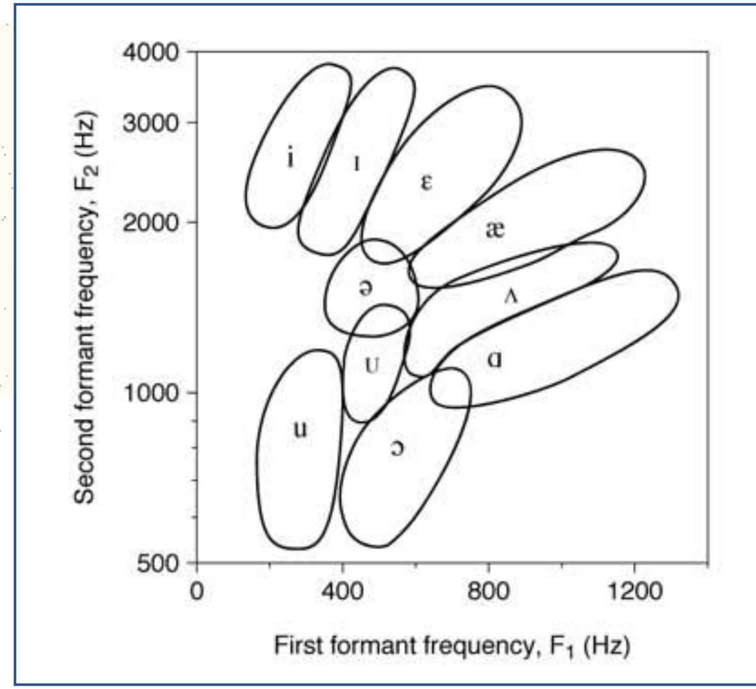
mean

standard deviation

Normal/Gaussian Distribution (Bell Shaped)

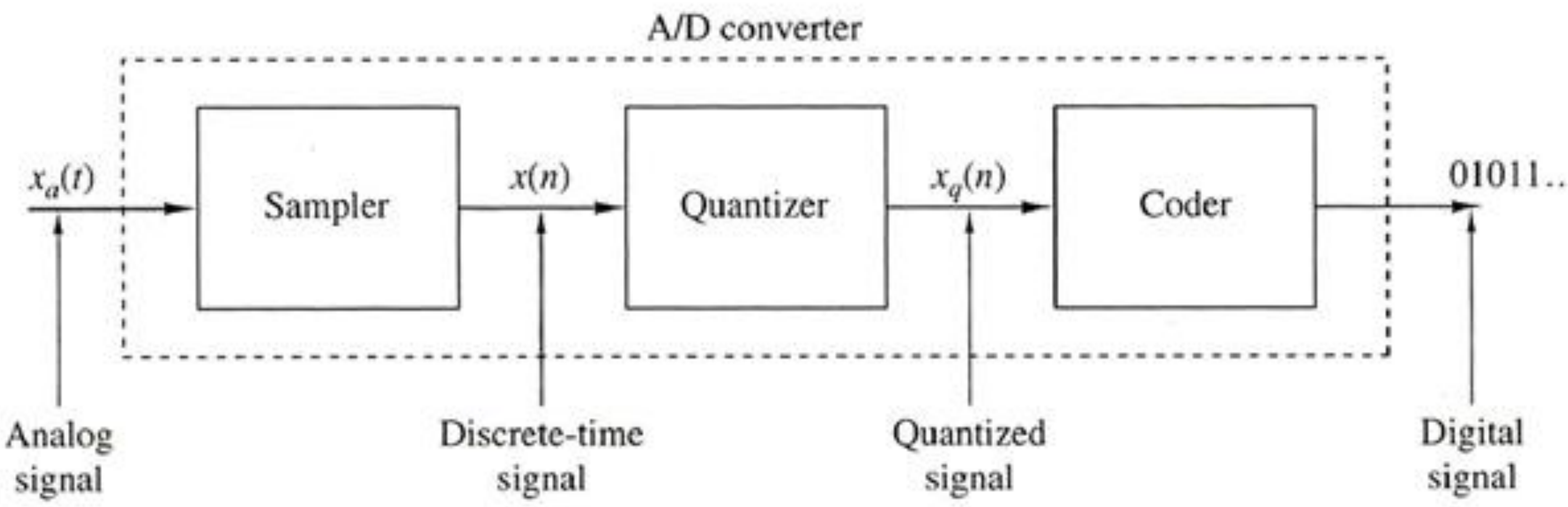
- **Example: Speech**





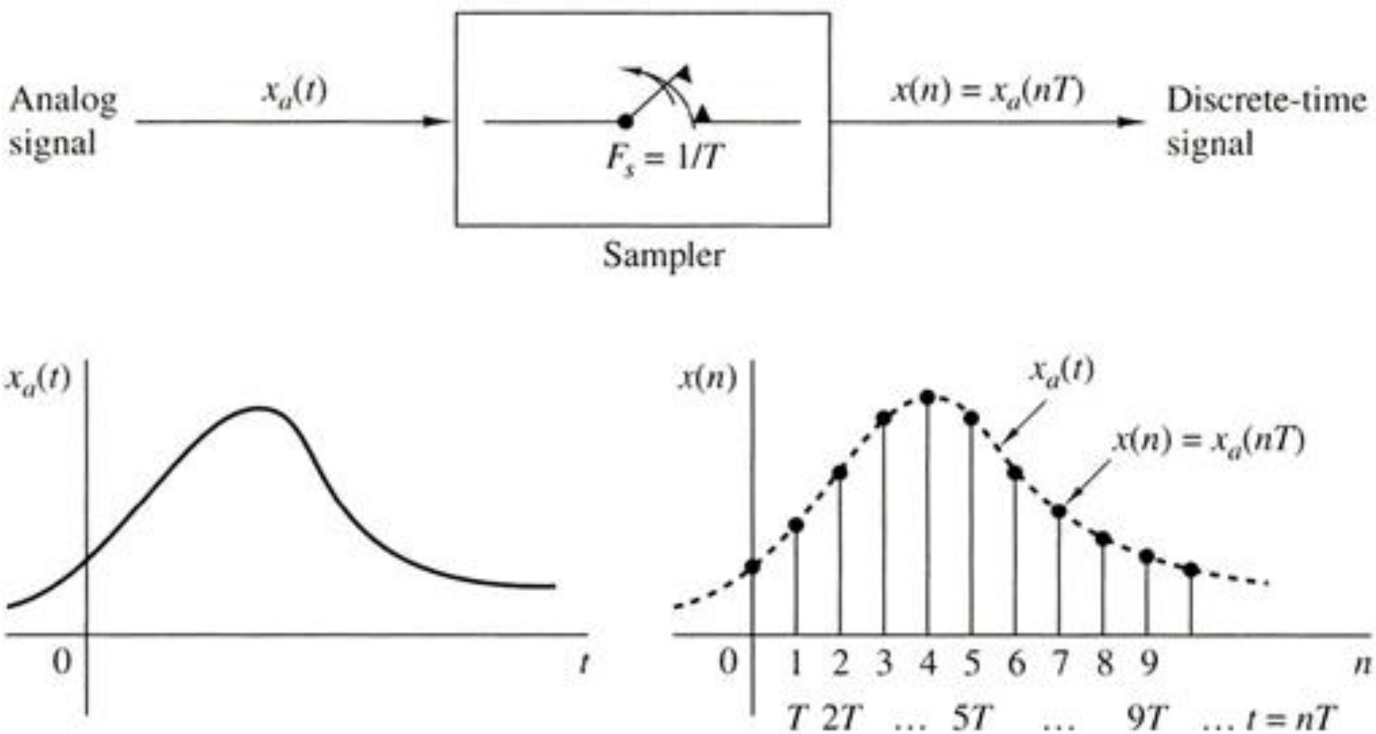
Analog-to-Digital Converter

Basic parts of an analog-to-digital converter



Sampler Component

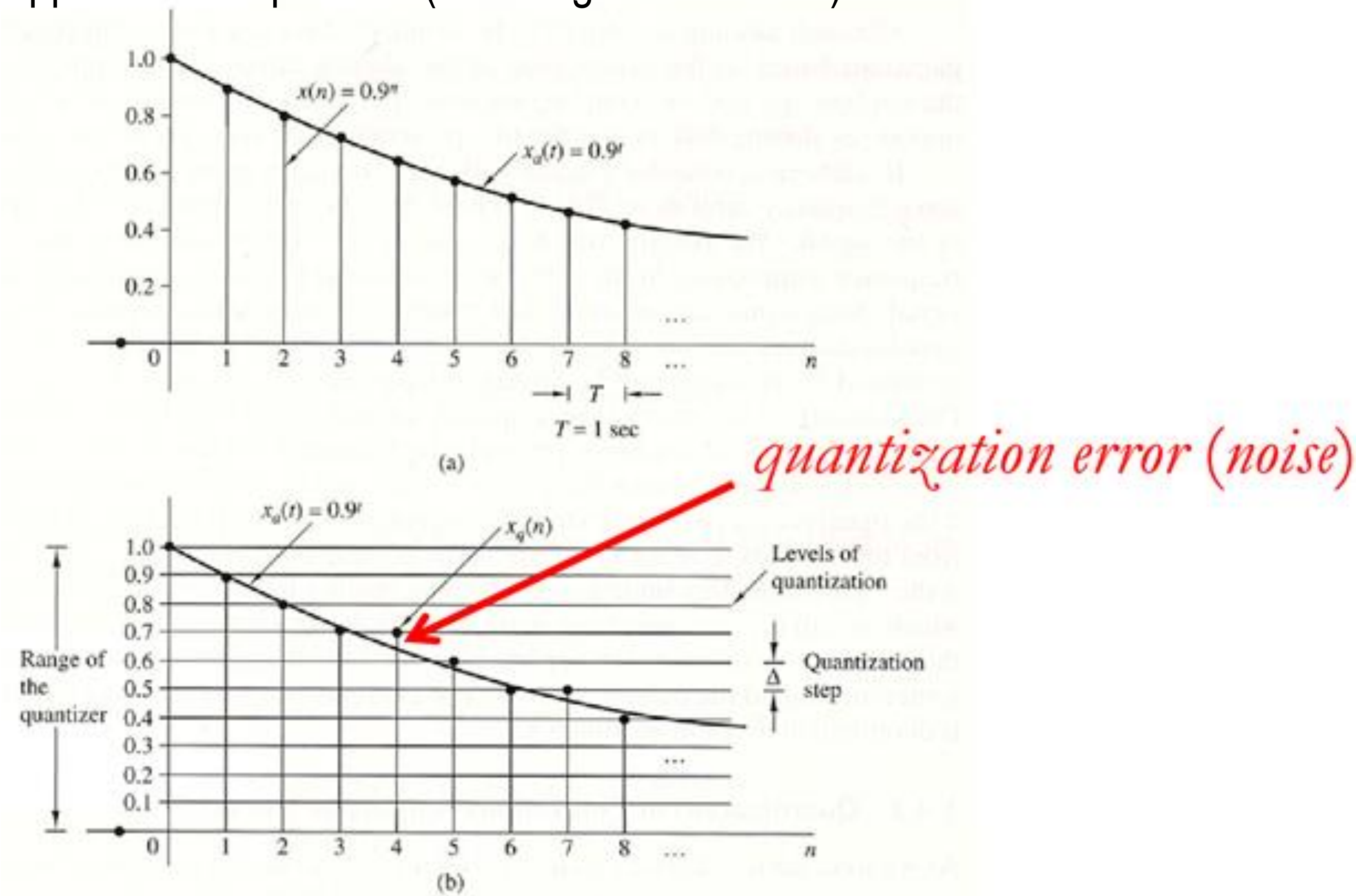
Periodic (uniform) sampling of an analog signal



"Taking samples" of the analog signal $x_a(t)$ every T seconds

Quantizer Component

- **Quantization:**
  - → Converting a discrete-time continuous-amplitude signal into a digital signal
  - → An approximation process (rounding or truncation)



Elementary Signals

Unit Impulse Signal / Unit Sample Sequence

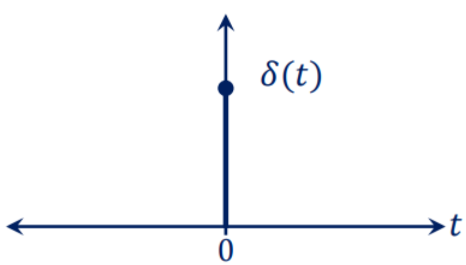
Definition

→ An impulse signal has zero value except at t = 0. It has an infinitely high-value t = 0.

→ For the unit impulse signal, the value at t = 0 is 1.

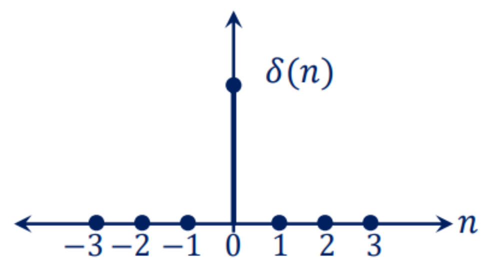
Continuous-Time Signal

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



Discrete-Time Signal

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



MATLAB

$$\delta(k - k_0) = \begin{cases} 1, & k = k_0 \\ 0, & k \neq k_0 \end{cases}, \quad k_1 \leq k_0 \leq k_2$$

```
function [x,k] = impseq(k0,k1,k2)
% Generates x(k) = delta(k-k0); k1 <= k <= k2
% ------------------------------------------------------------
% [x,k] = impseq(k0,k1,k2)
%
k = [k1:k2]; x = [(k-k0) == 0];
```

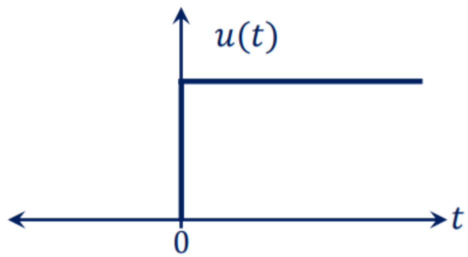
Unit Step Signal / Unit Step Sequence

Definition

→ A unit step signal has a unity value for $t \geq 0$ else zero value

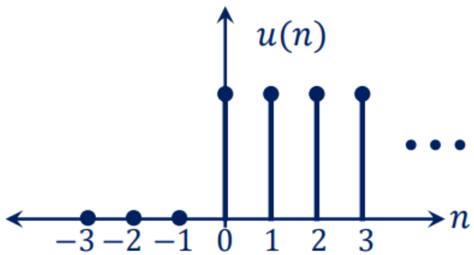
Continuous-Time Signal

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



Discrete-Time Signal

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



MATLAB

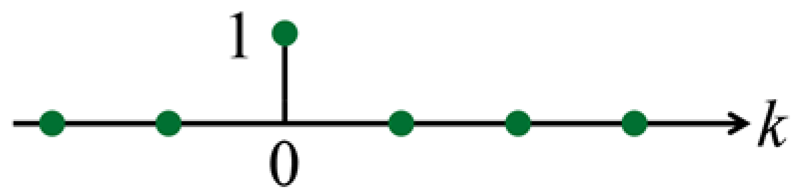
$$u(k-k_0) = \begin{cases} 1, & k \geq k_0 \\ 0, & k < k_0 \end{cases}, \quad k_1 \leq k_0 \leq k_2$$

```
function [x,k] = stepseq(k0,k1,k2)
% Generates x(k) = u(k-k0); k1 <= k <= k2
% -----------------------------------------------------
% [x,k] = stepseq(k0,k1,k2)
%
k = [k1:k2]; x = [(k-k0) >= 0];
```

Relationship between $\delta(k)$ and $u(k)$

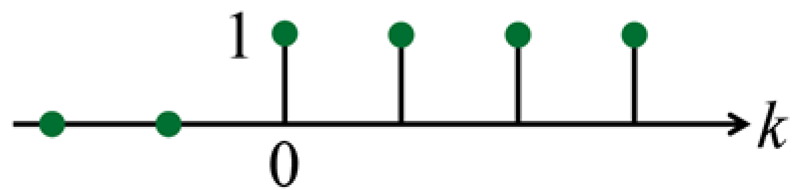
First order backward difference

$$\delta[k] = u[k] - u[k-1]$$



Running sum

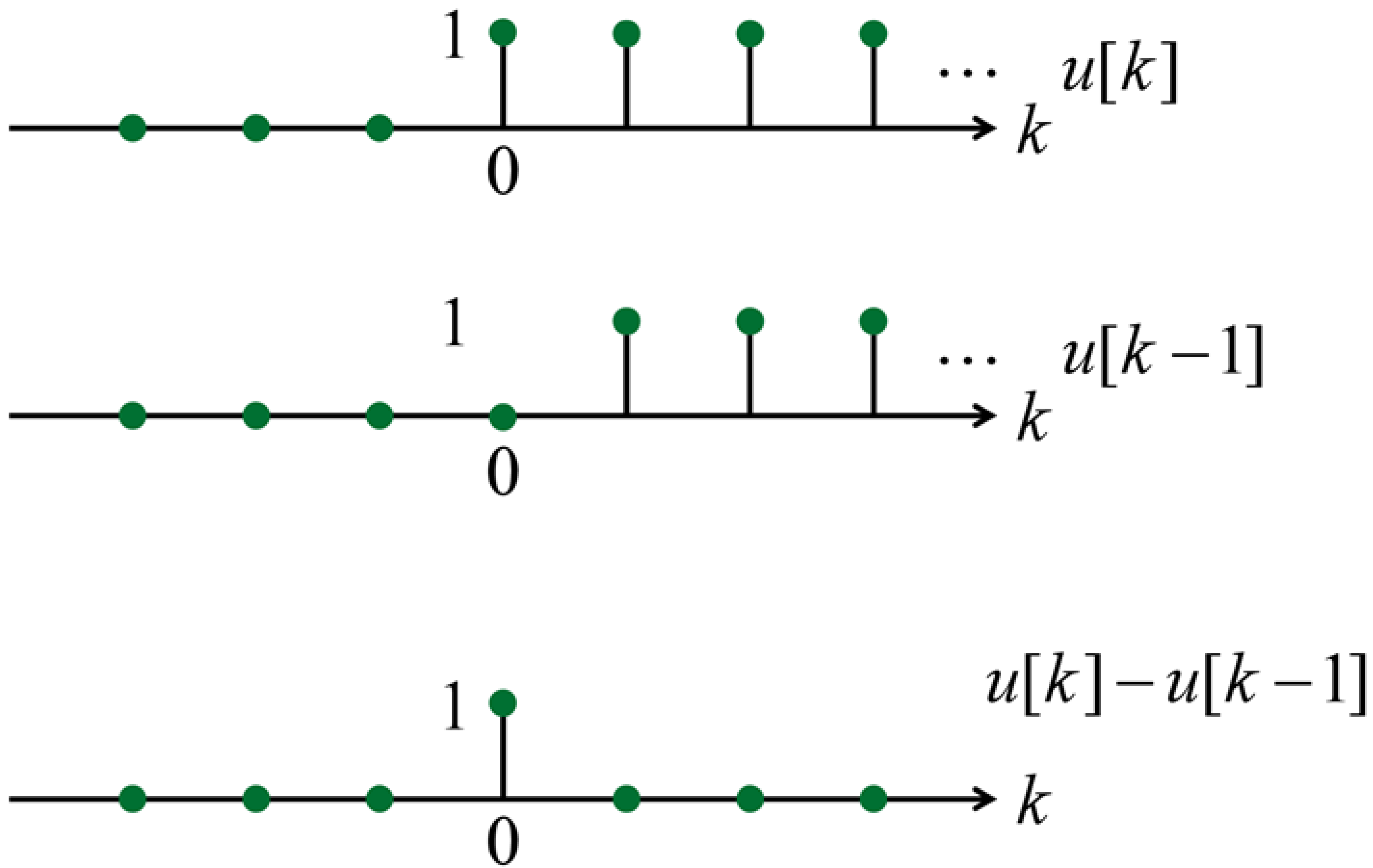
$$u[k] = \sum_{n=-\infty}^{k} \delta[n]$$



The *unit impulse* can be used to *sample* a discrete-time signal $x[k]$:

$$x[0] = \sum_{n=-\infty}^{\infty} x[n]\delta[n], \qquad x[k] = \sum_{n=-\infty}^{\infty} x[n]\delta[k-n]$$

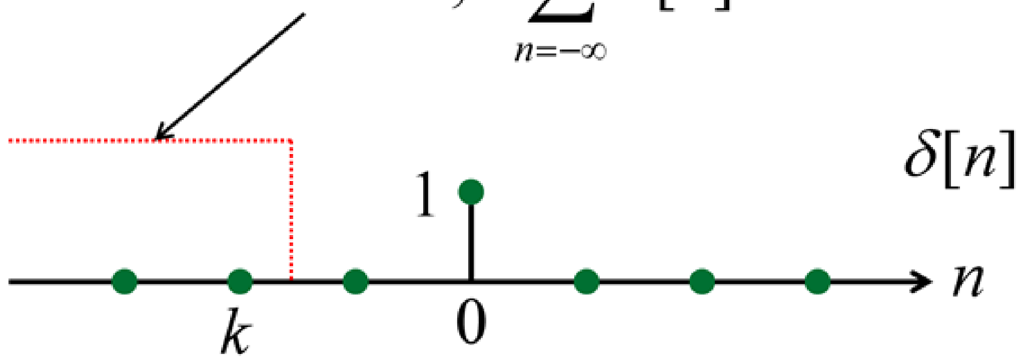
Relationship between $\delta(k)$ and $u(k)$



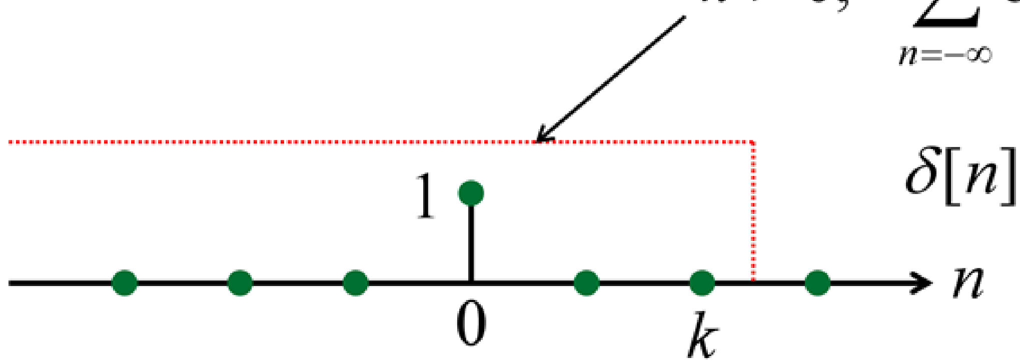
Relationship between $\boldsymbol{\delta(k)}$ and $\boldsymbol{u(k)}$

$$u[k] = \sum_{n=-\infty}^{k} \delta[n]$$

$$k < 0, \quad \sum_{n=-\infty}^{k} \delta[n] = 0$$

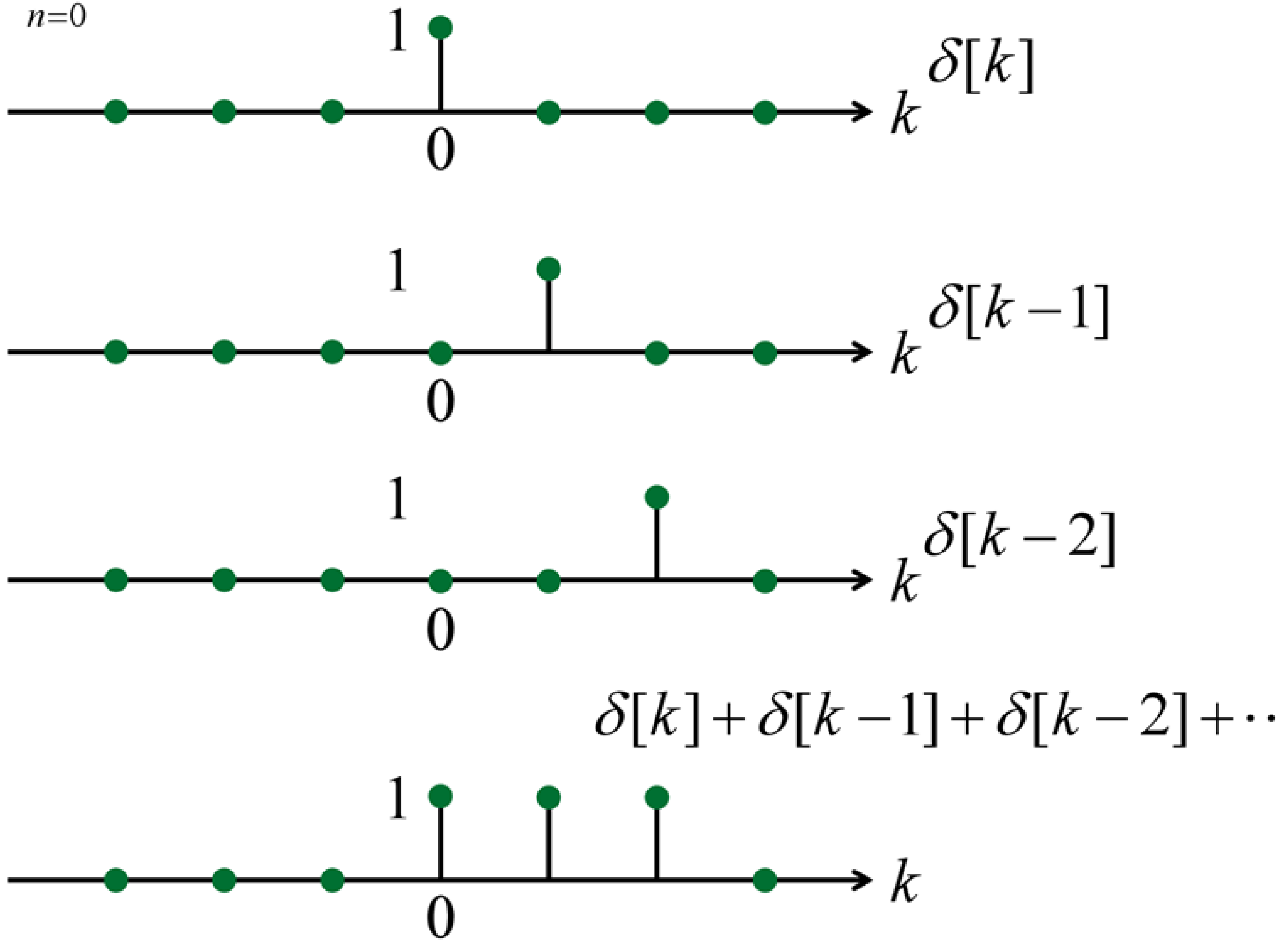


$$k > 0, \quad \sum_{n=-\infty}^{k} \delta[n] = 1$$



Relationship between $\delta(k)$ and $u(k)$

$$u[k] = \sum_{n=0}^{\infty} \delta[k-n]$$

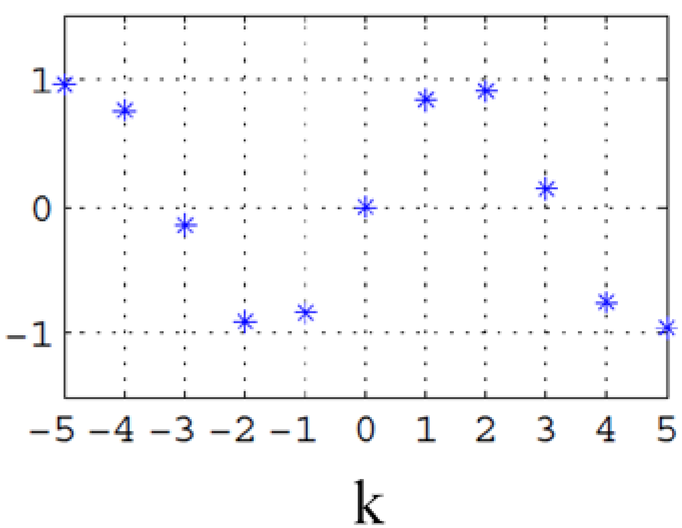


Relationship between $\delta(k)$ and $u(k)$

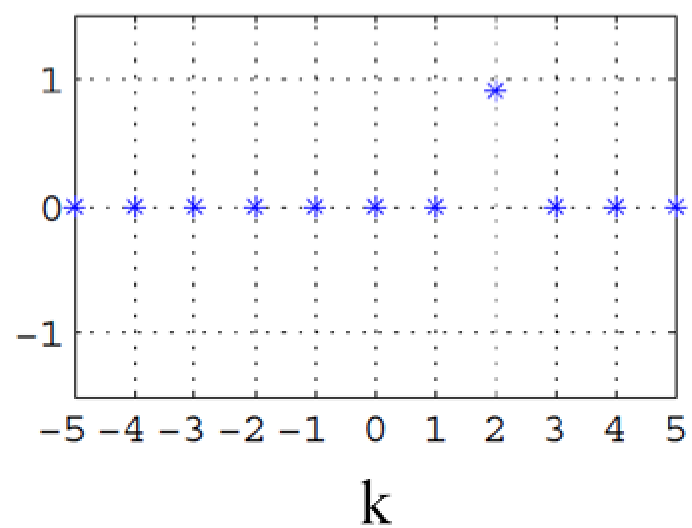
To pick the n-th sample of a signal $x[k]$, multiply $x[k]$ by $\delta[k-n]$.

Sampling property of unit impulse: $x[k]\delta[k-n] = x[n]\delta[k-n]$

$x[k]$



$x[k]\delta[k-2]$

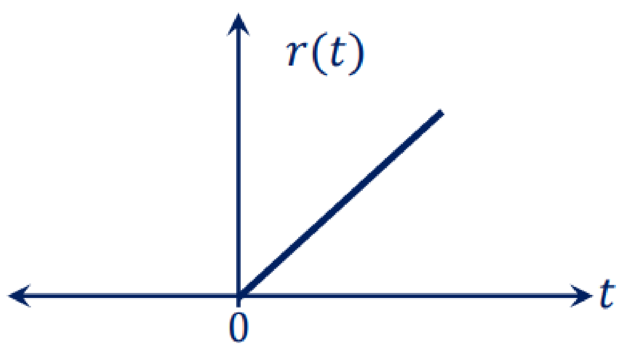


Unit Ramp Signal / Unit Ramp Sequence

- **Definition**
  - → A ramp step signal has a unity slop value for $t \geq 0$, otherwise, it has zero value

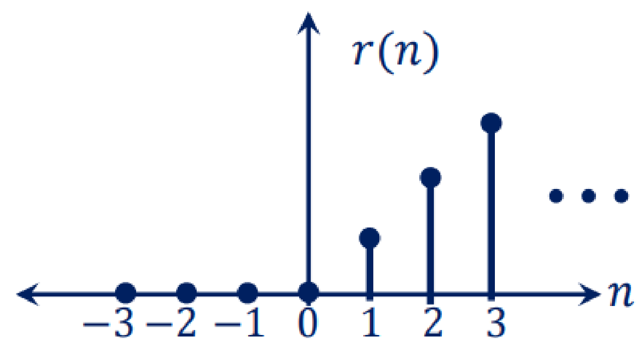
Continuous-Time Signal

$$r(t) = \begin{cases} t & t \geq 0 \\ 0 & t < 0 \end{cases}$$



Discrete-Time Signal

$$r(n) = \begin{cases} n & n \geq 0 \\ 0 & n < 0 \end{cases}$$



Rectangular Pulse Signal

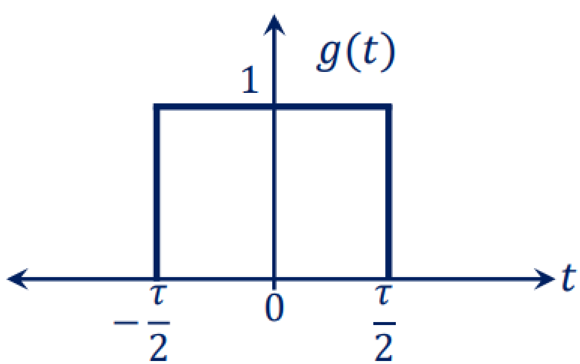
▪ Definition

→ A unit rectangular pulse has unit amplitude within a time interval, otherwise, it has zero value

→ It is also called the Gate pulse, Pulse function, or Window function, etc.

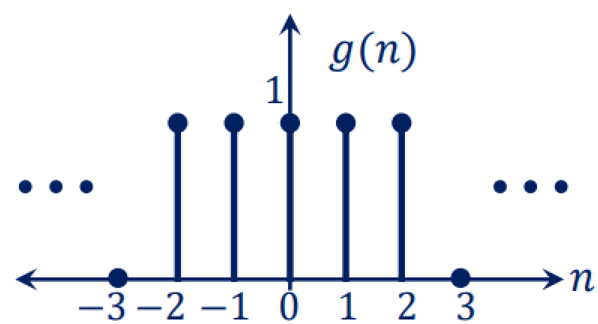
Continuous-Time Signal

$$g(t) = \begin{cases} 1 & -\frac{\tau}{2} \leq t \leq +\frac{\tau}{2} \\ 0 & \text{Otherwise} \end{cases}$$



Discrete-Time Signal

$$g(n) = \begin{cases} 1 & -m \leq n \leq +m \\ 0 & \text{Otherwise} \end{cases}$$



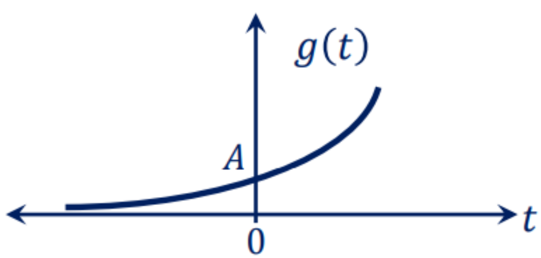
Exponential Signal

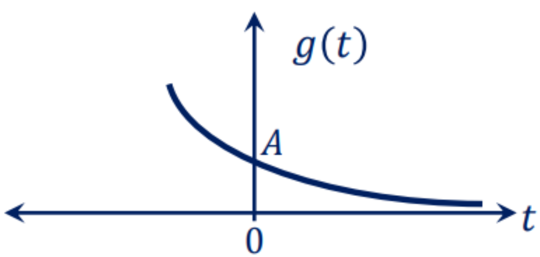
Definition

→ An exponential signal can either have exponentially rising or falling amplitude depending upon its exponent value

Continuous-Time Signal

$$g(t) = Ae^{bt} \qquad A > 0$$

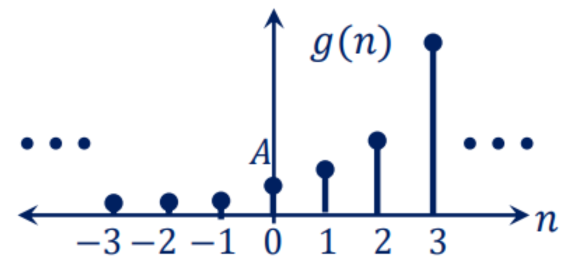




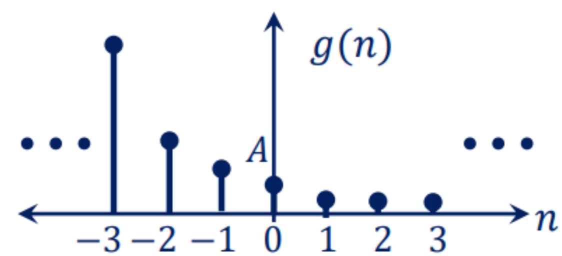
Discrete-Time Signal

$$g(n) = Ae^{bn} \qquad A > 0$$

$b > 0$



$b < 0$



Exponential Signal

- **MATLAB**

Real-valued exponential sequence $x(k) = a^k, \forall k; \quad a \in R$

$$x(k) = (0.9)^k, 0 \le k \le 10$$

```
>> k = [0:10]; x = (0.9).^k;
```

Complex-valued exponential sequence $x(k) = e^{(\sigma + j\omega_0)k}, \forall k;$

$$x(k) = \exp[(2 + j3)k], 0 \le k \le 10$$

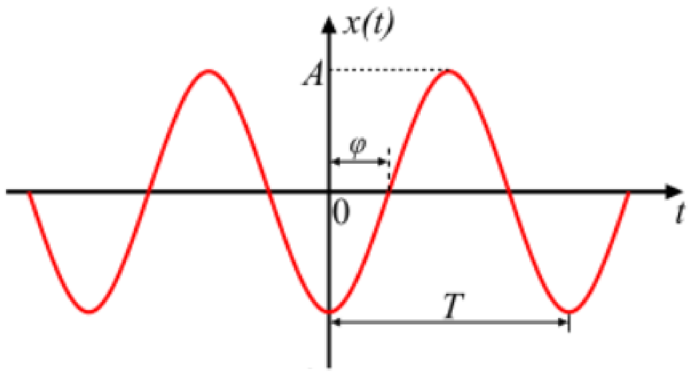
```
>> k = [0:10]; x = exp((2+3j)*k);
```

Sinusoidal Signal / Sinusoidal Sequence

Definition

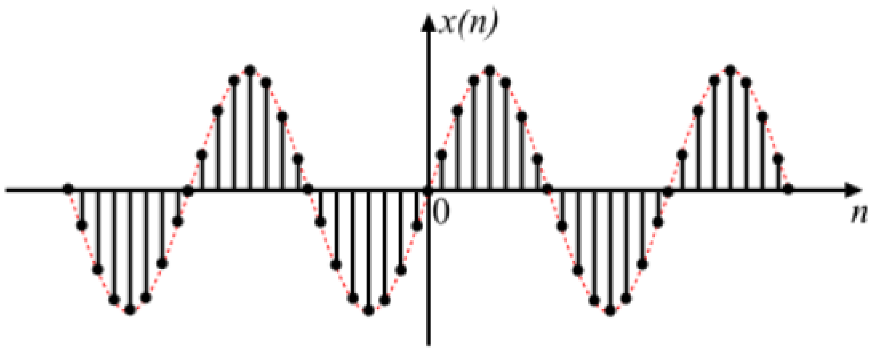
→ a function that describes a smooth periodic oscillation

Continuous-Time Sinusoidal Signal



$x(t) = A \sin(\omega t + \varphi) = A \sin(2\pi f t + \varphi)$

Discrete-Time Sinusoidal Signal



$x(n) = A \sin(\omega n + \varphi) = A \sin(2\pi f n + \varphi)$

MATLAB

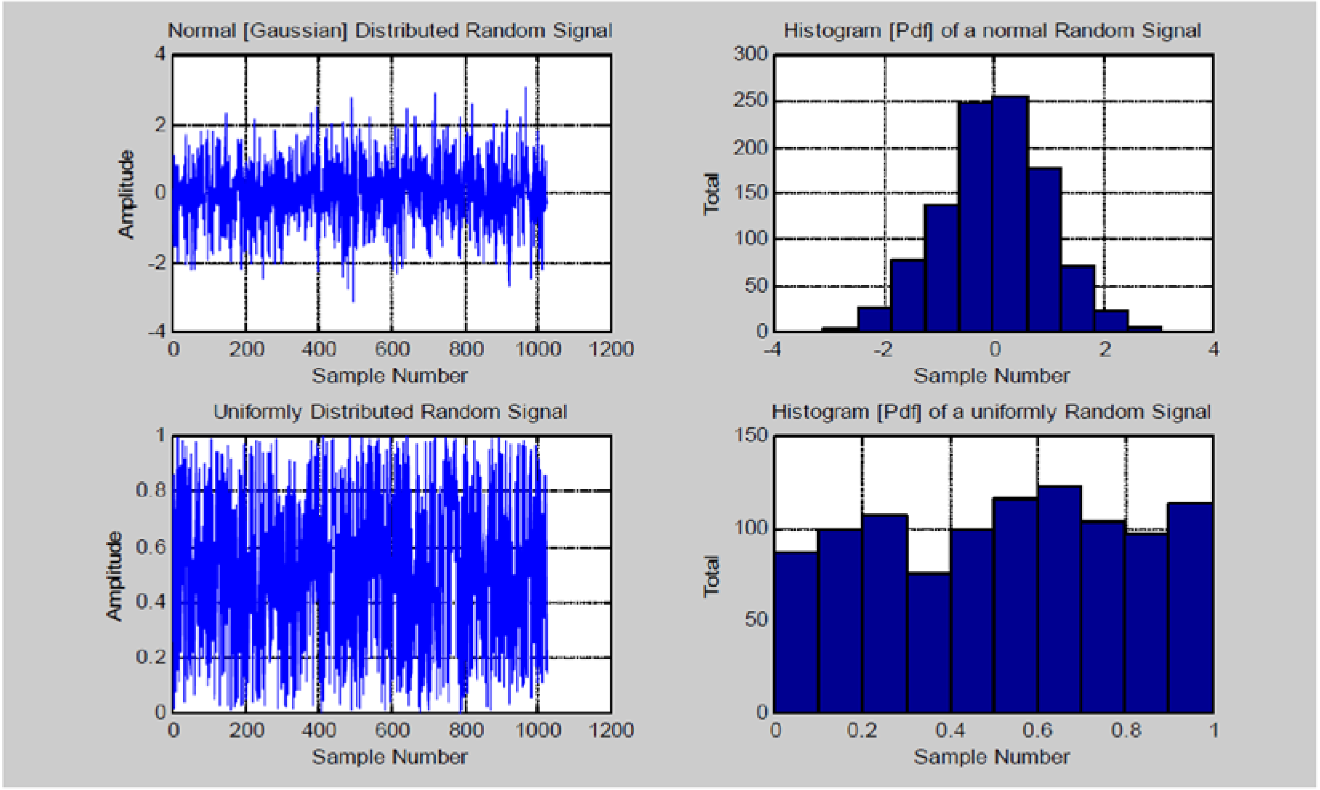
$$x(k) = 3\cos(0.1\pi k + \pi/3) + 2\sin(0.5\pi k), 0 \le k \le 10$$

```
>> k = [0:10]; x = 3*cos(0.1*pi*k+pi/3) + 2*sin(0.5*pi*k);
```

Random Signal / Random Sequence

- **MATLAB**

rand(1,N)	*generates a length* N *random sequence whose elements are uniformly distributed between* $[0,1]$
randn(1,N)	*generates a length* N *Gaussian random sequence with mean* 0 *and variance* 1



Normally and uniformly distributed random signals

Random Signal / Random Sequence

```
>> N=1024;             % define number of samples
>> R1=randn(1,N);      % generate normal random numbers
>> R2=rand(1,N);       % generate uniformly random numbers
>> figure(1);          % select the figure
>> subplot(2,2,1);     % subdivide the figure into 4 quadrants
>> plot(R1);           % plot R1 in the first quadrant
>> grid;
>> title('Normal [Gaussian] Distributed Random Signal');
>> xlabel('Sample Number');
>> ylabel('Amplitude');
>> subplot(2,2,2);     % select the second quadrant
>> hist(R1);           % plot the histogram of R1
>> grid;
>> title('Histogram [Pdf] of a normal Random Signal');
>> xlabel('Sample Number');
>> ylabel('Total');
>> subplot(2,2,3);
>> plot(R2);
>> grid;
>> title('Uniformly Distributed Random Signal');
>> xlabel('Sample Number');
>> ylabel('Amplitude');
>> subplot(2,2,4);
>> hist(R2);
>> grid;
>> title('Histogram [Pdf] of a uniformly Random Signal');
>> xlabel('Sample Number');
>> ylabel('Total');
```

Signal Operations

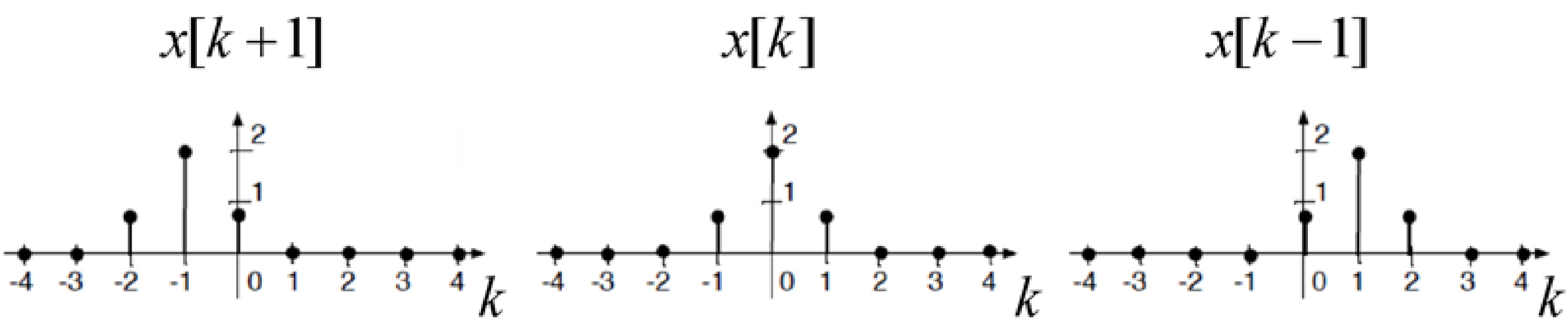
Transformations: Shift, Reversal, Scaling

- **Time Shift**

$$x(t-t_0), \quad x[k-k_0]$$

If $t_0 > 0, \quad k_0 > 0,$ signal is shifted to the *right*

If $t_0 < 0, \quad k_0 < 0,$ signal is shifted to the *left*



Transformations: Shift, Reversal, Scaling

- **MATLAB**

$$y(k) = \{x(k-m)\}$$

Each sample of $x(k)$ *is shifted by an amount of* m *to obtain a shifted sequence* $y(k)$. *If we let* $n=k-m$, *then* $k=n+m$ *and the above operation is given by*

$$y(n+m) = \{x(n)\}$$

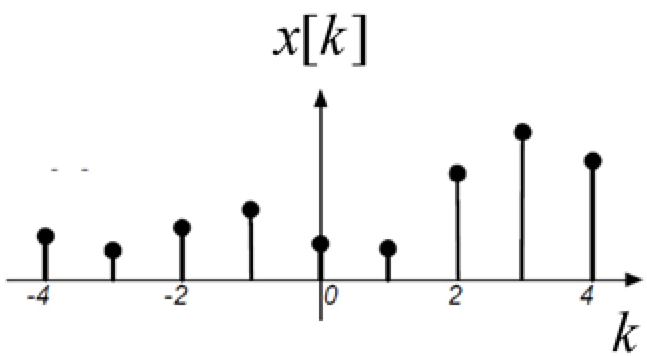
Hence shifting has no effect on the vector x, *but the vector* k *is changed by adding* m *to each element.*

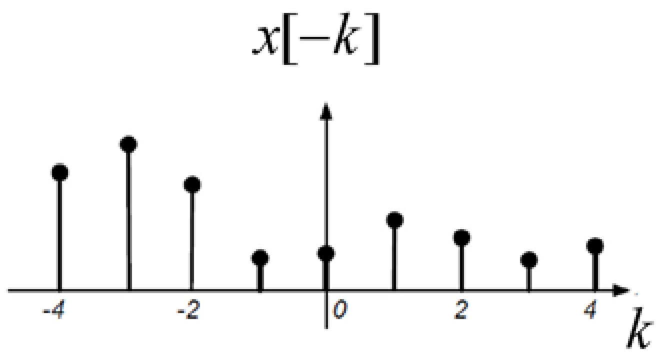
```
function [y,k] = sigshift(x,n,m)
% implement y(k) = x(k-m)
% ----------------------------------------
% [y,k] = sigshift(x,n,m)
%
k = n+m; y = x;
```

Transformations: Shift, Reversal, Scaling

- **Time Reversal**

$$x(-t), \quad x[-k]$$





- **MATLAB**

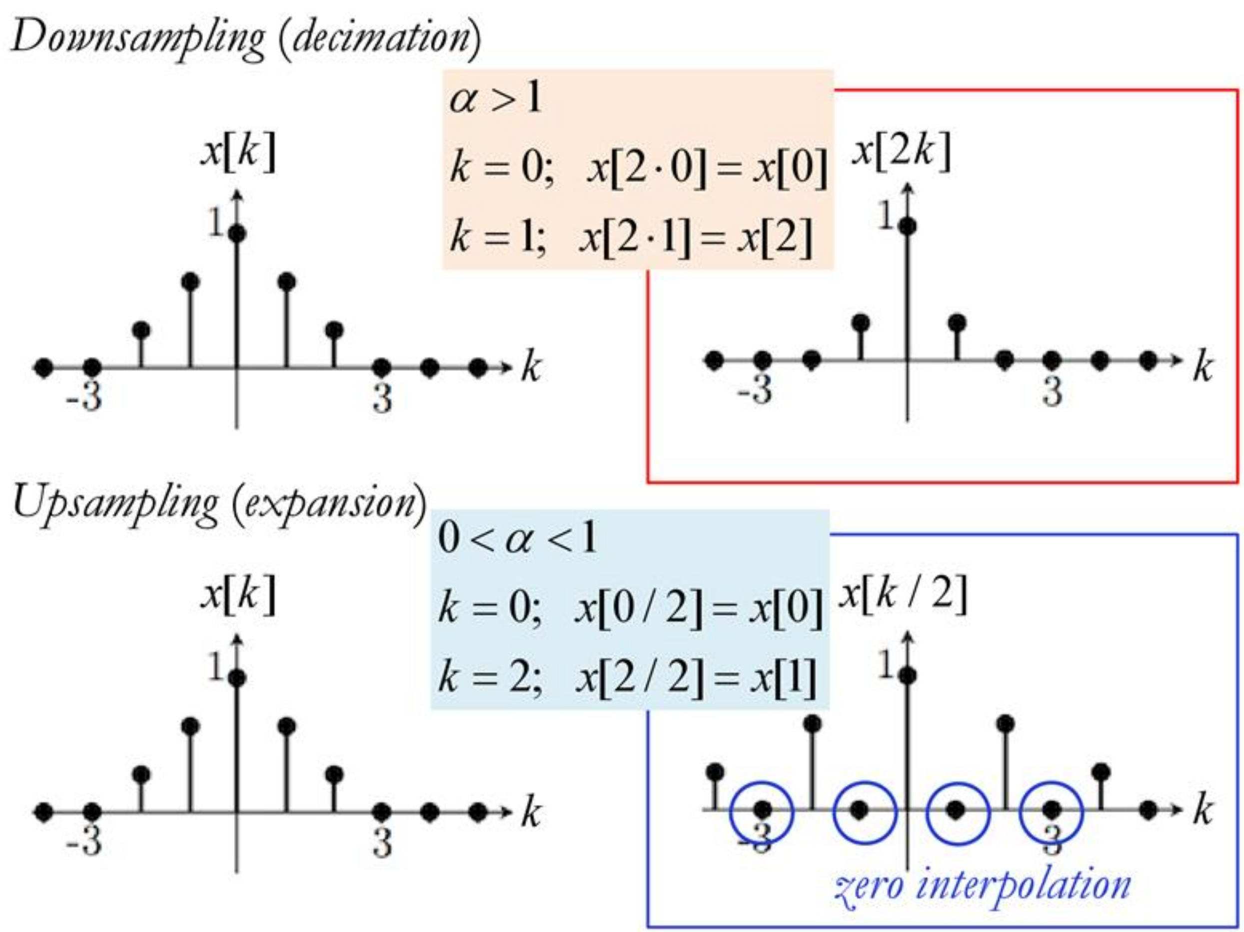
$y(k) = \{x(-k)\}$ fliplr(x) *for sample values*
-fliplr(k) *for sample positions*

```
function [y,k] = sigfold(x,k)
% implement y(k) = x(-k)
% ----------------------------------------
% [y,k] = sigfold(x,k)
%
y = fliplr(x); k = -fliplr(k);
```

Transformations: Shift, Reversal, Scaling

- **Time Scaling** $x(\alpha t)$, $x[\alpha k]$ If $\alpha > 1$, signal appears *compressed*

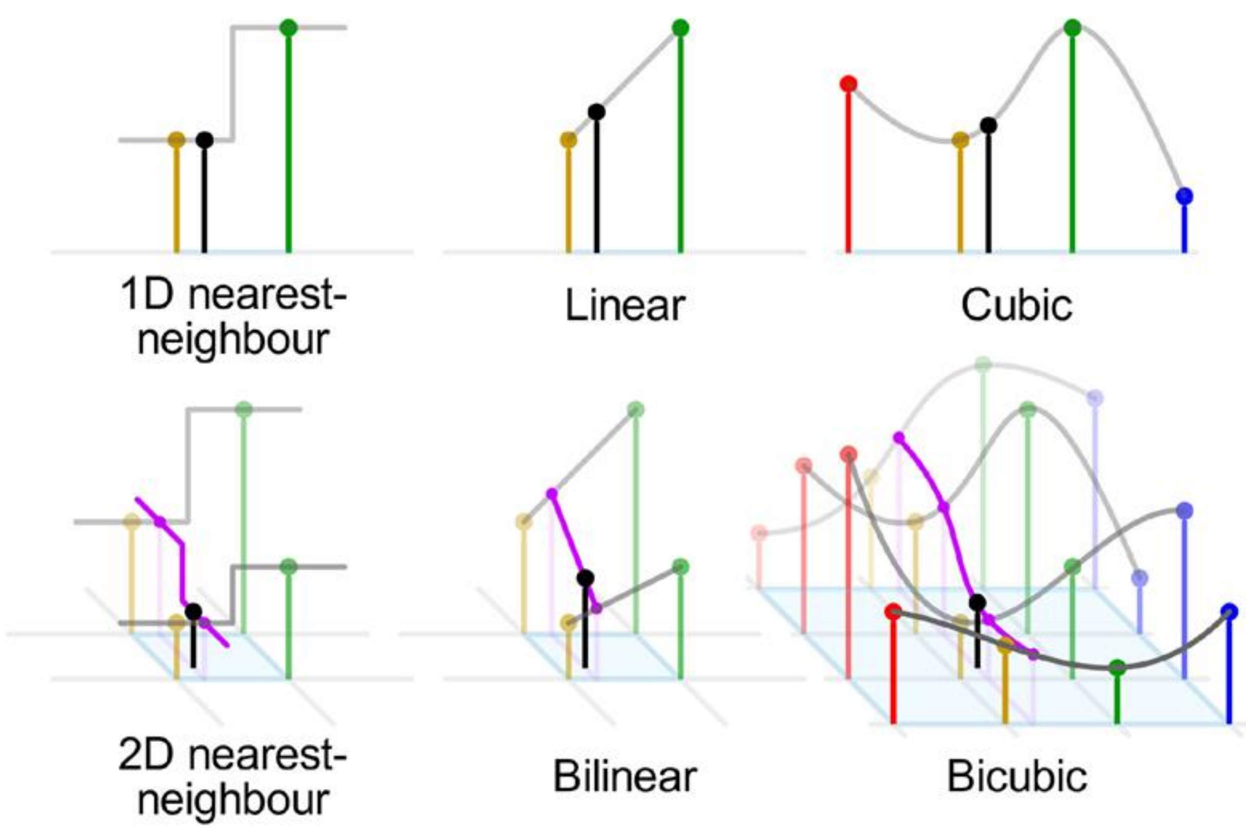
  If $1 > \alpha > 0$, signal appears *stretched*



Transformations: Shift, Reversal, Scaling

- **Time Scaling**

Image Interpolation Techniques (*Wikipedia*)



Transformations: Shift, Reversal, Scaling

- **MATLAB**

$$\alpha\{x(k)\} = \{\alpha x(k)\}$$

An arithmetic operator "" is used to implement the scaling operation in MATLAB.*

Operations: Addition & Multiplication

- **Signal Addition**

$$\{x_1(k)\} + \{x_2(k)\} = \{x_1(k) + x_2(k)\}$$

```
Function [y,k] = sigadd(x1,k1,x2,k2);
% implements  y(k) = x1(k)+x2(k)
% -----------------------------------------------------------
% y  = sum sequence over k, which includes k1 and k2

k = min(min(k1),min(k2)):max(max(k1),max(k2)); % duration of  y(k)
y1 = zeros(1,length(k)); y2 = y1;                % initialization
y1(find((k>=min(k1))&(k<=max(k1))==1))=x1;      % x1 with duration y
y2(find((k>=min(k2))&(k<=max(k2))==1))=x2;      % x2 with duration y
y = y1 + y2;                                     % sequence addition
```

Operations: Addition & Multiplication

▪ Signal Multiplication

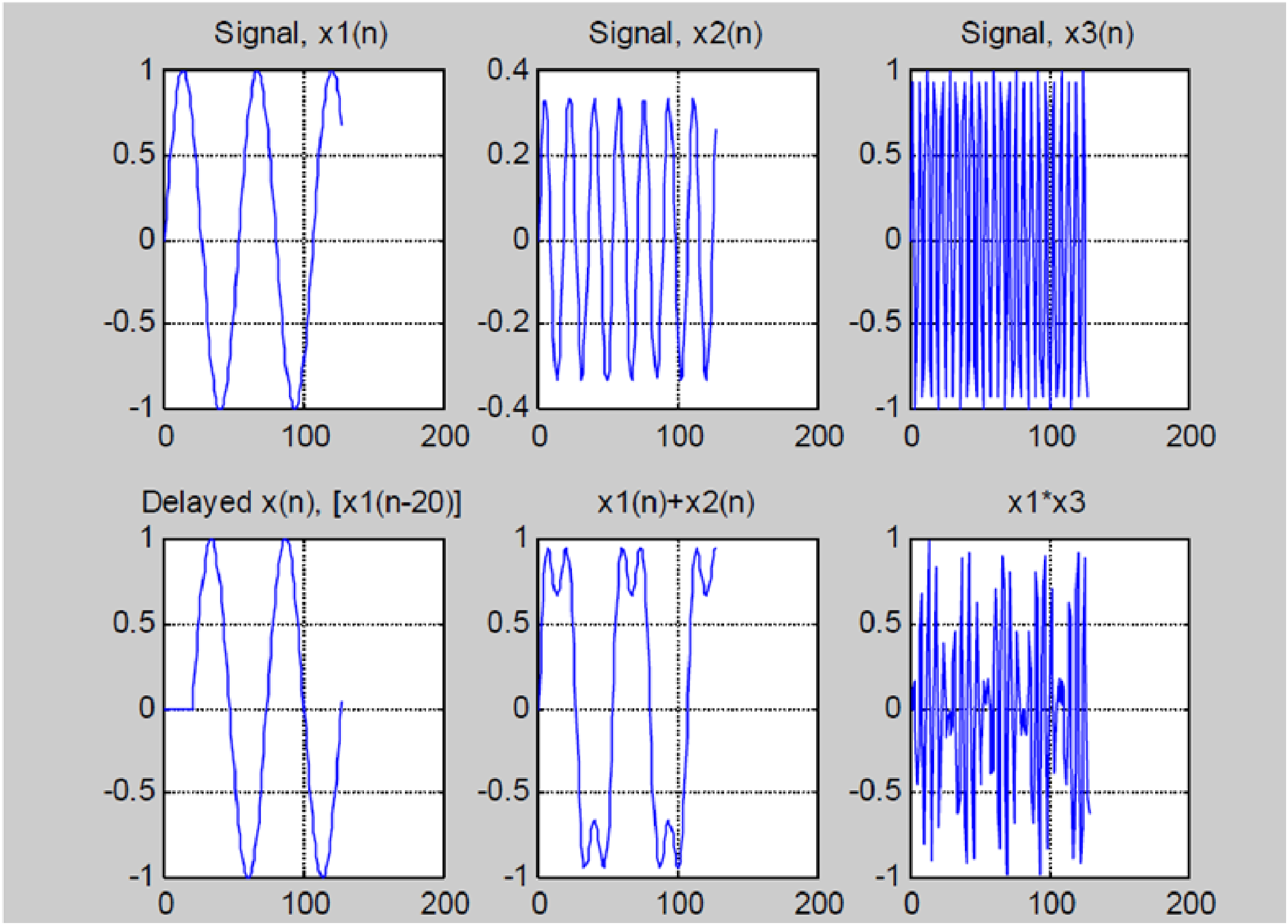
$$\{x_1(k)\}\cdot\{x_2(k)\}=\{x_1(k)x_2(k)\}$$

```
Function [y,k] = sigmult(x1,k1,x2,k2);
% implements  y(k) = x1(k)*x2(k)
% ------------------------------------------------------------
% [y,k] = sigmult(x1,k1,x2,k2)
%   y = product sequence over k, which includes k1 and k2
% x1 = first sequence over k1
% x2 = second sequence over k2 (k2 can be different from k1)

k = min(min(k1),min(k2)):max(max(k1),max(k2));
y1 = zeros(1,length(k)); y2 = y1;
y1(find((k>=min(k1))&(k<=max(k1))==1))=x1;
y2(find((k>=min(k2))&(k<=max(k2))==1))=x2;
y = y1 .* y2;
```

Operations: Addition & Multiplication

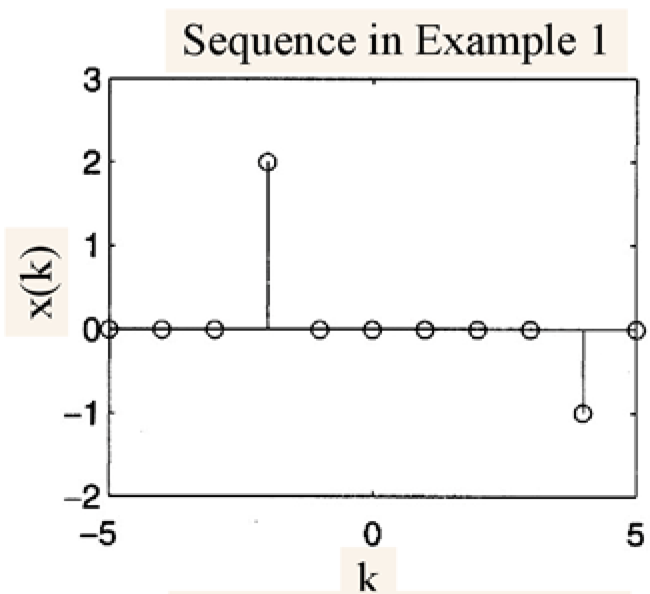
- **Example:**



Example #1:

$$x(k) = 2\delta(k+2) - \delta(k-4), \quad -5 \le k \le 5$$

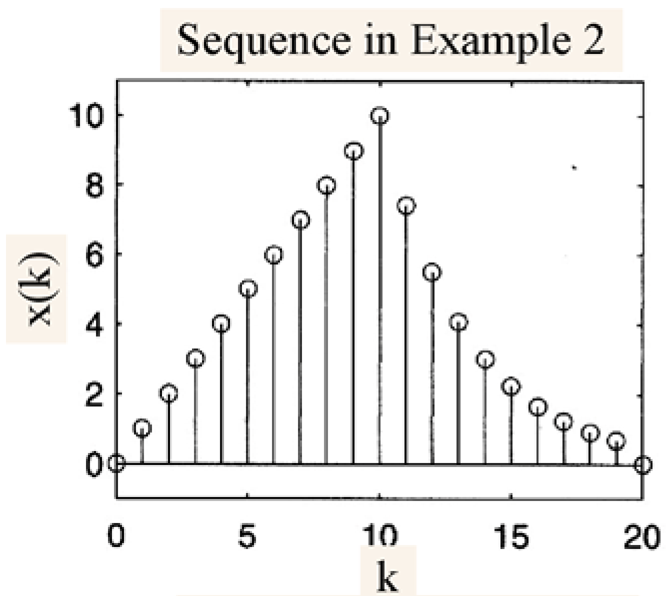
```
>> k = [-5:5];
>> x = 2*impseq(-2,-5,5) – impseq(4,-5,5);
>> subplot(2,2,1); stem(k,x); title('Sequence in Example 1')
>> xlabel('k'); ylabel('x(k)');
```



Example #2:

$$x(k) = k[u(k) - u(k-10)] + 10e^{-0.3(k-10)}[u(k-10) - u(k-20)],$$
$$0 \le k \le 20$$

```
>> k = [0:20]; x1 = k.*(stepseq(0,0,20)-stepseq(10,0,20));
>> x2 = 10*exp(-0.3*(k-10)).*(stepseq(10,0,20)-stepseq(20,0,20));
>> x = x1+x2;
>> subplot(2,2,2); stem(k,x); title('Sequence in Example 2');
>> xlabel('k'); ylabel('x(k)');
```

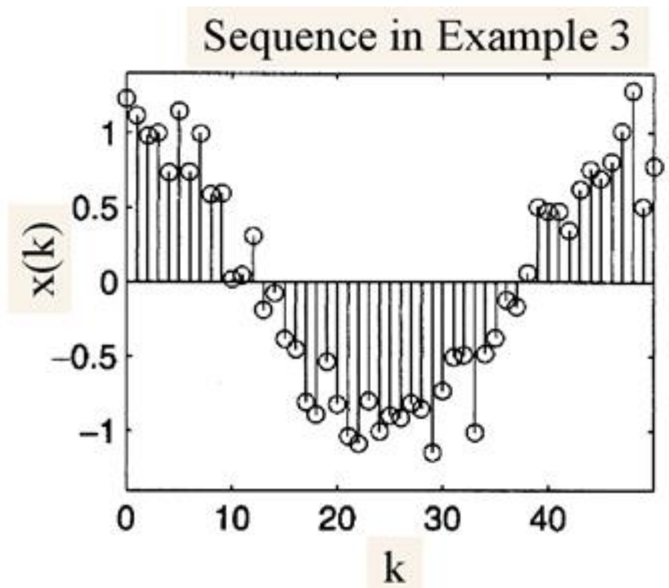


Example #3:

$$x(k) = \cos(0.04\pi k) + 0.2w(k), \quad 0 \le k \le 50$$

A Gaussian random sequence with zero mean and unit variance

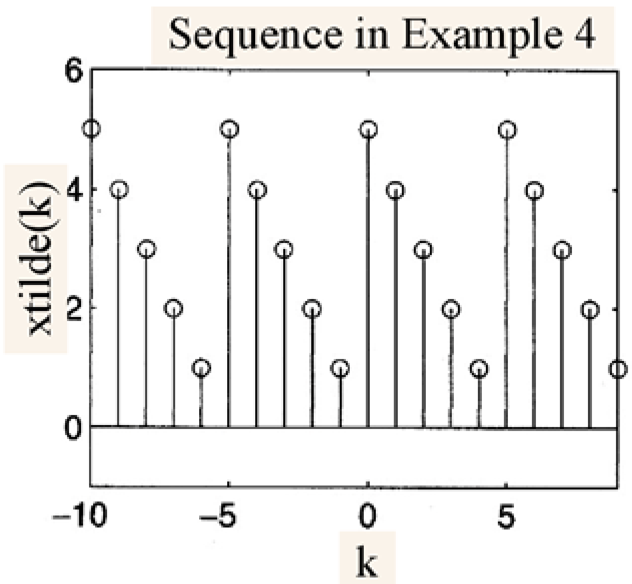
```
>> k = [0:50]; x = cos(0.04*pi*k)+0.2*randn(size(k));
>> subplot(2,2,3); stem(k,x); title('Sequence in Example 3');
>> xlabel('k'); ylabel('x(k)');
```



Example #4:

$$\tilde{x}(k) = \{\ldots,5,4,3,2,1,\underset{\uparrow}{5},4,3,2,1,5,4,3,2,1,\ldots\}, \quad -10 \le k \le 9$$

```
>> k = [-10:9]; x = [5,4,3,2,1];
>> xtilde = x' * ones(1,4); xtilde = (xtilde(:))';
>> subplot(2,2,4); stem(k,xtilde); title('Sequence in Example 4');
>> xlabel('k'); ylabel('xtilde(k)');
```



Example #5:

$$x(k) = \{1,2,\underset{\uparrow}{3},4,5,6,7,6,5,4,3,2,1\}$$

$a.\quad x_1(k) = 2x(k-5) - 3x(k+4)$

$b.\quad x_2(k) = x(3-k) + x(k)x(k-2)$

The sequence $x(k)$ is nonzero over $-2 \le k \le 10$. Hence

```
>> k = -2:10; x = [1:7,6:-1:1];
```

will generates $x(k)$.

Example #5:

a. $x_1(k) = 2x(k-5) - 3x(k+4)$ *shifting* $x(k)$ *by* -4

Obtained by shifting $x(k)$ *by* 5

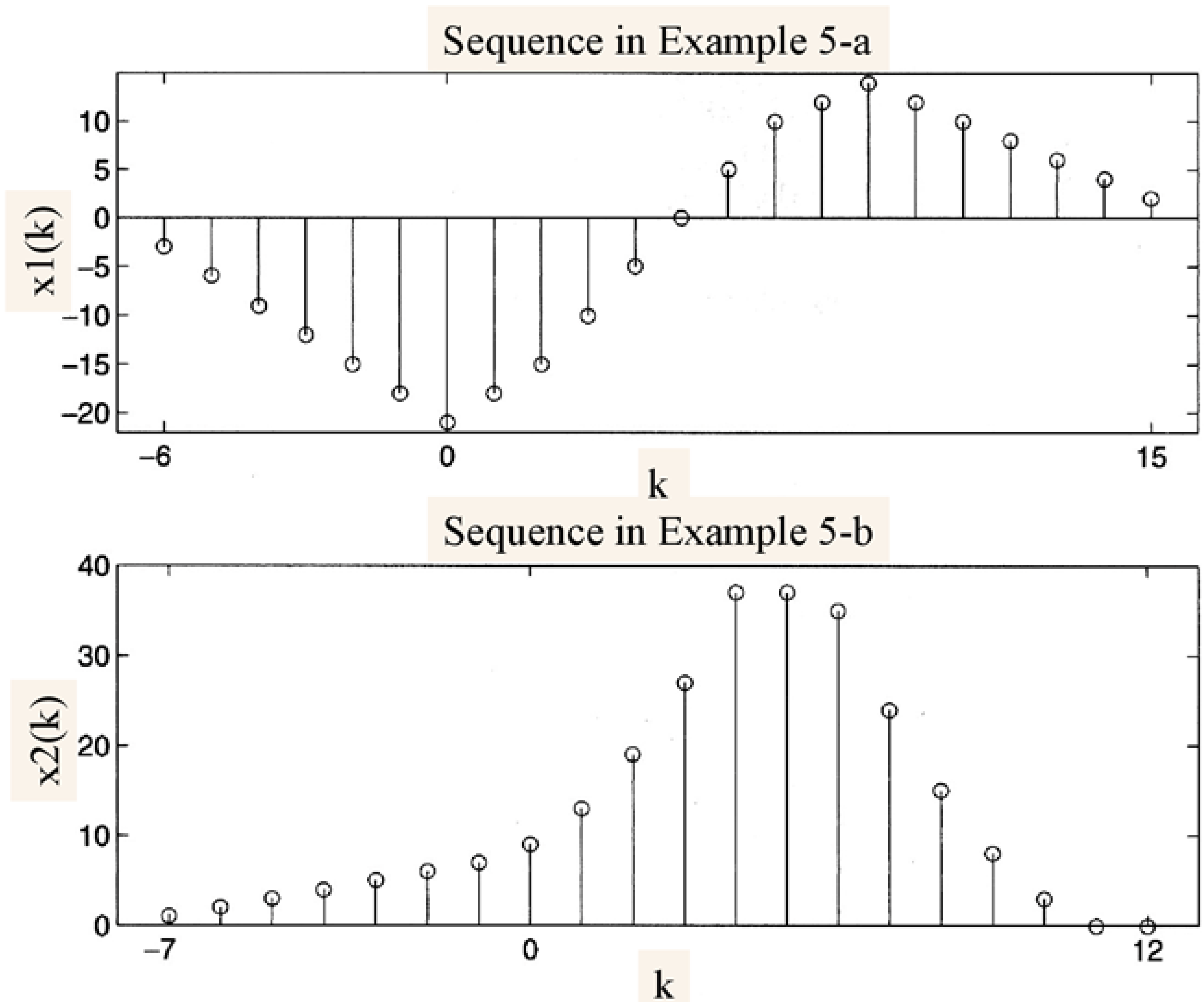
```
>> [x11,k11] = sigshift(x,k,5); [x12,k12] = sigshift(x,k,-4);
>> [x1,k1] = sigadd(2*x11,k11,-3*x12,k12);
>> subplot(2,1,1); stem(k1,x1); title('Sequence in Example 5-1')
>> xlabel('k'); ylabel('x1(k)');
```

b. $x_2(k) = x(3-k) + x(k)x(k-2)$

Can be written as $x(-(k\text{-}3))$

```
>> [x21,k21] = sigfold(x,k); [x21,k21] = sigshift(x21,k21,3);
>> [x22,k22] = sigshift(x,k,2); [x22,k22] = sigmult(x,k,x22,k22);
>> [x2,k2] = sigadd(x21,k21,x22,k22);
>> subplot(2,1,2); stem(k2,x2); title('Sequence in Example 5-2')
>> xlabel('k'); ylabel('x2(k)');
```

Example #5:

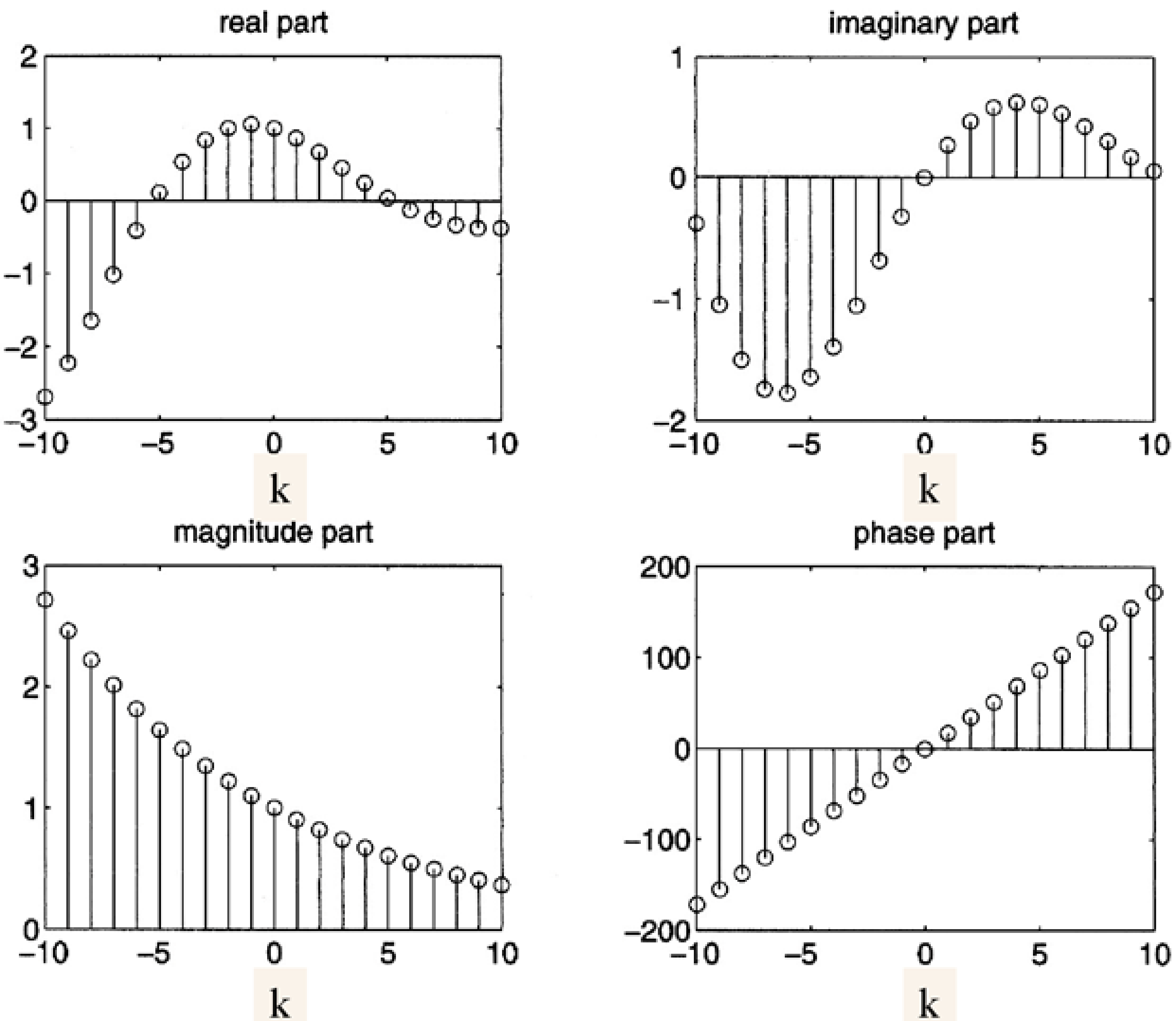

Sequence in Example 5-a
x1(k)
10
5
0
−5
−10
−15
−20
−6
0
15
k
Sequence in Example 5-b
x2(k)
40
30
20
10
0
−7
0
12
k

Example #6:

$$x(k) = e^{(-0.1+j0.3)k}, \quad -10 \le k \le 10$$

```
>> k = [-10:1:10]; alpha = -0.1+0.3j;
>> x = exp(alpha*k);
>> subplot(2,2,1); stem(k, real(k));title('real part');xlabel('k')
>> subplot(2,2,2); stem(k,imag(k));title('imaginary part');xlabel('k')
>> subplot(2,2,3); stem(k,abs(x));title('magnitude part');xlabel('k')
>> subplot(2,2,4); stem(k,(180/pi)*angle(k));title('phase part');xlabel('k')
```

Example #6:



real part
imaginary part
magnitude part
phase part
2
1
0
−1
−2
−3
−10
−5
0
5
10
k
−2
3
200
100
−100
−200

Example #7:

Unit sample synthesis

Any arbitrary sequence $$x(k) = \sum_{m=-\infty}^{\infty} x(m)\delta(k-m)$$ *A weighted sum of delayed and scaled unit sample sequences*

Even and odd synthesis

A real-valued sequence

$$x_e(-k) = x_e(k)$$ *even (symmetric)*

$$x_o(-k) = -x_o(k)$$ *odd (antisymmetric)*

$$x(k) = x_e(k) + x_o(k)$$ *Can be decomposed into its even and odd components*

$$x_e(k) = \frac{1}{2}[x(k) + x(-k)], \quad x_o(k) = \frac{1}{2}[x(k) - x(-k)]$$

Example #7:

```
function [xe, xo, c] = evenodd(x,k);
% Real signal decomposition into even and odd parts
% -----------------------------------------------------------
% [xe, xo, c] = evenodd(x,k)
%
if any(imag(x) ~= 0)
        error('x is not a real sequence')
end
c = -fliplr(k)
c1 = min([c,k]); c2 = max([c,k]); c = c1:c2;
kc = k(1)-c(1); k1 = 1:length(k);
x1 = zeros(1,length(c)); x1(k1+kc) = x; x = x1;
xe = 0.5*(x + fliplr(x)); xo = 0.5*(x – fliplr(x));
```

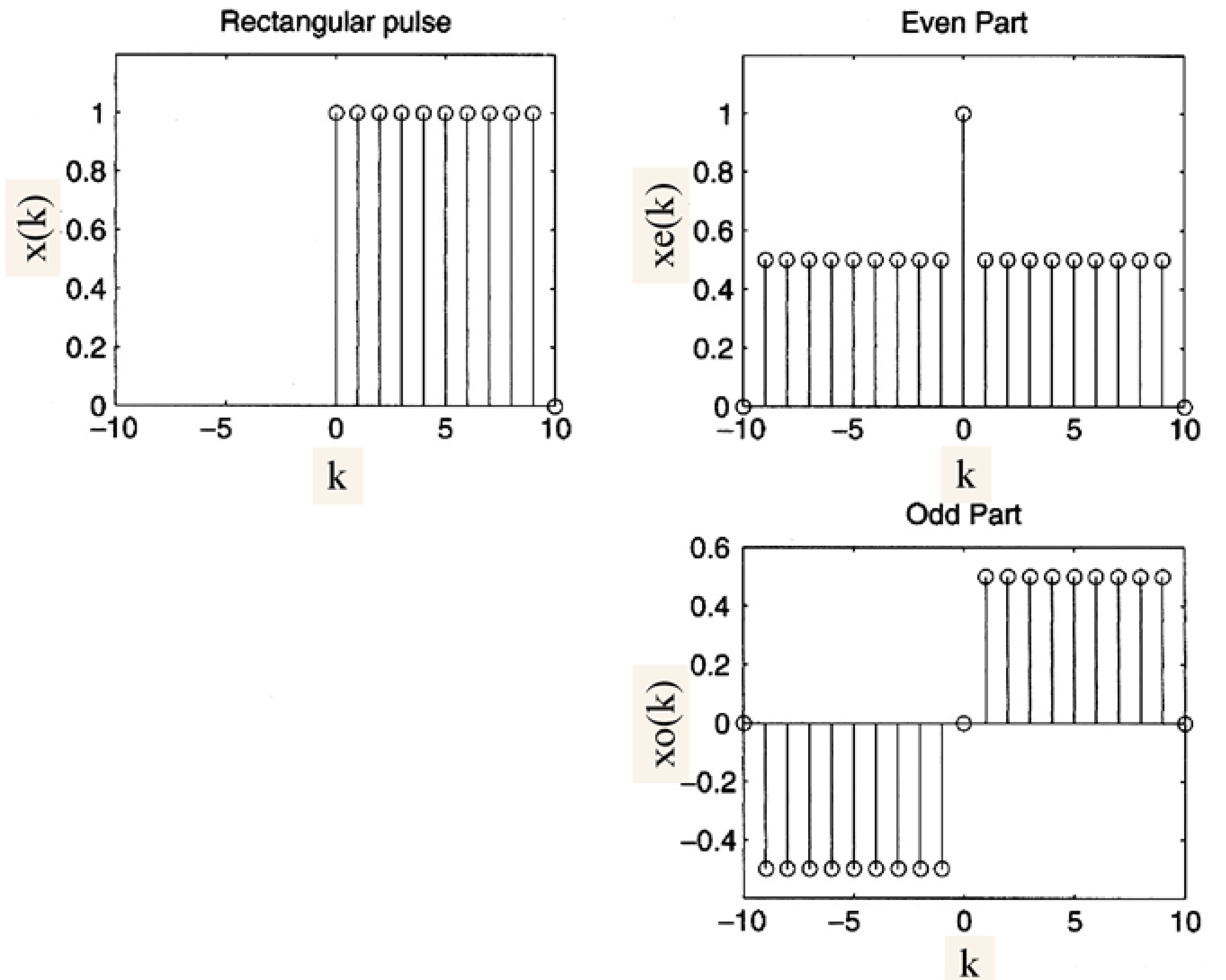
Example #7:

$x(k) = u(k) - u(k-10)$. *Decompose* $x(k)$ *into even and odd components.*

The sequence $x(k)$, *which is nonzero over* $0 \le k \le 9$, *is called a rectangular pulse.*

```
>> k = [1:10]; x = stepseq(0,0,10)-stepseq(10,0,10);
>> [xe,xo,c] = evenodd(x,k);
>> subplot(2,2,1); stem(k,x); title('Rectangular pulse')
>> xlabel('k'); ylabel('x(k)'); axis([-10,10,0,1.2])
>> subplot(2,2,2); stem(c,xe); title('Even part');
>> xlabel('k'); ylabel('xe(k)'); axis([-10,10,0,1.2])
>> subplot(2,2,4); stem(c,xo); title('Odd part');
>> xlabel('k'); ylabel('xo(k)'); axis([-10,10,-0.6,0.6])
```

Example #7:



Rectangular pulse
x(k)
1
0.8
0.6
0.4
0.2
0
−10
−5
0
5
10
k
Even Part
xe(k)
1
0.8
0.6
0.4
0.2
0
−10
−5
0
5
10
k
Odd Part
xo(k)
0.6
0.4
0.2
0
−0.2
−0.4
−10
−5
0
5
10
k

Discrete Systems

Discrete Systems

A discrete system is described as an operator $T[\cdot]$ that takes a sequence $x(k)$ and transforms it into another sequence $y(k)$.

called excitation

called response

$$y(k) = T[x(k)]$$

In DSP, we will say that the system processes an *input* signal into an *output* signal.

Linear Systems

A discrete system $T[\cdot]$ is a linear operator $L[\cdot]$ *iff* $L[\cdot]$ satisfies the principle of superposition, namely,

$$L[\alpha_1 x_1(k) + \alpha_2 x_2(k)] = \alpha_1 L[x_1(k)] + \alpha_2 L[x_2(k)], \forall \alpha_1, \alpha_2, x_1(k), x_2(k)$$

Using $x(k) = \sum_{m=-\infty}^{\infty} x(m)\delta(k-m)$, *Unit sample synthesis*

impulse response $g(k,m)$

$$y(k) = L[x(k)] = L\left[\sum_{m=-\infty}^{\infty} x(m)\delta(k-m)\right] = \sum_{m=-\infty}^{\infty} x(m)\boxed{L[\delta(k-m)]}$$

$$= \sum_{m=-\infty}^{\infty} x(m)g(k,m)$$

superposition summation

The response at time k
due to a unit sample at time m

Linear Time Invariant (LTI) Systems

A linear system in which an input-output pair, $x(k)$ and $y(k)$, is invariant to a shift m in time is called an LTI system.

$$y(k) = L[x(k)] \Rightarrow L[x(k-m)] = y(k-m)$$

For an LTI system the $L[\cdot]$ and the shifting operators are reversible as shown below.

$$x(k) \longrightarrow \boxed{L[\cdot]} \longrightarrow y(k) \longrightarrow \boxed{\text{Shifted by } m} \longrightarrow y(k\text{-}m)$$

$$x(k) \longrightarrow \boxed{\text{Shifted by } m} \longrightarrow x(k\text{-}m) \longrightarrow \boxed{L[\cdot]} \longrightarrow y(k\text{-}m)$$

Example:

Determine whether the following systems are linear:

$$y(k) = 3x^2(k)$$

Determine whether the following linear systems are time-invariant:

$$y(k) = 10\sin(0.1\pi k)x(k)$$

Stability

To avoid building harmful systems or to avoid burnout or saturation in the system operation

A system is said to be bounded-input bounded-output (BIBO) *stable if every bounded input produces a bounded output.*

$$|x(k)| < \infty \Rightarrow |y(k)| < \infty, \forall x, y$$

An LTI *system is* BIBO *stable iff its impulse response is absolutely summable.*

$$\text{BIBO Stability} \Leftrightarrow \sum_{-\infty}^{\infty} |g(k)| < \infty$$

Stability

A system is said to be causal if the output at index k_0 *depends only on the input up to and including the index* k_0*; that is the output does not depend on the future values of the input.*

An LTI system is causal iff its impulse response

$$g(k) = 0, \quad k < 0$$

Such a sequence is termed a causal sequence.

Convolution

Let the rectangular pulse $x(k)=u(k)-u(k-10)$ be an input to an LTI system with impulse response

$$g(k) = (0.9)^k u(k)$$

Determine the output $y(k)$.

$$y(k) = \sum_{m=0}^{9} (1)(0.9)^{(k-m)} u(k-m)$$

$$= (0.9)^k \sum_{m=0}^{9} (0.9)^{-m} u(k-m)$$

Convolution

CASE i $k < 0$

$$u(k-m) = 0, \quad 0 \le m \le 9$$

$$y(k) = 0$$ *The nonzero values of* $x(k)$ *and* $g(k)$ *do not overlap.*

CASE ii $0 \le k < 9$

$$u(k-m) = 1, \quad 0 \le m \le k$$

$$y(k) = (0.9)^k \sum_{m=0}^{k} (0.9)^{-m} = (0.9)^k \sum_{m=0}^{k} \left[(0.9)^{-1}\right]^m$$

$$= (0.9)^k \frac{1-(0.9)^{-(k+1)}}{1-(0.9)^{-1}} = 10\left[1-(0.9)^{k+1}\right], \quad 0 \le k < 9$$

The impulse response $g(k)$ *partially overlaps the input* $x(k)$.

Convolution

CASE iii $k \geq 9$

$$u(k-m)=1, \quad 0 \leq m \leq 9$$

$$y(k)=(0.9)^k \sum_{m=0}^{9} (0.9)^{-m}$$

$$=(0.9)^k \frac{1-(0.9)^{-10}}{1-(0.9)^{-1}} = 10(0.9)^{k-9}\left[1-(0.9)^{10}\right], \quad k \geq 9$$

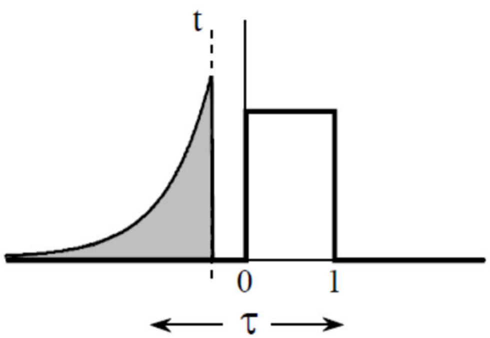
$g(k)$ *completely overlaps* $x(k)$.

Review Class 2: Convolution

Calculating Convolution by Segments

No overlap

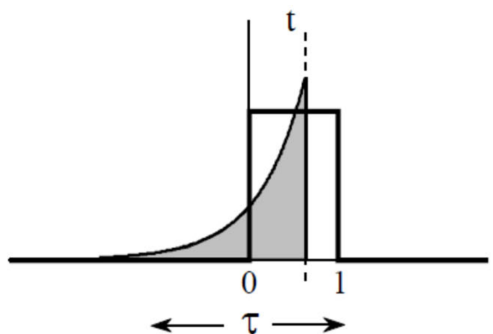
$(t < 0)$



$$y(t) = 0, \quad t < 0$$

Partial overlap

$(0 \le t \le 1)$



$$y(t) = \int_{-\infty}^{\infty} u(\tau) g(t-\tau) d\tau$$

$$= \int_{0}^{t} 1 \cdot \alpha e^{-\alpha(t-\tau)} d\tau$$

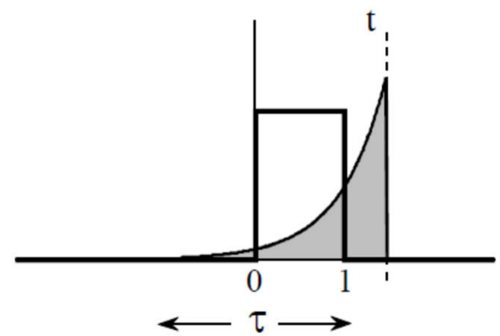
$$= e^{-\alpha t} [e^{\alpha\tau}] \Big|_{0}^{t}$$

$$= e^{-\alpha t} [e^{\alpha t} - 1]$$

$$= 1 - e^{-\alpha t}, \quad 0 \le t \le 1$$

Full overlap

$(t > 1)$



$$y(t) = \int_{-\infty}^{\infty} u(\tau) g(t-\tau) d\tau$$

$$= \int_{0}^{1} 1 \cdot \alpha e^{-\alpha(t-\tau)} d\tau$$

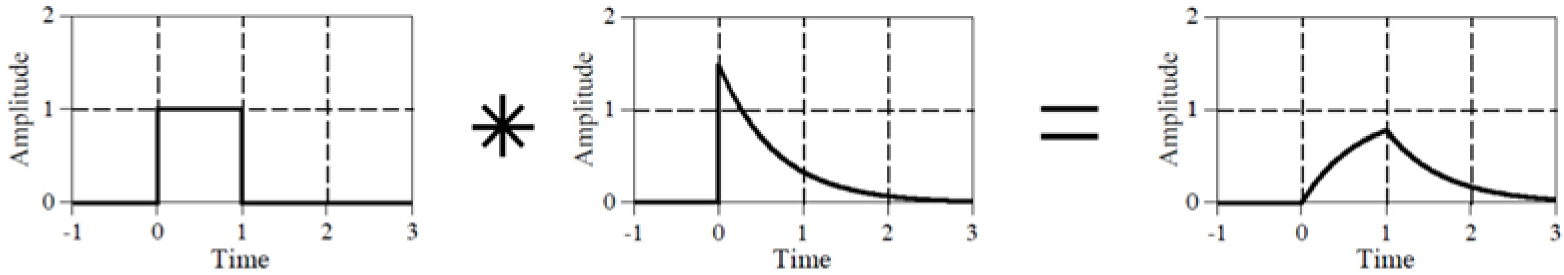
$$= e^{-\alpha t} [e^{\alpha\tau}] \Big|_{0}^{1}$$

$$= [e^{\alpha} - 1] e^{-\alpha t}, \quad t > 1$$

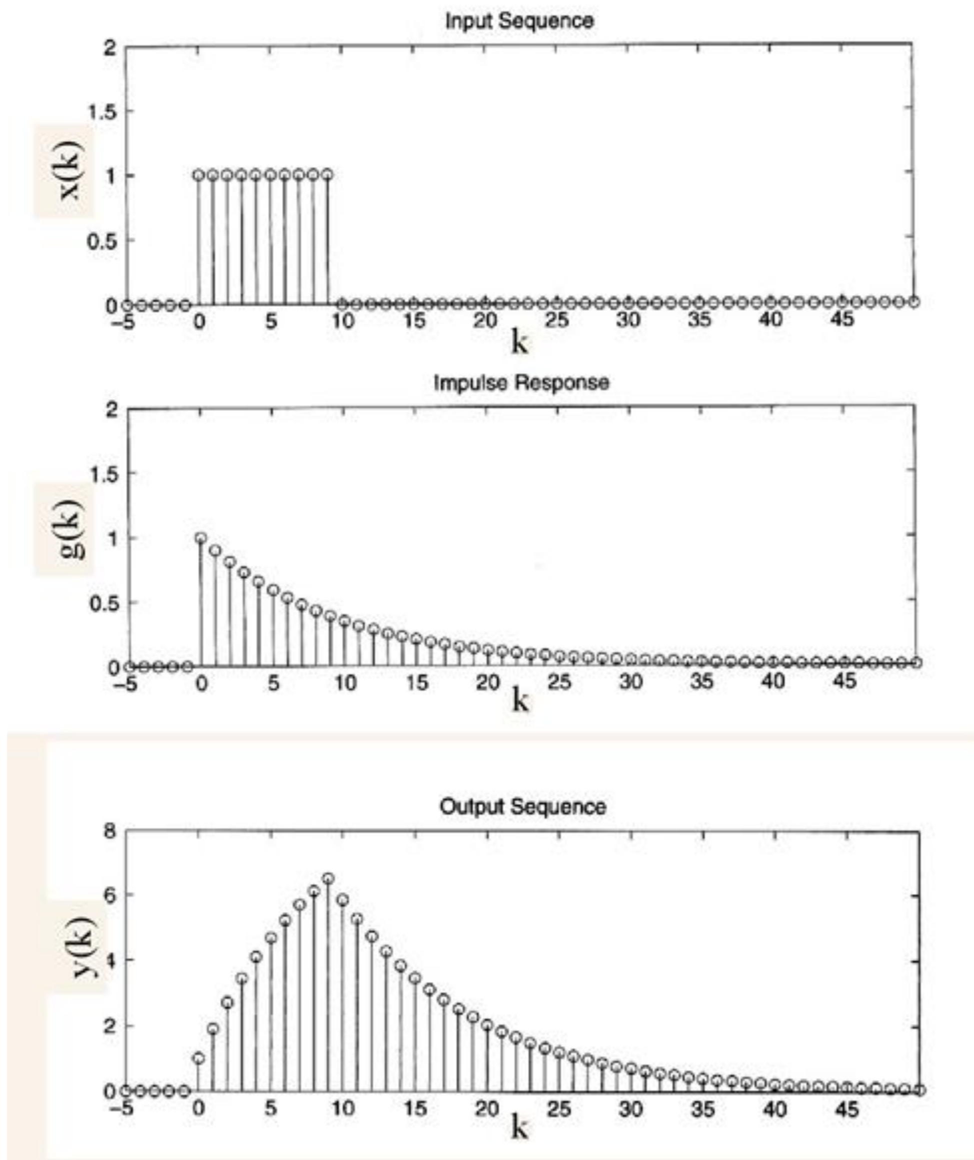
$$g(t) = 0, \quad t < 0$$

$$g(t) = \alpha e^{-\alpha t}, \quad t \ge 0$$

Review Class 2: Convolution



Convolution



Example #8:

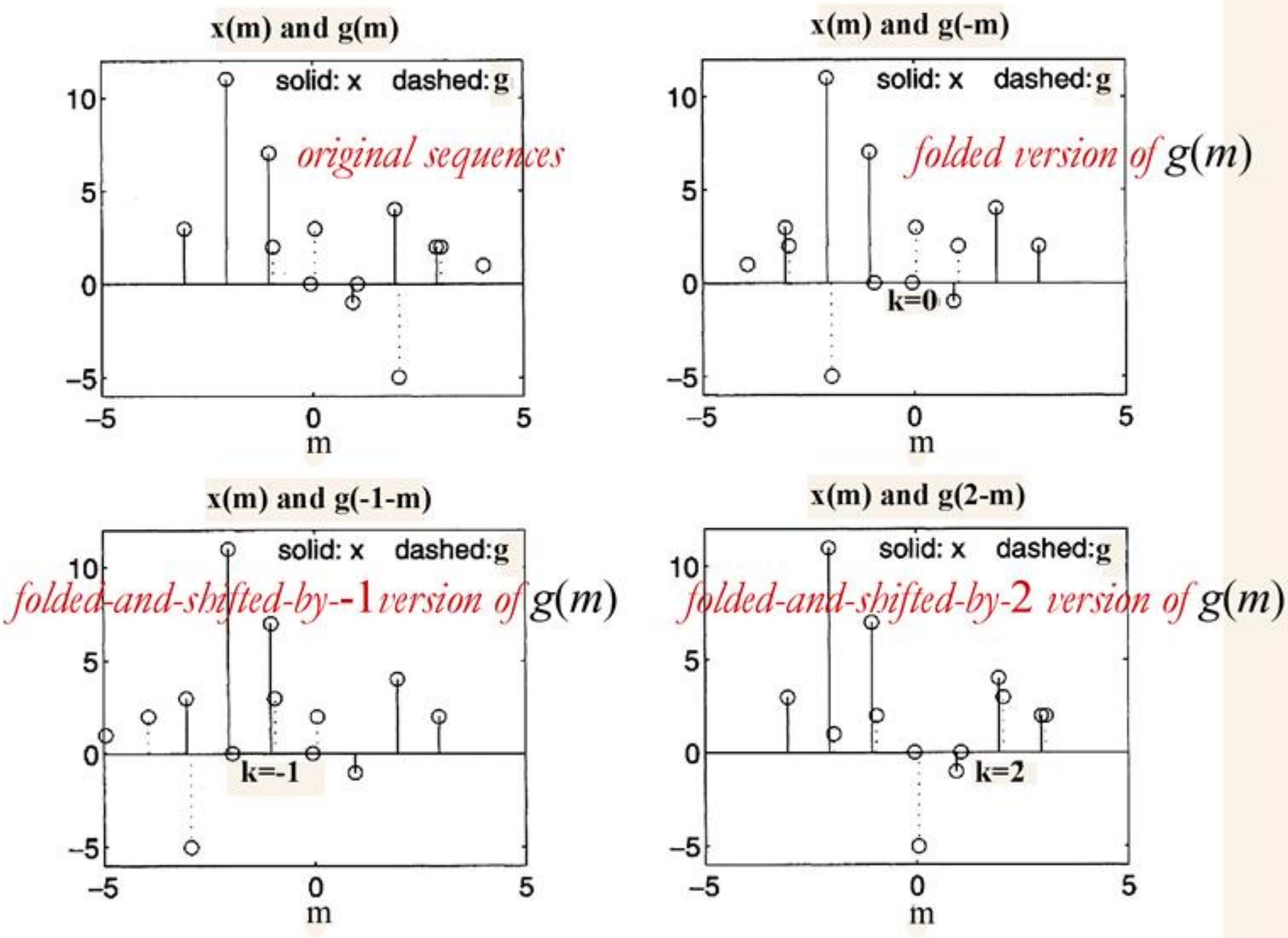
$$x(k) = [3, 11, 7, \underset{\uparrow}{0}, -1, 4, 2], \quad -3 \le k \le 3$$

$$g(k) = [2, \underset{\uparrow}{3}, 0, -5, 2, 1], \quad -1 \le k \le 4$$

Determine the convolution $y(k) = x(k) * g(k)$

```
>> x = [3, 11, 7, 0, -1, 4, 2]; g = [2, 3, 0, -5, 2, 1];
>> y = conv(x, g)
y =
   6   31   47    6   -51   -5   41   18  -22   -3    8    2
```

Example #8:



x(m) and g(m)
solid: x dashed: g
original sequences
x(m) and g(-m)
solid: x dashed: g
folded version of g(m)
k=0
x(m) and g(-1-m)
solid: x dashed: g
folded-and-shifted-by--1 version of g(m)
k=-1
x(m) and g(2-m)
solid: x dashed: g
folded-and-shifted-by-2 version of g(m)
k=2
m

Example #8:

The conv *function neither provides nor accepts any timing information if the sequences have arbitrary support.*

$$\{x(k);\ \ k_{xb} \le k \le k_{xe}\},\ \ \{g(k);\ \ k_{gb} \le k \le k_{ge}\}$$

The beginning and end points of $y(k)$ are

$$k_{yb} = k_{xb} + k_{gb},\ \ k_{ye} = k_{xe} + k_{ge}$$

```
Function [y,ky] = conv_m(x,kx,g,kg)
% Modified convolution routine for signal processing
% ---------------------------------------------------------------
% [y,ky] = conv_m(x,kx,g,kg)
% [y,ky] = convolution results
% [x,kx] = first signal
% [g,kg] = second signal
%
kyb = kx(1)+kg(1); kye = kx(length(x)) + kg(length(g));
ky = [kyb:kye]; y = conv(x,g);
```

Example #8:

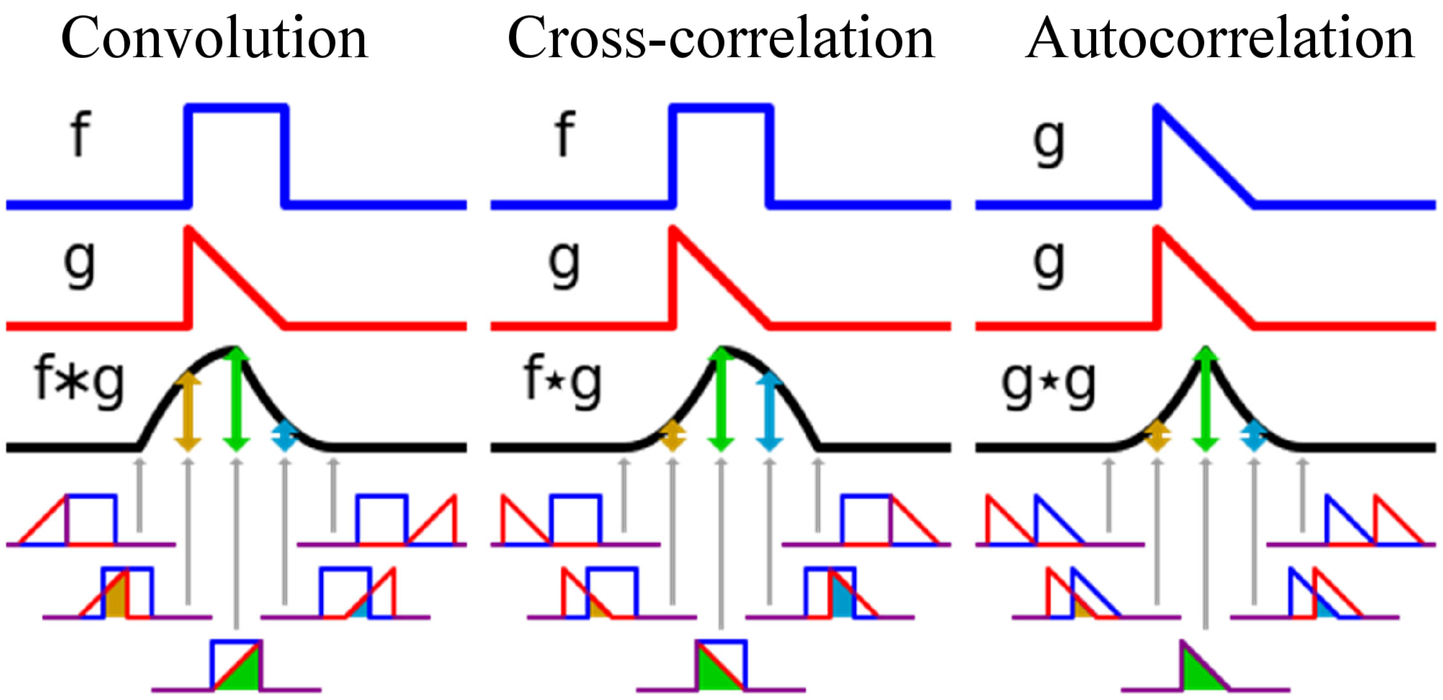
using the conv_m function

```
>> x = [3, 11, 7, 0, -1, 4, 2]; kx = [-3:3];
>> g = [2, 3, 0, -5, 2, 1]; kg = [-1:4];
```

```
>> [y,ky] = conv_m(x,kx,g,kg)
y =
    6   31   47    6  -51   -5   41   18  -22   -3    8    2
ky =
   -4   -3   -2   -1    0    1    2    3    4    5    6    7
```

Cross-Correlation & Auto-Correlation

"measuring the similarity"



$$\rho_{xy}(l) = \frac{r_{xy}(l)}{\sqrt{r_{xx}(0) r_{yy}(0)}}$$

$$\rho_{xx}(l) = \frac{r_{xx}(l)}{r_{xx}(0)}$$

normalized cross-correlation *normalized autocorrelation*

Cross-Correlation & Auto-Correlation

$x(k), y(k)$

each of which has finite energy

Cross-correlation

$$r_{xy}(l) = \sum_{k=-\infty}^{\infty} x(k)y(k-l), \quad l = 0, \pm 1, \pm 2, \ldots$$

$$r_{xy}(l) = \sum_{k=-\infty}^{\infty} x(k+l)y(k), \quad l = 0, \pm 1, \pm 2, \ldots$$

$$r_{yx}(l) = \sum_{k=-\infty}^{\infty} y(k)x(k-l)$$

$$r_{yx}(l) = \sum_{k=-\infty}^{\infty} y(k+l)x(k)$$

$$r_{xy}(l) = r_{yx}(-l)$$

Autocorrelation

$$r_{xx}(l) = \sum_{k=-\infty}^{\infty} x(k)x(k-l)$$

$$r_{xx}(l) = \sum_{k=-\infty}^{\infty} x(k+l)x(k)$$

Cross-Correlation & Auto-Correlation

$$x(k)=\left\{\ldots,0,0,2,-1,3,7,\underset{\uparrow}{1},2,-3,0,0,\ldots\right\}$$

$$y(k)=\left\{\ldots,0,0,1,-1,2,-2,\underset{\uparrow}{4},1,-2,5,0,0,\ldots\right\}$$

$$r_{xy}(0)=\sum_{k=-\infty}^{\infty}x(k)y(k)=\left\{\ldots,0,0,2,1,6,-14,\underset{\uparrow}{4},2,6,0,0,\ldots\right\}=0$$

$$r_{xy}(1)=\sum_{k=-\infty}^{\infty}x(k)y(k-1)=\left\{\ldots,0,0,0,-1,-3,14,\underset{\uparrow}{-2},8,-3,0,0,\ldots\right\}=13$$

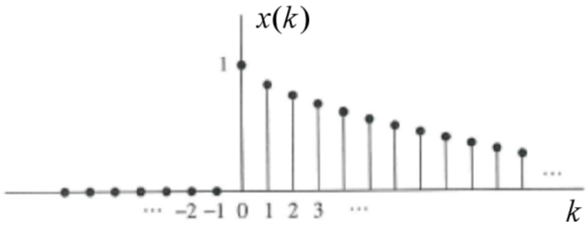
$$r_{xy}(2)=-18,\quad\cdots,\quad r_{xy}(6)=-3,\quad r_{xy}(l)=0, l\geq 7$$

$$r_{xy}(-1)=\sum_{k=-\infty}^{\infty}x(k)y(k+1)=0$$

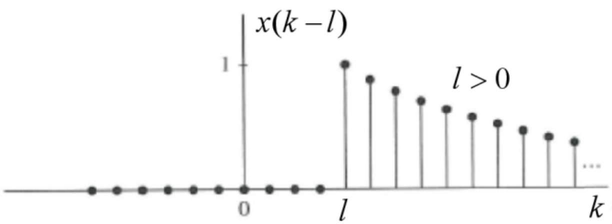
$$r_{xy}(-2)=33,\quad\cdots,\quad r_{xy}(-7)=10,\quad r_{xy}(l)=0, l\leq -8$$

$$r_{xy}=\left\{10,-9,19,36,-14,33,0,\underset{\uparrow}{7},13,-18,16,-7,5,-3\right\}$$ *the cross-correlation sequence*

Cross-Correlation & Auto-Correlation



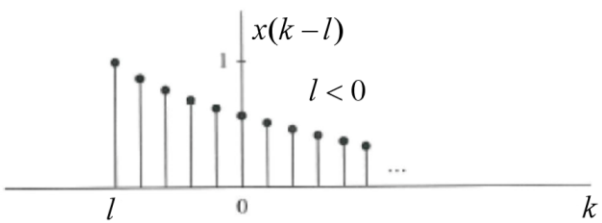
$$x(k) = a^k u(k), \quad 0 < a < 1$$



$l \geq 0$

$$r_{xx}(l) = \sum_{k=l}^{\infty} x(k)x(k-l) = \sum_{k=l}^{\infty} a^k a^{k-l} = a^{-l} \sum_{k=l}^{\infty} \left(a^2\right)^k$$

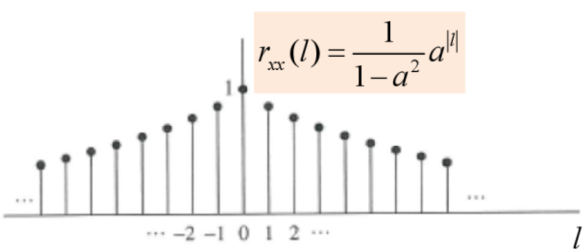
$$= \frac{1}{1-a^2} a^{|l|}, \quad l \geq 0$$



$l < 0$

$$r_{xx}(l) = \sum_{k=0}^{\infty} x(k)x(k-l) = \sum_{k=0}^{\infty} a^k a^{k-l} = a^{-l} \sum_{k=0}^{\infty} \left(a^2\right)^k$$

$$= \frac{1}{1-a^2} a^{-l}, \quad l < 0$$



$$\boxed{r_{xx}(l) = \frac{1}{1-a^2} a^{|l|}, \quad -\infty < l < \infty}$$ *autocorrelation*

$$r_{xx}(-l) = r_{xx}(l), \quad r_{xx}(0) = \frac{1}{1-a^2}$$

Sequence Correlation

The crosscorrelation $r_{yx}(l)$ can be put in the form

$$r_{yx}(l) = y(l) * x(-l)$$

with the autocorrelation $r_{xx}(l)$ in the form

$$r_{xx}(l) = x(l) * x(-l)$$

Therefore the correlations can be computed using the conv_m *function if sequences are of finite duration.*

Lab #5.1 (1 pt.): Due Jan 10

$$x(k) = [A, 11, B, \underset{\uparrow}{0}, -1, 4, C]$$ *a prototype sequence*

$$y(k) = x(k - D) + w(k)$$ *noise-corrupted-and-shifted version*

$w(k)$ *is Gaussian sequence with* $\boxed{\textit{mean } 0}$ *and* $\boxed{\textit{variance } 1}$. *Compute the cross-correlation between* $y(k)$ *and* $x(k)$.

location of the peak

the width of the distribution

the standard normal

Use Your ID: sGFEDCBA

Example #9:

$$x(k) = [3, 11, 7, \underset{\uparrow}{0}, -1, 4, 2]$$

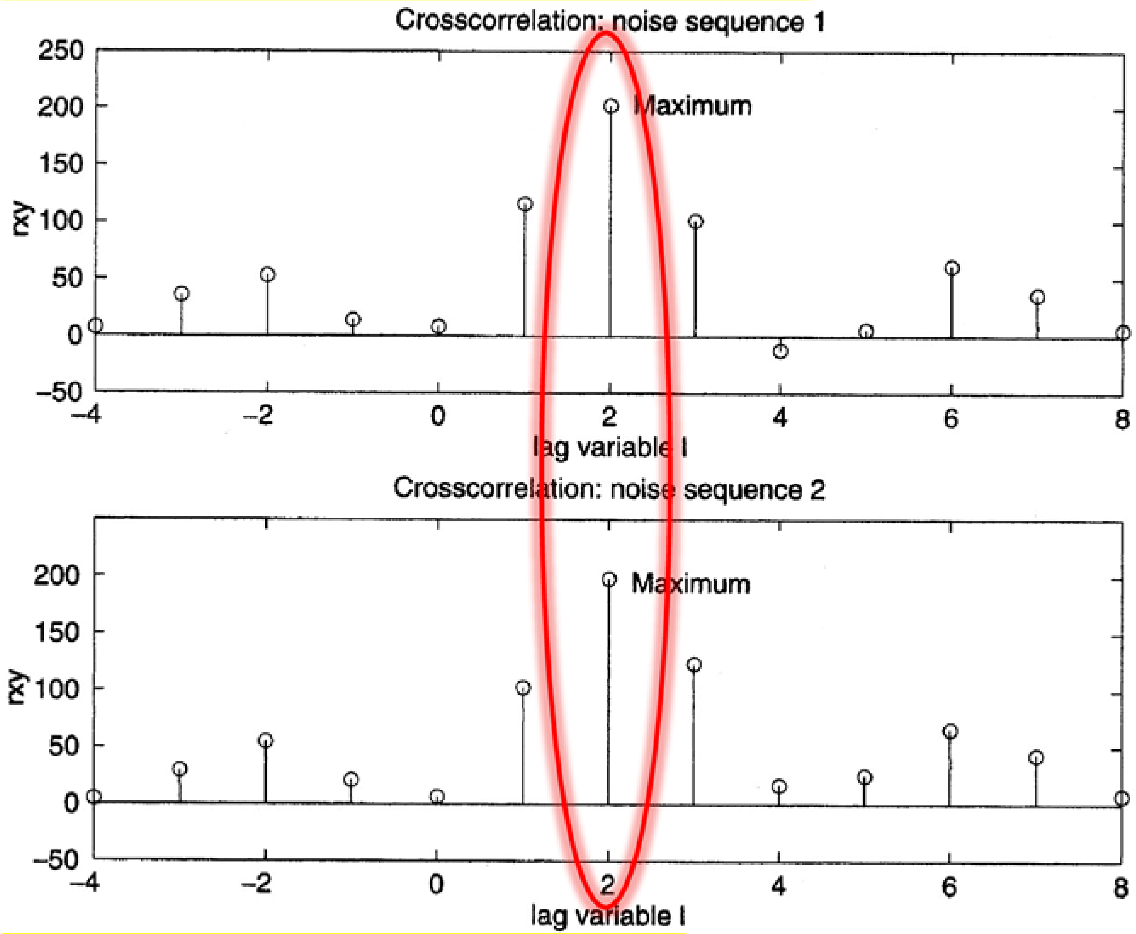
$$y(k) = x(k-2) + w(k)$$

```
>> x = [3, 11, 7, 0, -1, 4, 2]; nx=[-3:3];    % given signal x(k)
>> [y, ny] = sigshift(x, nx, 2);              % obtain x(k-2)
>> w = randn(1, length(y)); nw = ny;          % generate w(k)
>> [y, ny] = sigadd(y, ny, w, nw);            % obtain y(k)=x(k-2)+w(k)
>> [x, nx] = sigfold(x, nx);                  % obtain x(-k)
>> [rxy, nrxy] = conv_m(y, ny, x, nx);        % crosscorrelation
```

`>> xcorr(x, y)`	The crosscorrelation between vectors x and y.
`>> xcorr(x)`	The autocorrelation of vector x.

Example #9:

The cross-correlation indeed peaks at $l=2$.



$y(k)$ is similar to $x(k)$ shifted by 2.

Difference Equations

An LTI discrete system can also be described by a linear constant coefficient difference equation of the form

$$\sum_{m=0}^{N} a_m y(k-m) = \sum_{n=0}^{M} b_n x(k-n), \quad \forall k$$

$$y(k) = \sum_{n=0}^{M} b_n x(k-n) - \sum_{m=1}^{N} a_m y(k-m)$$

A solution to this equation can be obtained in the form

$$y(k) = y_H(k) + y_P(k)$$

Difference Equations

To solve difference equations numerically, given the input and the difference equation coefficients.

```
y = filter(b,a,x)

where

b = [b0, b1, …, bM]; a = [a0, a1, …, aN];
```

To compute and plot impulse response,

```
g = impz(b,a,k);
```

Lab #5.2 (1 pt.): Due Jan 10

$$y(k) - y(k-A) + 0.9y(k-B) = x(k); \quad \forall k$$

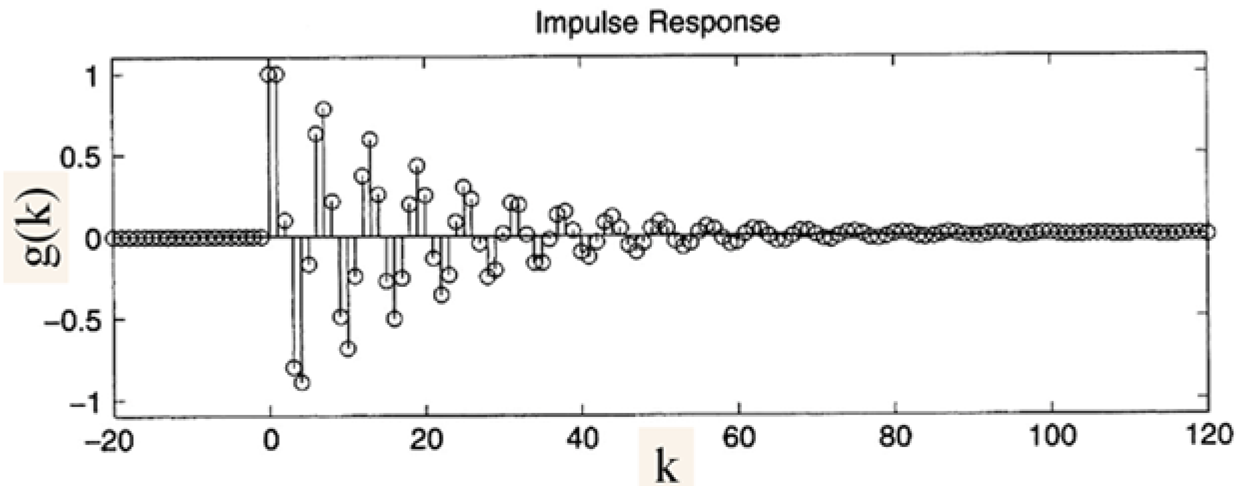
Calculate and plot the *impulse response* $g(k)$ at k = -20,…,120.

Is the system specified by $g(k)$ stable?

Use Your ID: sGFEDCBA

Example #10:

$$y(k) - y(k-1) + 0.9y(k-2) = x(k); \quad \forall k$$



Example #11:

$$x(k) = u(k) - u(k-10)$$

$$g(k) = (0.9)^k u(k)$$

If one or both sequences in the convolution are of infinite length, then the conv *function cannot be used.*

```
>> b = [1]; a = [1,-0.9];
>> k = -5:50; x = stepseq(0,-5,50) – stepseq(10,-5,50);
>> y = filter(b,a,x);
>> subsplot(2,1,2); stem(k,y); title('Output sequence')
>> xlabel('k'); ylabel('y(k)'); axis([-5,50,-0.5,8])
```

Example #11:

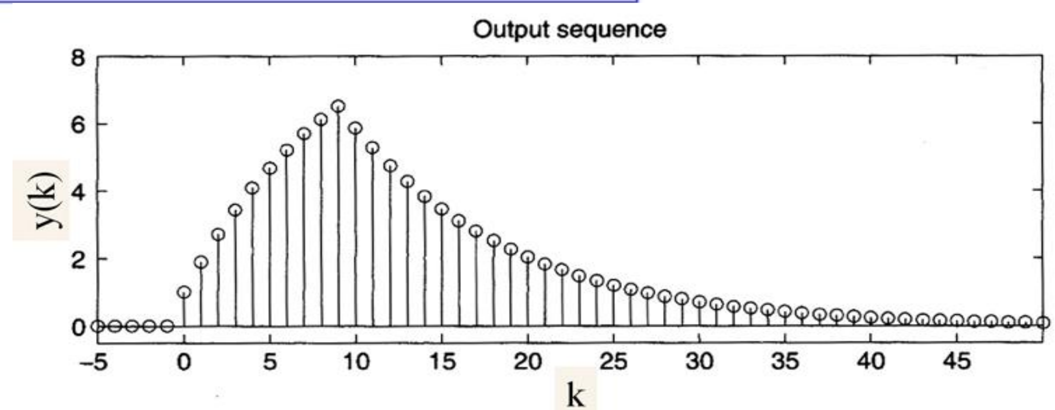
The LTI system, given by the impulse response g(k), *can be described by a difference equation:*

$$(0.9)g(k-1) = (0.9)(0.9)^{k-1}u(k-1) = (0.9)^k u(k-1)$$

$$\begin{aligned} g(k)-(0.9)g(k-1) &= (0.9)^k u(k)-(0.9)^k u(k-1) \\ &= (0.9)^k \left[u(k)-u(k-1)\right] \\ &= (0.9)^k \delta(k) \\ &= \delta(k) \end{aligned}$$

g(k) *is the output of an LTI system, when the input is* $\delta(k)$.

$$y(k)-0.9y(k-1)=x(k)$$



Zero-Input Zero-State Response

The difference equation is generally solved forward in time from $k=0$*. Therefore initial conditions on* $x(k)$ *and* $y(k)$ *are necessary to determine the output for* $k \geq 0$*. The difference equation is then given by*

$$y(k) = \sum_{n=0}^{M} b_n x(k-n) - \sum_{m=1}^{N} a_m y(k-m); \quad k \geq 0$$

subject to the initial conditions:

$$\{y(k); -N \leq k \leq -1\}, \quad \{x(k); -M \leq k \leq -1\}$$

A solution can be obtained in the form

$$y(k) = \underbrace{\boxed{y_{ZI}(k)}}_{\textit{the zero-input solution}} + \underbrace{\boxed{y_{ZS}(k)}}_{\textit{the zero-state solution}}$$

filter

Digital Filters

Digital Filters

- Filter is generic name that means a linear time-invariant system designed for a specific job of frequency selection or frequency discrimination.
- Discrete-time LTI systems are called digital filters: FIR filter, IIR filter

FIR Filters

If the unit impulse response of an LTI system is of finite duration, then the system is called a *finite-duration impulse response* (or FIR) filter. Hence for an FIR filter

$$g(k) = 0 \;\; for \;\; k < k_1 \;\; and \;\; for \;\; k > k_2.$$

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

IIR Filters

If the unit impulse response of an LTI system is of infinite duration, then the system is called an *infinite-duration impulse response* (or IIR) filter.

$$\sum_{m=0}^{N} a_k y(k-m) = x(k)$$

Thank you

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY
1990