Discrete-Time Signals and Systems

EE4015 Digital Signal Processing

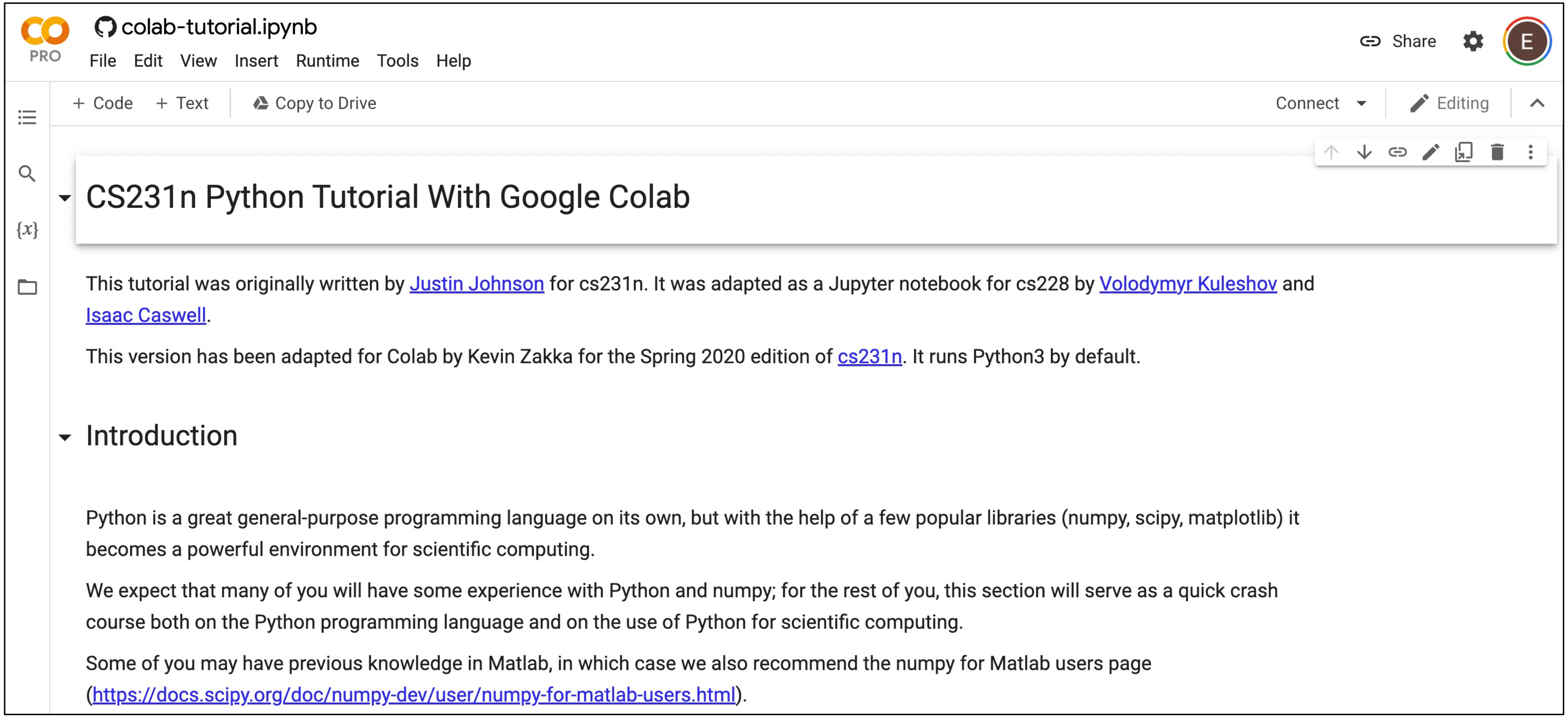
Dr. Lai-Man Po
Department of Electrical Engineering
City University of Hong Kong

Week 2 Messages

- Recommended Technical Presentation for Group Project Development on "Upscaling Images with Neural Networks" by Geoffrey Litt
  - https://www.youtube.com/watch?v=RhUmSeko1ZE
  - This is a great technical presentation for students to learn about industry presentation styles and to identify the topic of your group project.
- Students, please form a 3-person project team on or before September 13, 2022, and send your list of members to Dr. Lai-Man Po at eelmpo@cityu.edu.hk .
- On the other hand, students are strongly recommended to try Google Colab to practice programming skills using Python, Numpy, Mathpoltlib, Scipy, etc. There is a good Colab Python tutorial here:
  - https://colab.research.google.com/github/cs231n/cs231n.github.io/blob/master/python-colab.ipynb

Python Tutorial with Google Colab

https://colab.research.google.com/github/cs231n/cs231n.github.io/blob/master/python-colab.ipynb



CS231n Python Tutorial With Google Colab

This tutorial was originally written by Justin Johnson for cs231n. It was adapted as a Jupyter notebook for cs228 by Volodymyr Kuleshov and Isaac Caswell.

This version has been adapted for Colab by Kevin Zakka for the Spring 2020 edition of cs231n. It runs Python3 by default.

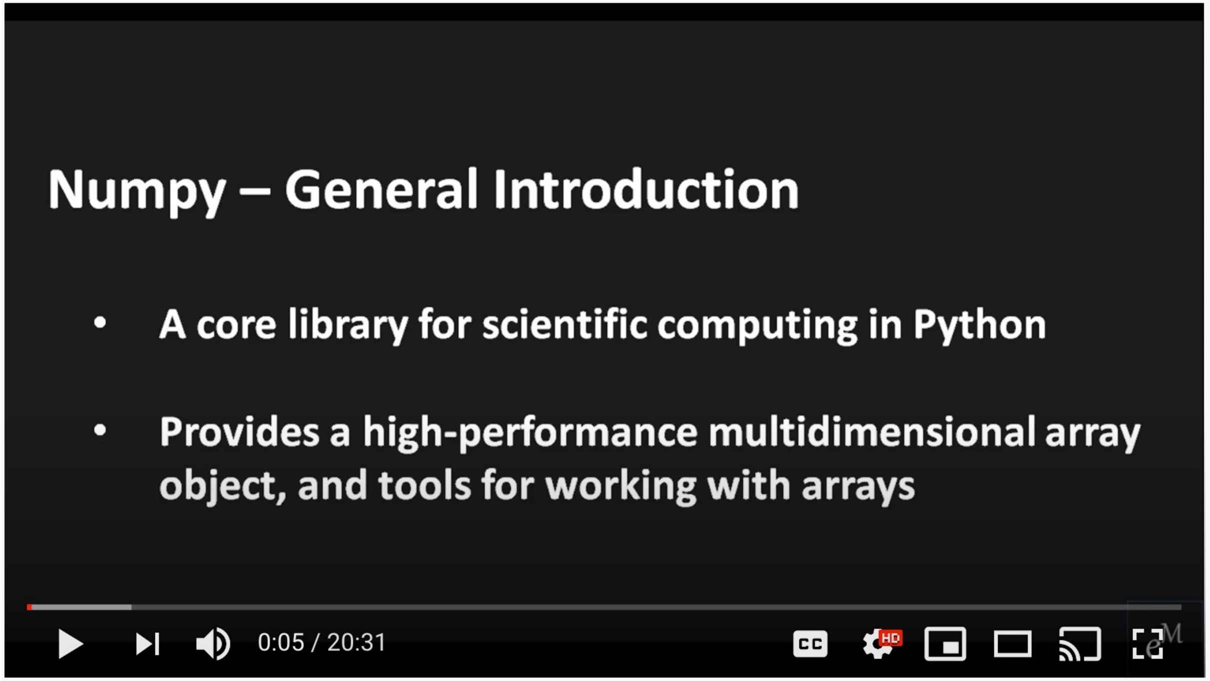
Introduction

Python is a great general-purpose programming language on its own, but with the help of a few popular libraries (numpy, scipy, matplotlib) it becomes a powerful environment for scientific computing.

We expect that many of you will have some experience with Python and numpy; for the rest of you, this section will serve as a quick crash course both on the Python programming language and on the use of Python for scientific computing.

Some of you may have previous knowledge in Matlab, in which case we also recommend the numpy for Matlab users page (https://docs.scipy.org/doc/numpy-dev/user/numpy-for-matlab-users.html).

Numpy Tutorial in 20 Minutes



Python Tutorial: Learn Numpy - Array Indexing & Creation, Basic & Advanced Operations in 20 Minutes

https://www.youtube.com/watch?v=G14STCiT2Jw

Content

Discrete-Time Signals (Sequences)

- Causality and Duration of Sequences
- Deterministic and Random Sequences
- Basic Sequences
  - Unit Sample, Unit Step, Ramp, Power, exponential and Sinusoid
- Composite Sequences
- Unit Sample based Composite Sequence Expression
- Two-Dimension Digital Signals (Optional)

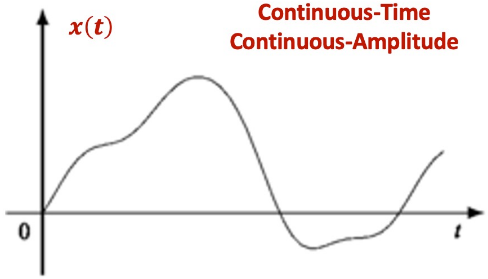
Discrete-Time Systems

- Definition of Discrete-Time (DT) Systems
- Classification of DT Systems
  - Memory, Time-Invariant, Linear, Causal and Stable
- Linear Time-Invariant (LTI) System
  - Impulse Response
  - Convolution
- Block Diagram Representation
- Difference Equations
- Finite Impulse Response (FIR) System
- Infinite Impulse Response (IIR) System
- Stability of LTI System

Discrete-Time Signals

Continuous-Time Signals (Analog Signals)

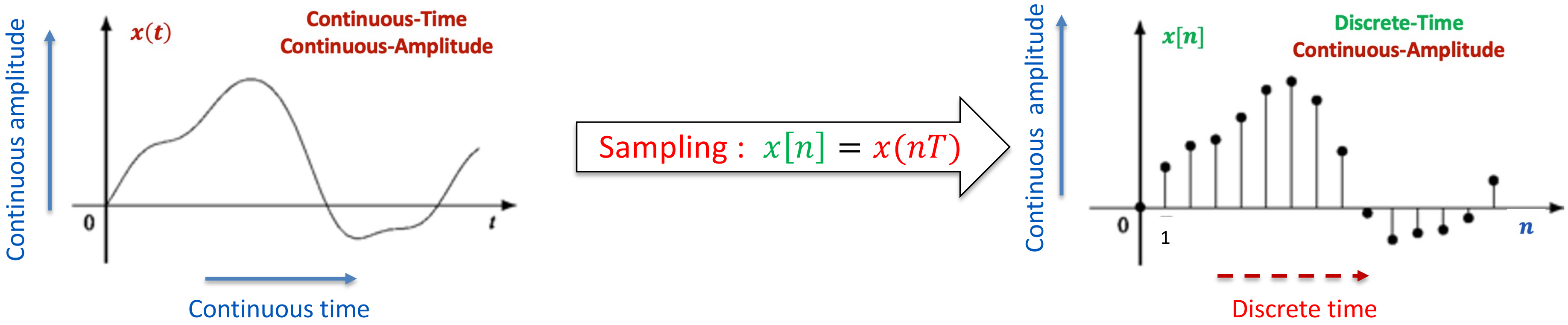
- Continuous-Time (CT) signal $x(t)$ is a signal that exists at every instant of time
  - A CT signal is often referred to as analog signal
  - The independent variable is a continuous variable
  - Continuous signal can assume any value over a continuous range of numbers



- Most of the signals in the physical world are CT signals.
- Examples: voltage & current, pressure, temperature, velocity, etc.

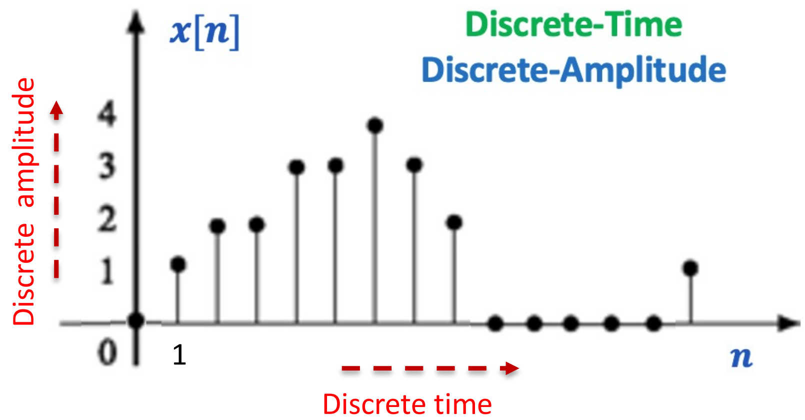
Discrete-Time Signals

- A signal defined only for discrete values of time is called a discrete-time (DT) signal or simply a sequence
- Discrete-time signal can be obtained by taking samples of an analog signal at discrete instants of time : $x[n] = x(nT)$
- The values of each sample $x[n]$ is continuous



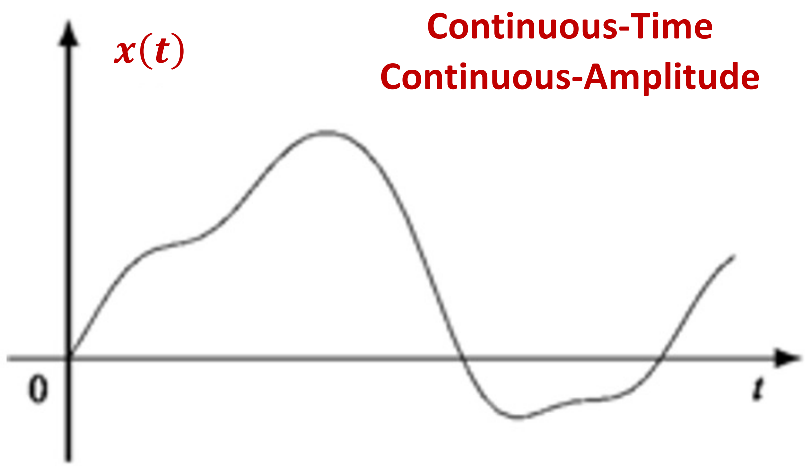
Digital Signals

- Digital signal is a discrete-time signal whose values are quantized and represented by digits
  - Discrete-Time
  - Discrete-Amplitude

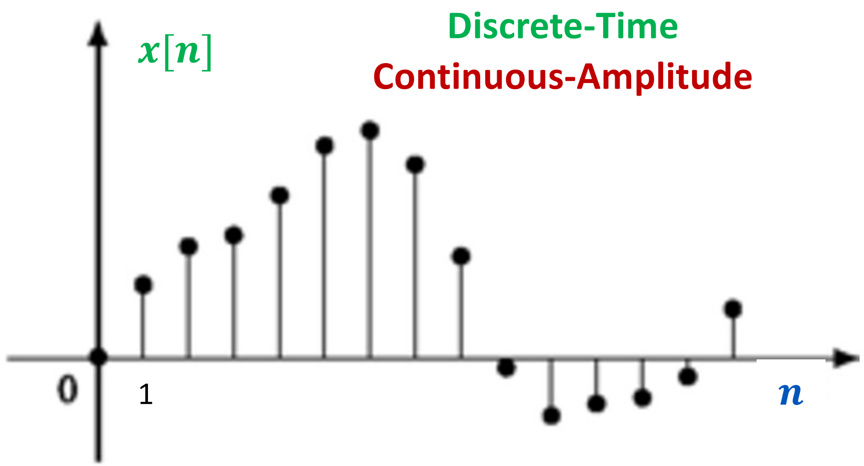


- The digital signal is the sampled and quantized (rounded) representation of the analog signal. A digital signal consists of a sequence of samples, which in this case are integers: 0, 1, 2, 2, 3, 3, 4, 3, ...

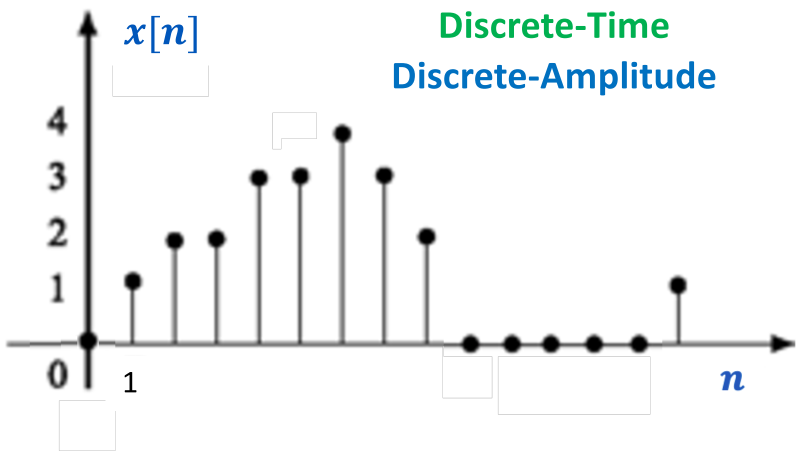
Continuous-Time Signal



Discrete-Time Signal

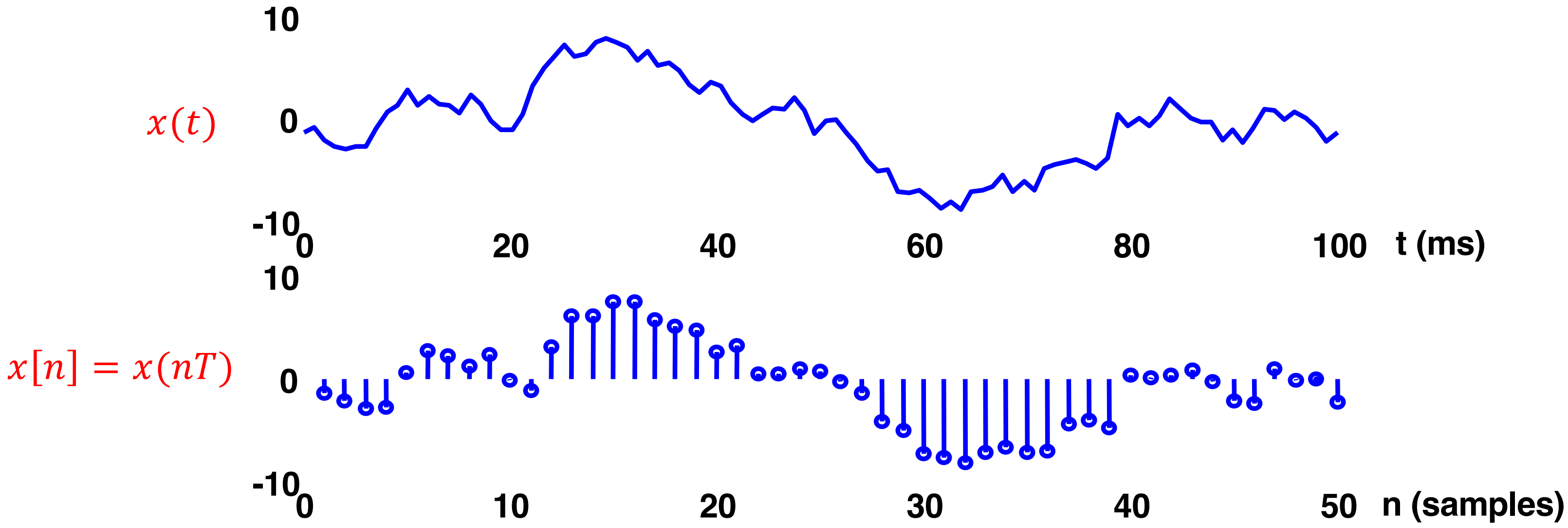


Digital Signal



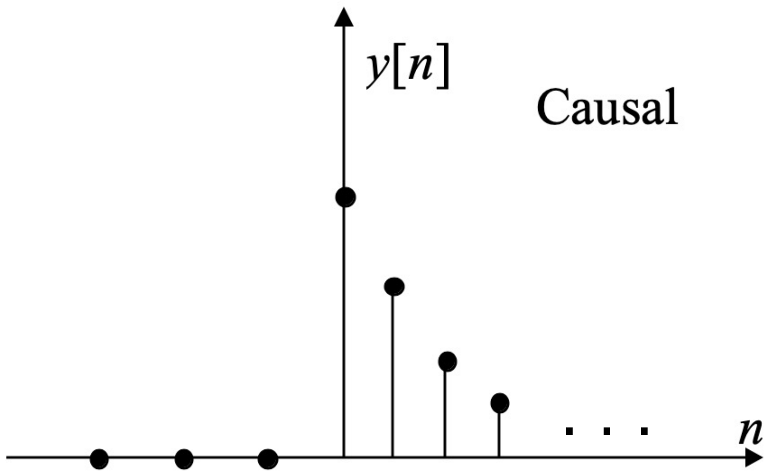
Discrete-Time Signals (Sequences)

- Discrete-Time (DT) signals are represented by sequence of numbers
  - The nth number in the sequence is represented with $x[n]$
  - Often times sequences are obtained by sampling of Continuous-Time signals
  - In this case $x[n]$ is value of the analog signal $x(t)$ at $x(nT)$ where T is the sampling period



Causal Sequences

- Discrete-time signal or sequence is called **causal** if it has zero values for $n < 0$.
- Here is an example of a causal sequence



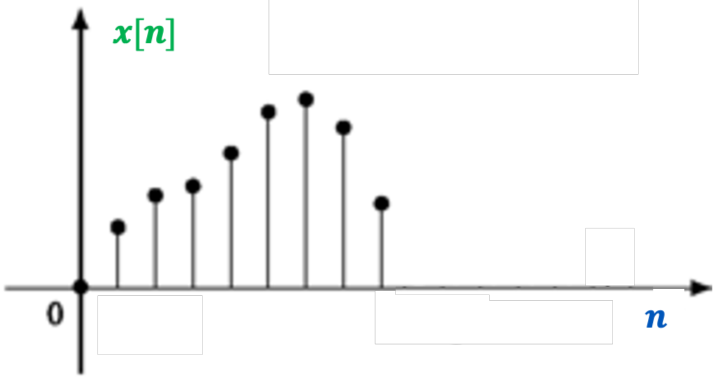
$$y[n] = \left(\tfrac{2}{3}\right)^n u[n]$$

Causality and Duration of Sequences

- A sequence that is nonzero only over a finite interval of indices is called a finite-duration (finite-length) sequence
- A sequence whose samples are zero-valued for negative indices is causal
- Non-causal sequence can have nonzero samples for negative indices

This is an example of causal finite-duration sequence

- $x[n] = 0$ for $n < 0$ (causal)
- $x[n] \neq 0$ only for $0 \leq n \leq 8$



Basic Discrete-Time Signals (Sequences)

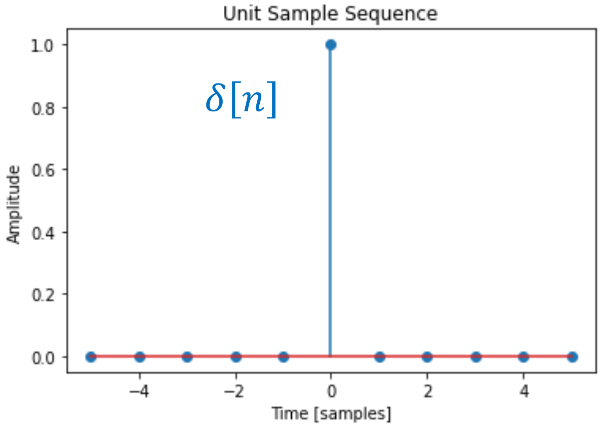
The most basic discrete-time signals (sequences) are

- Unit Impulse Sequence
- Unit Step Sequence
- Unit Ramp Sequence
- Power Sequence
- Exponential Sequence
- Sinusoidal Sequence

Unit Impulse Sequence

- The **unit impulse (sample) sequence** is defined as the sequence with values

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



- The unit impulse sequence $\delta[n]$ has an amplitude of zero at all samples except $n = 0$, where it has the value 1.
- Every discrete-time signal can be written as a sum of unit impulse sequences, using the amplitude at each sample.

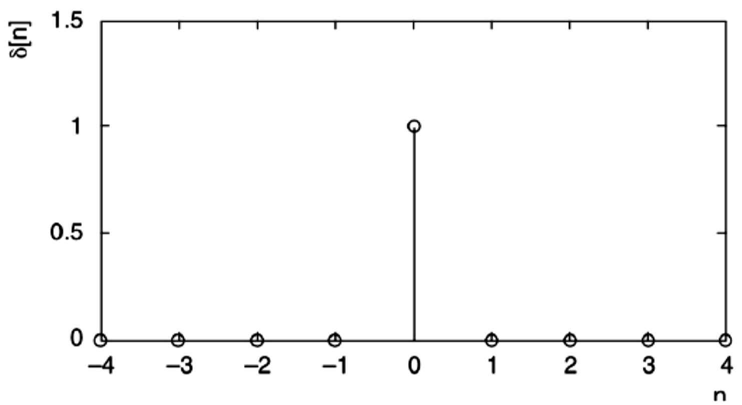
Examples of Unit Impulse Sequence

Determine the following:

(a) $\delta[0]$

(b) $\delta[3]$

(c) $\delta[-2]$



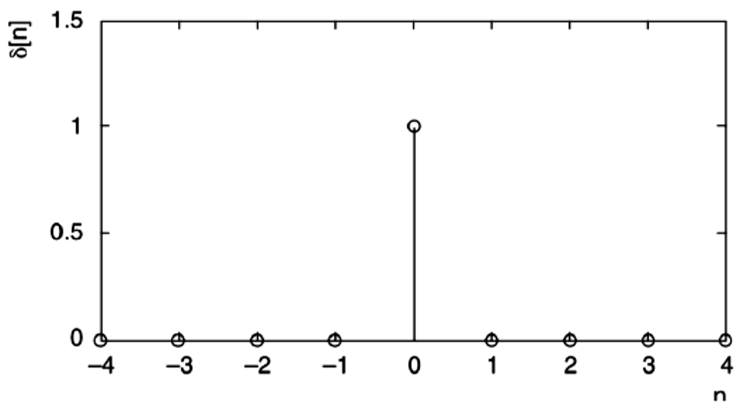
Examples of Unit Impulse Sequence

Determine the following:

(a) $\delta[0]$

(b) $\delta[3]$

(c) $\delta[-2]$



<u>Answers</u>

(a) $\delta[0] = 1$

(b) $\delta[3] = 0$

(c) $\delta[-2] = 0$

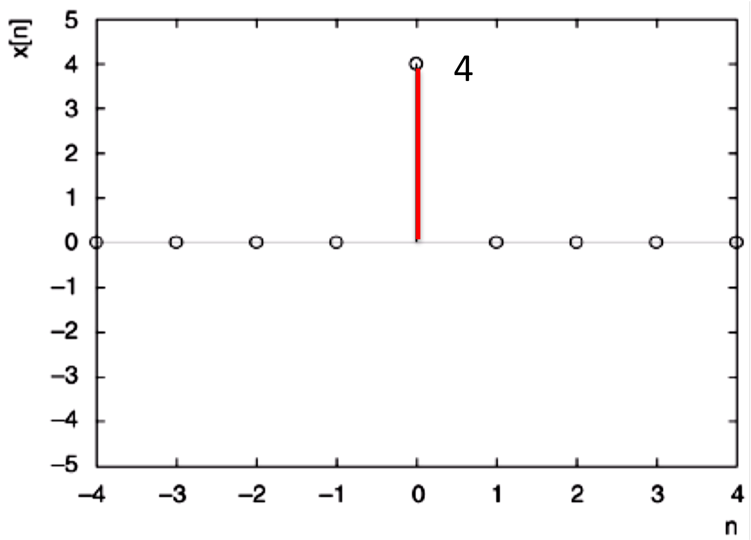
Scaled Unit Impulse Function

1. Draw the sequence of $x[n] = 4\delta[n]$

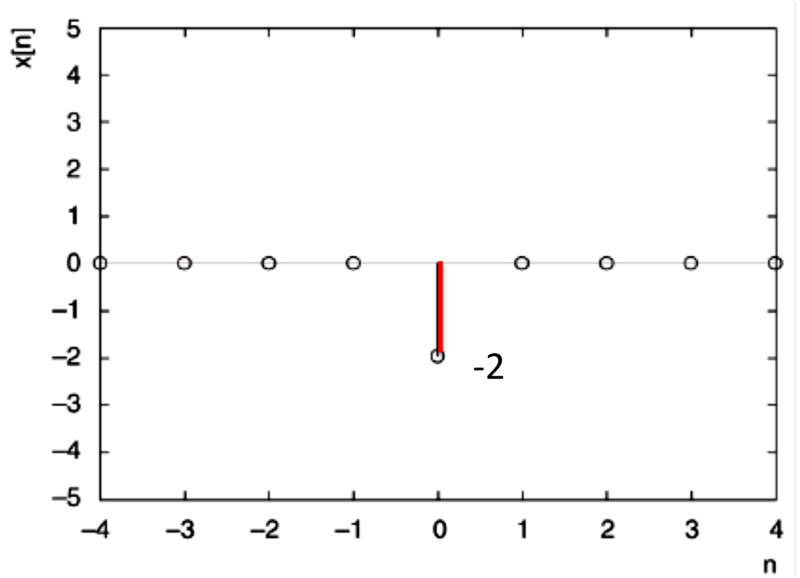
2. Draw the sequence of $x[n] = -2\delta[n]$

Scaled Unit Impulse Function

1. Draw the sequence of $x[n] = 4\delta[n]$



2. Draw the sequence of $x[n] = -2\delta[n]$



Shifted Unit Impulse Sequences

1. Draw the sequence of $x[n] = \delta[n-2]$

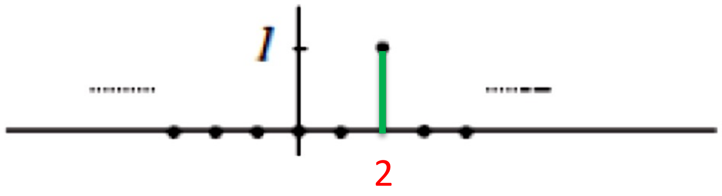
2. Draw the sequence of $x[n] = \delta[n+2]$

Shifted Unit Impulse Sequences

1. Draw the sequence of $x[n] = \delta[n-2]$

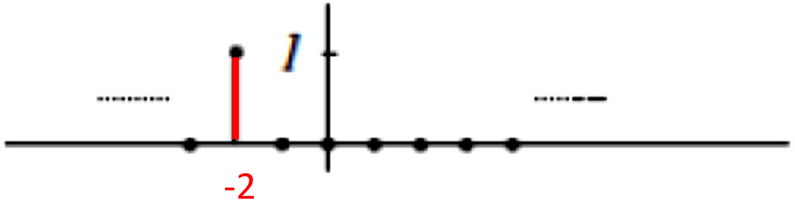
2. Draw the sequence of $x[n] = \delta[n+2]$

$\delta[n-2]$



Shift to right

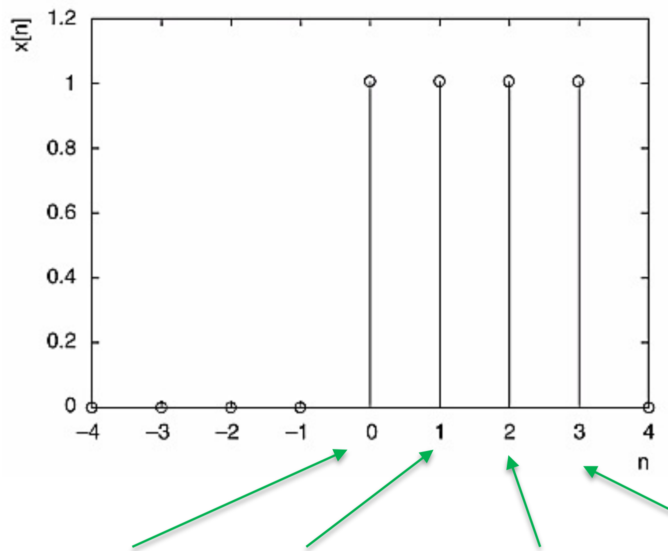
$\delta[n+2]$



Shift to left

Finite-Duration Sequence in Terms of Unit Impulses (1)

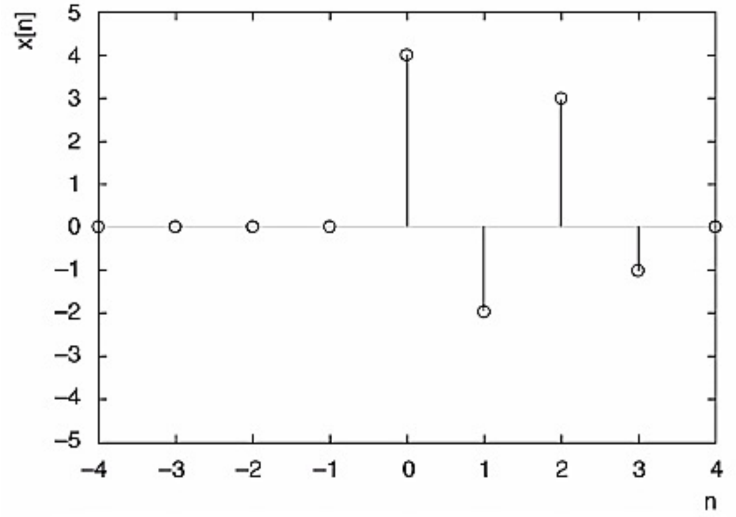
- Write a sequence $x[n]$ in terms of $\delta[n]$ to describe the sequence as shown below:



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

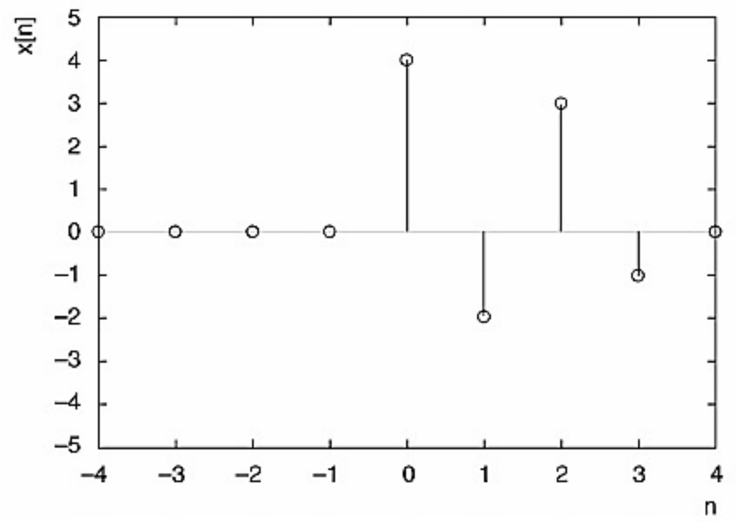
Finite-Duration Sequence in Terms of Unit Impulses (2)

- Write a sequence x[n] in terms of $\delta[n]$ to describe the sequence as shown below:



Finite-Duration Sequence in Terms of Unit Impulses (2)

- Write a sequence x[n] in terms of $\delta[n]$ to describe the sequence as shown below:



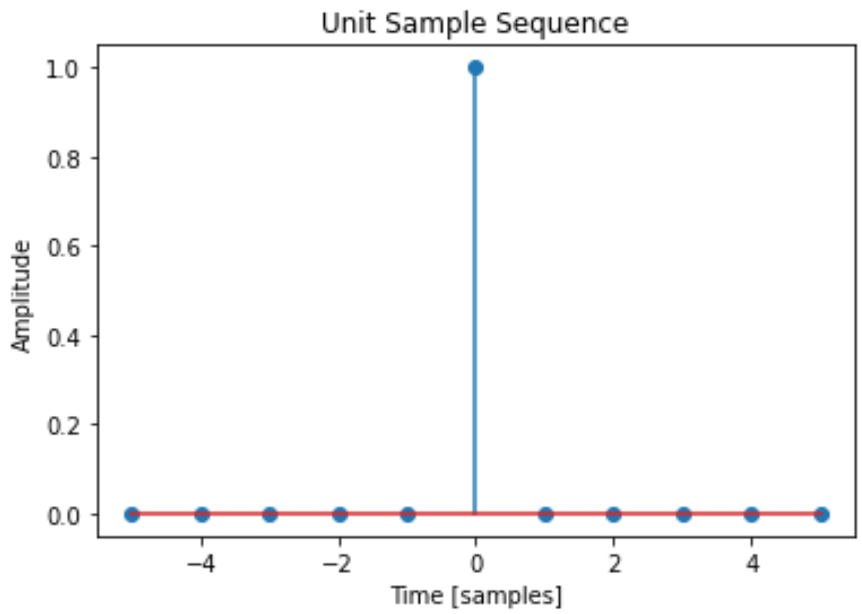
$$x[n] = 4\delta[n] - 2\delta[n-1] + 3\delta[n-2] - \delta[n-3]$$

Unit Impulse Sequence : Python Code

Python Code

```
import matplotlib.pyplot as plt
import numpy as np

from scipy import signal

imp = signal.unit_impulse(11, 'mid')

plt.stem(np.arange(-5, 6), imp)
plt.xlabel('Time [samples]')
plt.ylabel('Amplitude')
plt.title('Unit Sample Sequence')

plt.show()
```

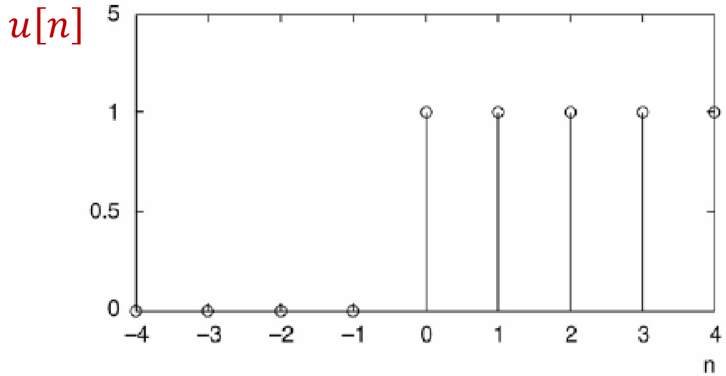
$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$



Unit Step Sequence

- The unit step sequence is defined as the sequence with values

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



- The unit step function $u[n]$ has an amplitude of zero for $n < 0$ and an amplitude of one for all other samples.
- The signal $u[-n]$ has the value one up to and including $n = 0$, and the value zero thereafter.

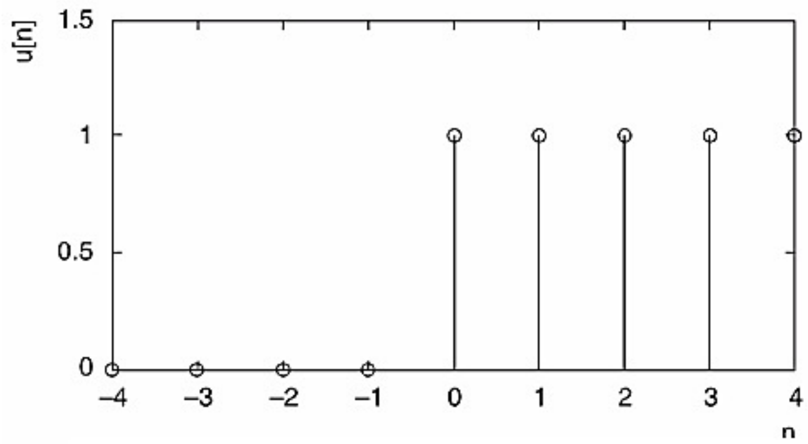
Examples of Unit Step Sequence

Determine the following:

a) **u[-1]**

b) **u[0]**

c) **u[1]**



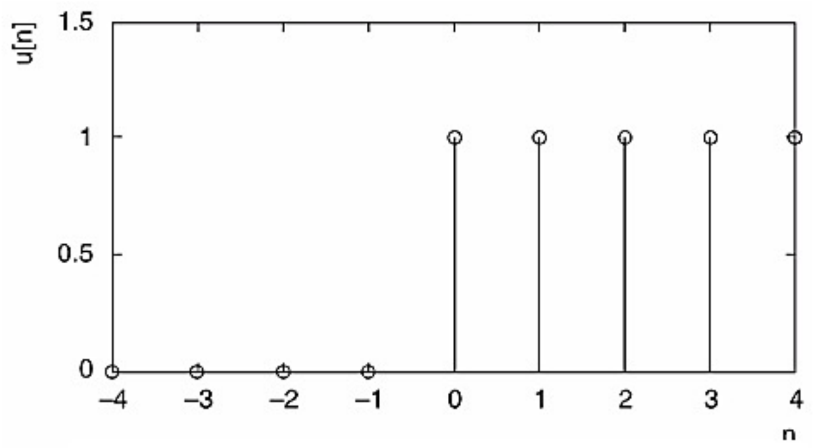
Examples of Unit Step Sequence

Determine the following:

a) u[-1]

b) u[0]

c) u[1]



Answers

a) u[-1] = 0

b) u[0] = 1

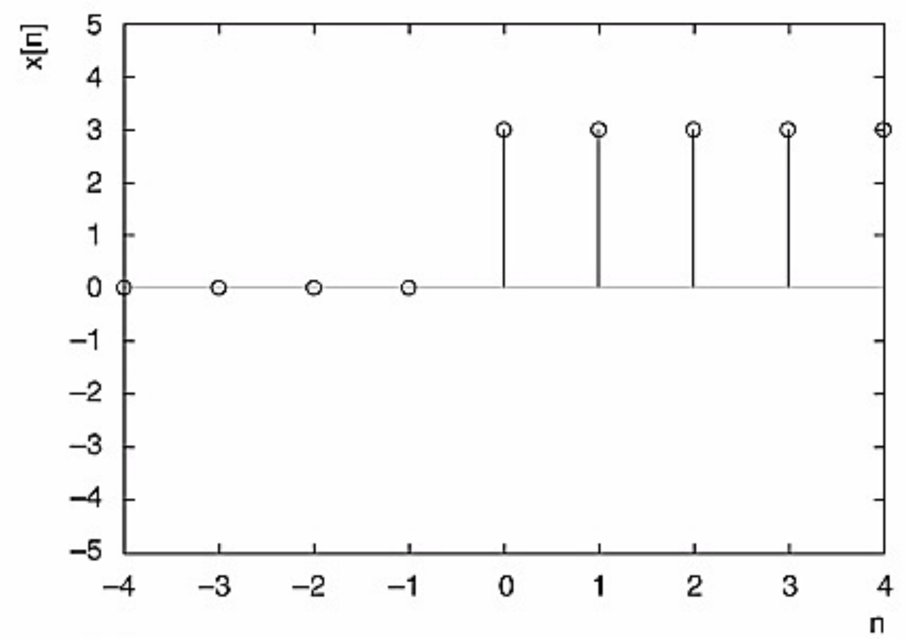
c) u[1] = 1

Scaled Unit Step Sequence

1. Draw the signal of $x[n] = 3u[n]$

Scaled Unit Step Sequence

1. Draw the signal of x[n] = 3u[n]

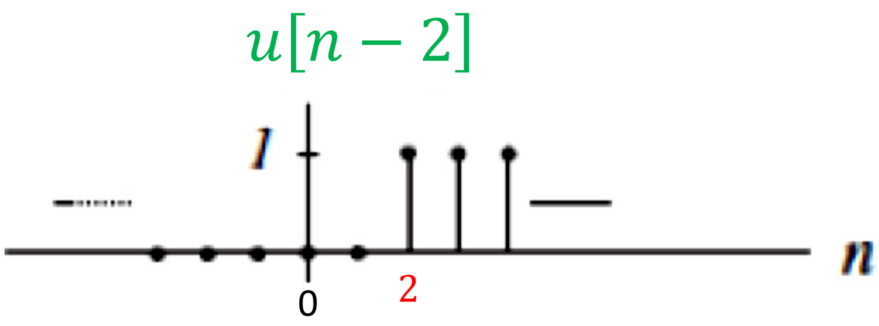


Shifted Unit Step Sequence

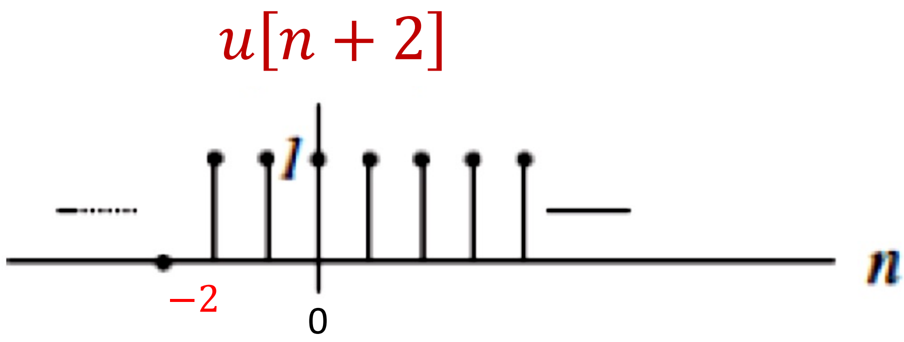
1. Draw the sequence of $x[n] = u[n-2]$ **2. Draw the sequence of** $x[n] = u[n+2]$

Shifted Unit Step Sequence

1. Draw the sequence of $x[n] = u[n-2]$ **2. Draw the sequence of** $x[n] = u[n+2]$



Shift to right



Shift to left

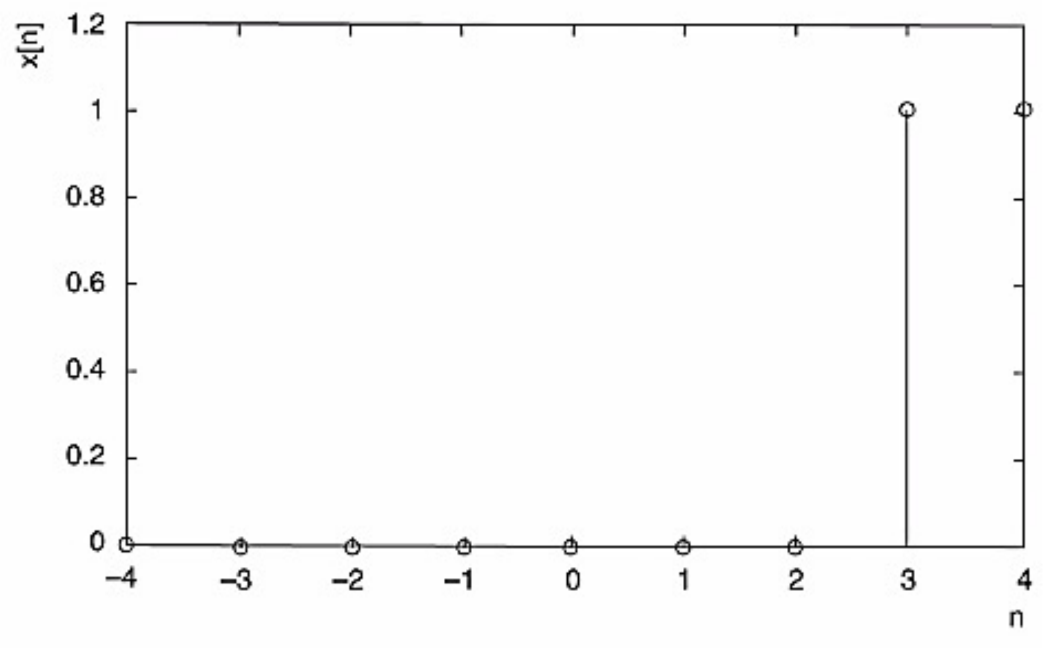
They are both **right-sided sequences** with $x[n] = 0$ for $n < 0$ or $n < -2$

Shifted Unit Step Sequence Example

Draw the sequence $x[n] = u[n-3]$

Shifted Unit Step Sequence Example

Draw the sequence $x[n] = u[n-3]$

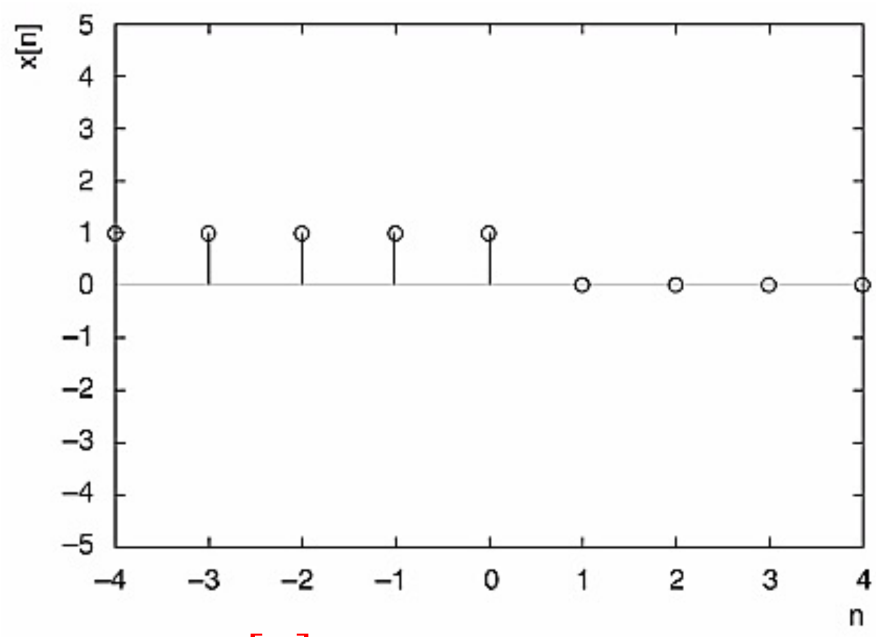


A Mirrored Unit Step Sequence

Draw a mirrored sequence $x[n] = u[-n]$

A Mirrored Unit Step Sequence

Draw a mirrored sequence $x[n] = u[-n]$



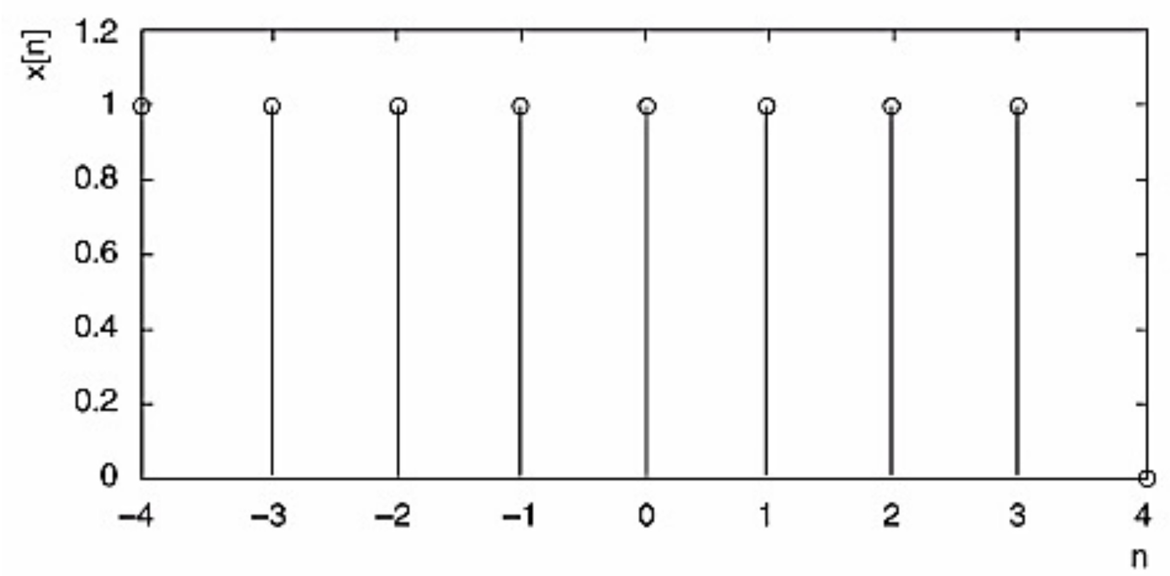
This is a **left-sided sequence** with $x[n] = 0$ for $n > 0$

A Right-Shifted Left-Sided Unit Step Sequence

Draw the sequence $x[n] = u[3-n]$

A Right-Shifted Left-Sided Unit Step Sequence

Draw the sequence $x[n] = u[3-n]$



Unit Step Sequence Example (1)

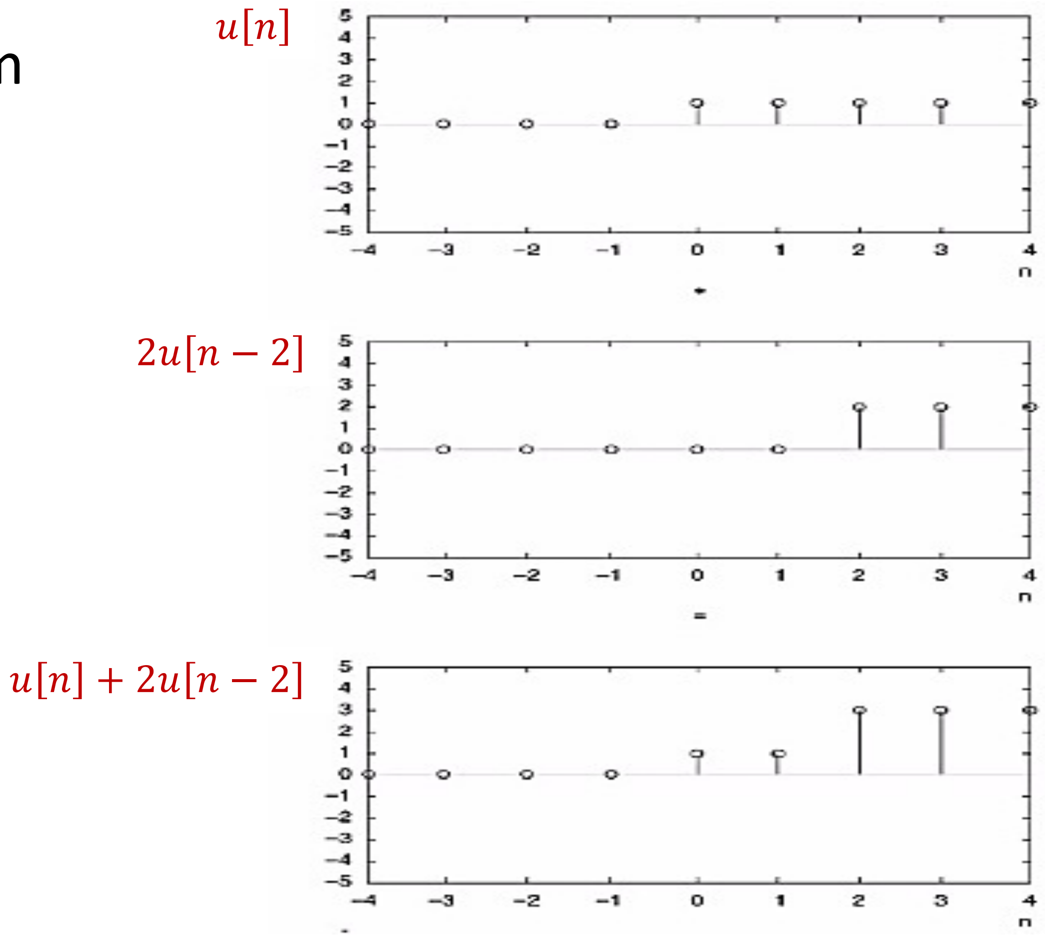
Draw the sequence as sum of two step sequences:

$$x[n] = u[n] + 2u[n-2]$$

Unit Step Sequence Example (1)

Draw the sequence as sum of two step sequences:

$$x[n] = u[n] + 2u[n-2]$$



Unit Step Sequence Example (2)

Draw the sequence as sum of two step sequences:

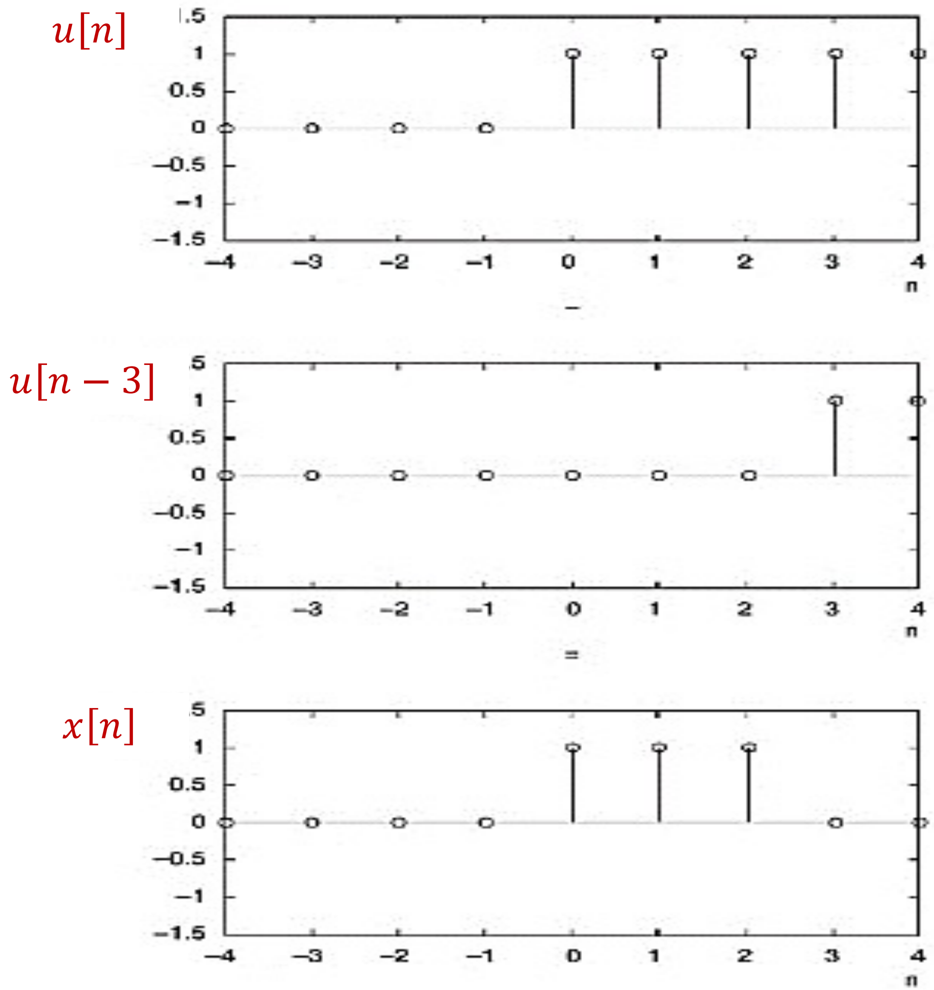
$$x[n] = u[n] - u[n-3]$$

Unit Step Sequence Example (2)

Draw the sequence as sum of two step sequences:

$$x[n] = u[n] - u[n-3]$$

This is a finite duration sequence



Unit Step Sequence Example (3)

A discrete-time signal is described as $x[n] = 4(u[n] - u[n-1])$.

Write the function that describes $x[n-3]$.

Unit Step Sequence Example (3)

A discrete-time signal is described as $x[n] = 4(u[n] - u[n-1])$.

Write the function that describes $x[n-3]$.

Answer

Simplify $x[n] = 4(u[n] - u[n-1]) = 4\delta[n]$

Substituting n = (n – 3) gives

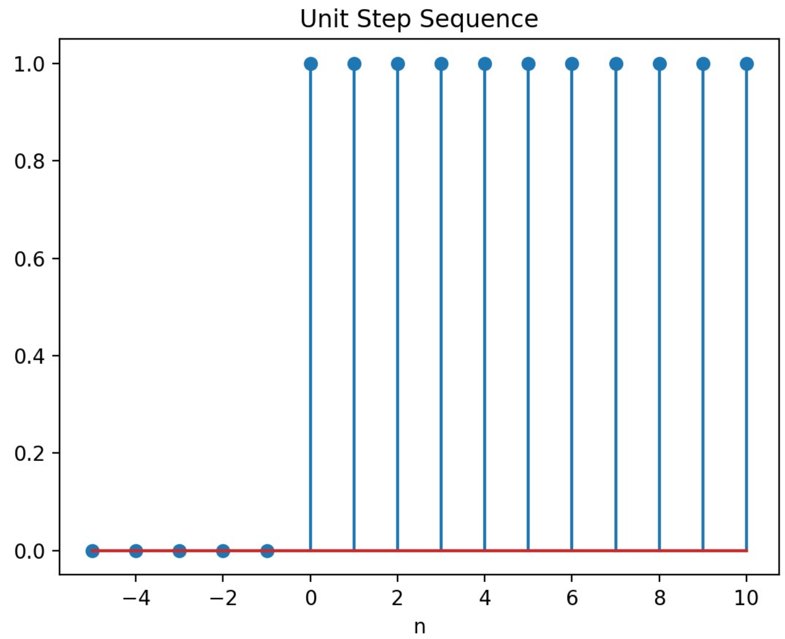
$$x[n-3] = 4\delta[n-3]$$

Unit Step Sequence : Python Code

Python Code

```
import matplotlib.pyplot as plt
import numpy as np

n = np.arange(start=-5, stop=11, step=1)

x = np.zeros(16)
x[5:16] = 1

plt.stem(n, x, bottom=0, markerfmt='o')

plt.xlabel('n')
plt.title('Unit Step Sequence')

plt.show()
```

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



Connection between Unit Sample and Step Sequences

- The unit step sequence can be expressed as a sum of impulse sequences:

$$u[n] = \delta[n] + \delta[n-1] + \delta[n-2] + \ldots$$

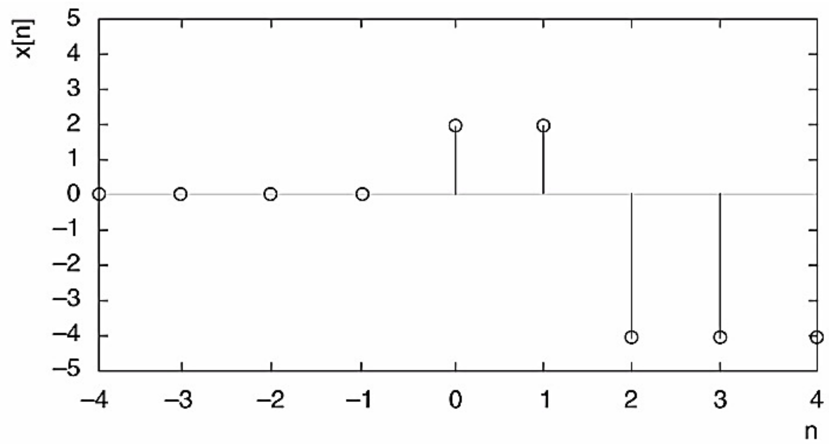
$$u[n] = \sum_{m=0}^{\infty} \delta[n-m]$$

- Similarly, unit sample sequence can also be expressed as a difference of two step sequences :

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

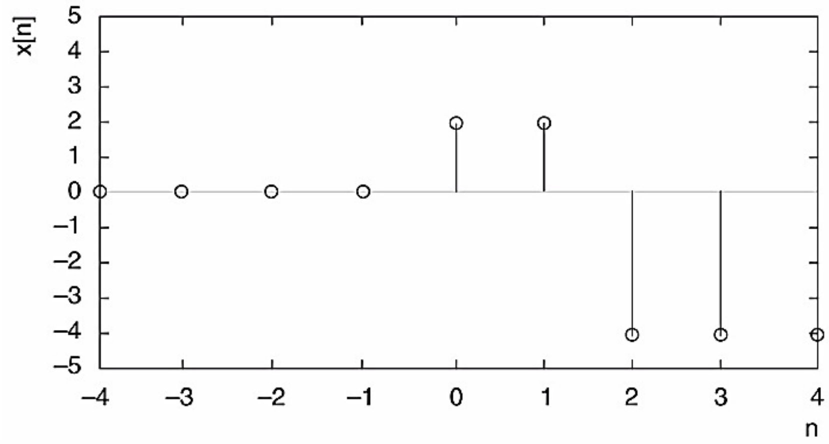
Connection between Impulse and Step Sequences

- **Write a sequence to describe the signal in terms of unit sample and unit step sequences of the figure.**



Connection between Impulse and Step Sequences

- **Write a sequence to describe the signal in terms of unit sample and unit step sequences of the figure.**



Unit Impulse Sequence

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

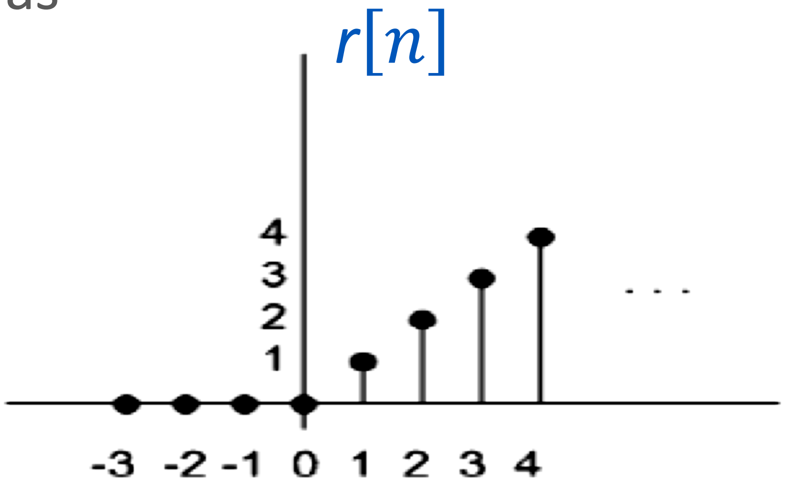
Unit Step Sequence

$x[n] = 2u[n] - 6u[n-2]$

Unit Ramp Sequence

- The unit-ramp sequence is defined as

$$r[n] = \begin{cases} n, & n \geq 0 \\ 0, & n < 0 \end{cases}$$



Power Sequences

- Power sequences take the form:

$$x[n] = A\,\alpha^{\beta n}$$

- In the special case where $\alpha = e$ (Euler number), such sequences are called exponential sequences.
- When β is positive, the values of the sequence grows.
- When β is negative the values of the sequence decays.
- When α is negative, the values of the sequence alternate positive and negative.
- The value of A is determined the magnitude/amplitude/value of the sequence when n = 0

Power Sequence Example 1

- **Draw a signal**

  $x[n] = 0.8(0.75)^n$

The magnitude of the sequence is decaying due to α is positive but less than one.

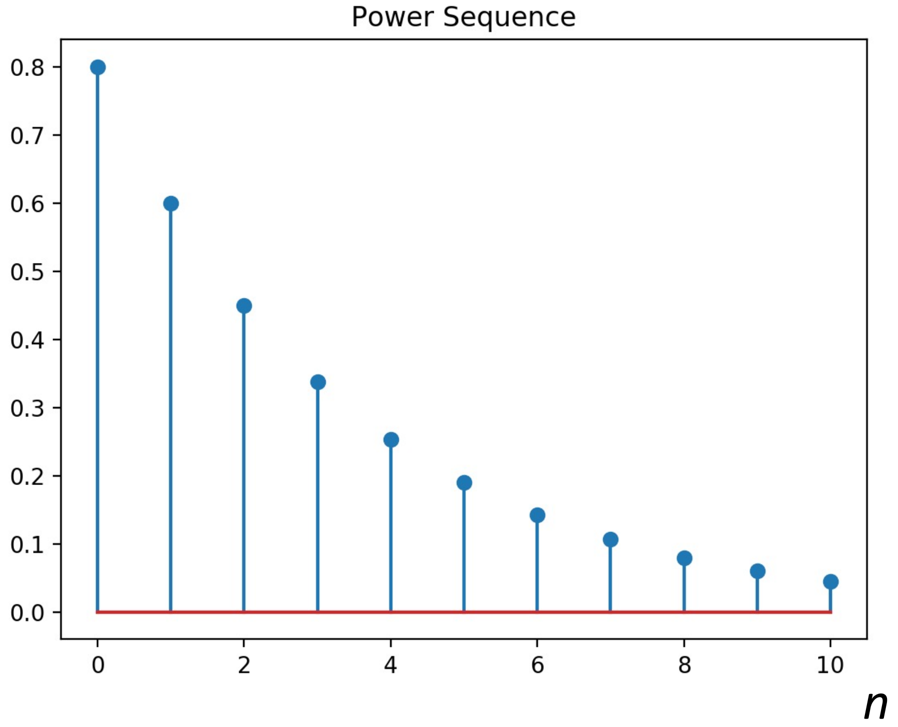
n

Power Sequence Example 1

- **Draw a signal**

  $x[n] = 0.8(0.75)^n$

The magnitude of the sequence is decaying due to α is positive but less than one.



Power Function Example 2

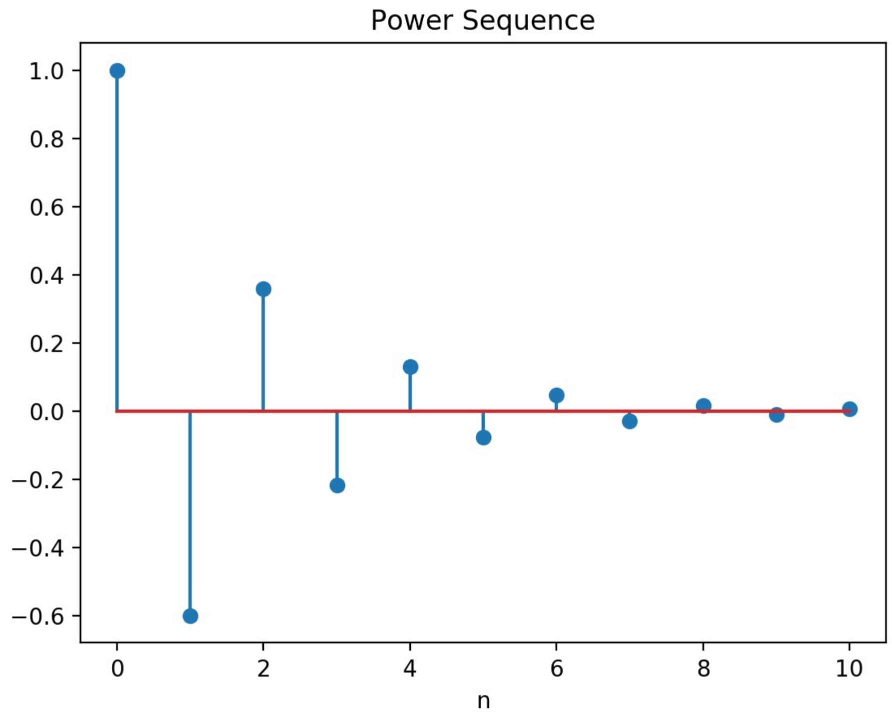
- **Draw a sequence of $x[n] = (-0.6)^n$**

The magnitude of the signal is alternating between positive and negative due to α is negative.

Power Function Example 2

- **Draw a sequence of $x[n] = (-0.6)^n$**

The magnitude of the signal is alternating between positive and negative due to α is negative.

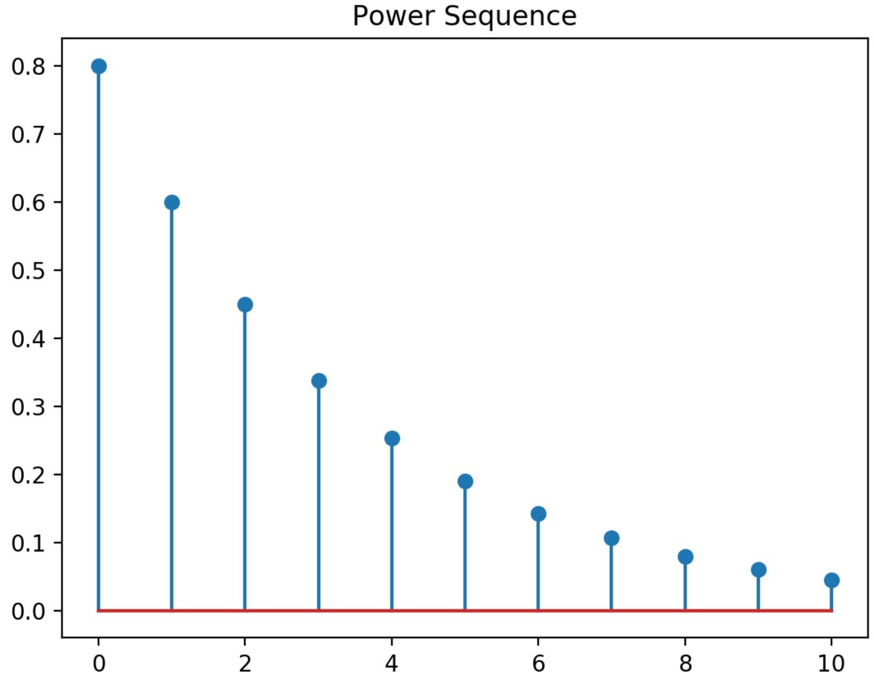


Power Sequence : Python Code

Python Code

```
import matplotlib.pyplot as plt
import numpy as np

n = np.arange(start=0, stop=11, step=1)

x = 0.8 * np.power(0.75,n)

plt.stem(n, x, bottom=0, markerfmt='o')

plt.xlabel('n')
plt.title('Power Sequence')

plt.show()
```

$$x[n] = 0.8\,(0.75^n)$$



Exponential Sequences

- Exponential functions are special cases of Power function, which take the form:

$$x[n] = A\, e^{\beta n}$$

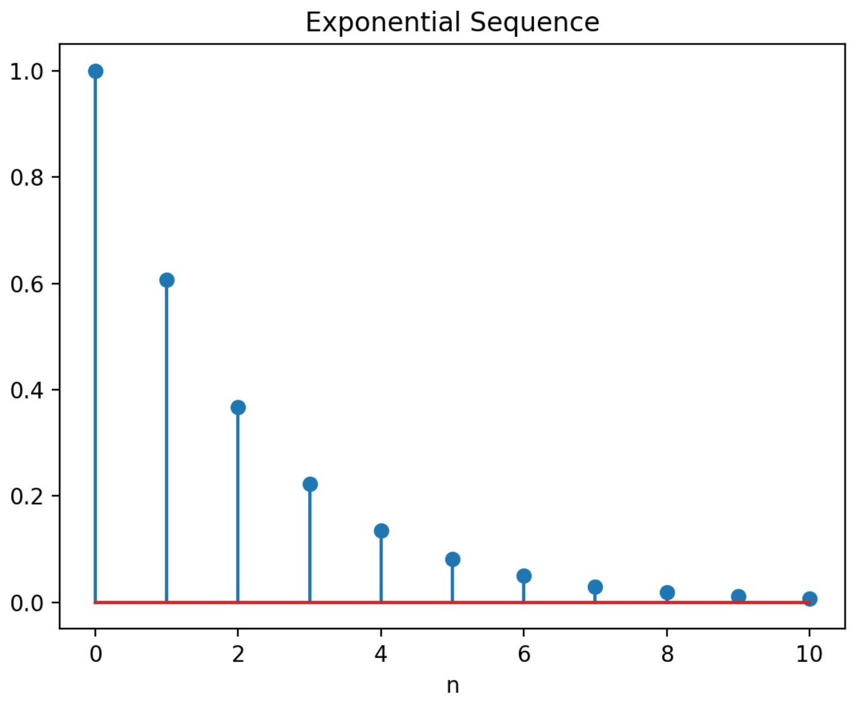
- Where e = 2.71828 (α is fixed and positive)
- When β is positive, the function grows.
- When β is negative the function decays.
- The value of A is determined the magnitude/amplitude/value of the function when n = 0

Exponential Sequence Example 1

- **Draw an Exponential Sequence**

$$x[n] = e^{-0.5n}$$

The magnitude of the signal is decaying due to β is negative and e > 1

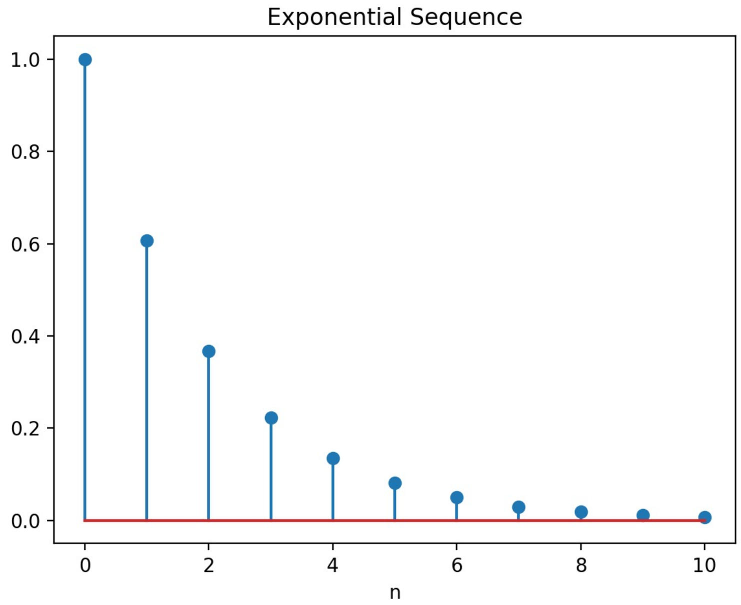


Exponential Sequence : Python Code

Python Code

```
import matplotlib.pyplot as plt
import numpy as np

n = np.arange(start=0, stop=11, step=1)

x = np.exp(-0.5*n)

plt.stem(n, x, bottom=0, markerfmt='o')

plt.xlabel('n')
plt.title('Exponential Sequence')

plt.show()
```

$$x[n] = e^{-0.5n}$$



Complex Exponential Sequences

- A sequence of the form

$$x[n] = A\, e^{j\beta n}$$

  is called a complex exponential sequence.
- For all n, samples of this sequence lie in the complex plane on a circle with radius A.
- By Euler's Formula, a complex exponential may be expressed as a rectangular-form complex number

$$e^{j\beta n} = \cos(\beta n) + j\sin(\beta n)$$

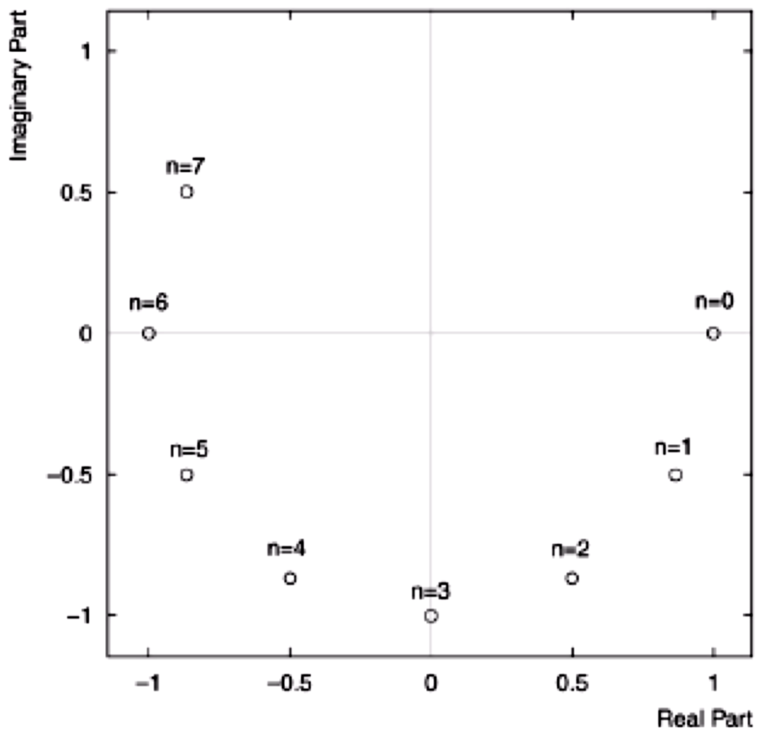
- The general form of a **complex exponential sequence** has the forms

$$x[n] = |A|e^{j(\omega_0 n+\phi)} = |A|\cos(\omega_0 n + \phi) + j|A|\sin(\omega_0 n + \phi)$$

Complex Exponential Sequence

Plot the first eight samples of a complex exponential sequence:

$$x[n] = e^{-j\pi n/6}$$



Using Euler's Formula

$$x[n] = \cos\left(\frac{\pi n}{6}\right) + j\sin\left(\frac{\pi n}{6}\right)$$

Sinusoidal Sequence

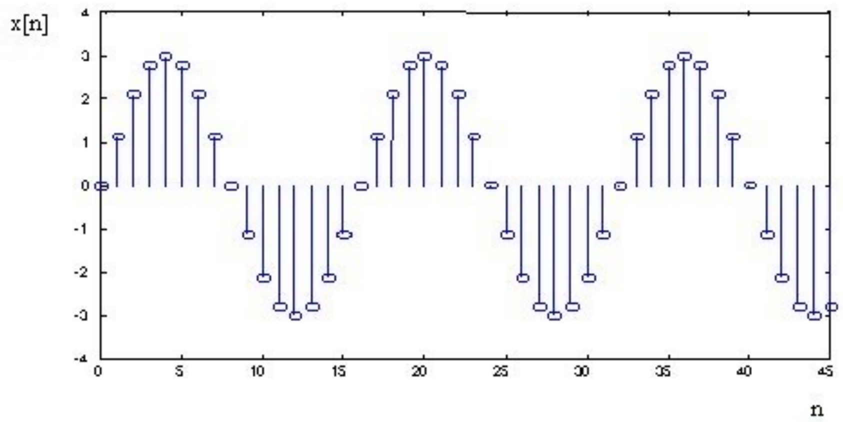
- The sinusoidal functions take the form

$$x[n] = A\cos(\omega n + \phi) \quad \text{or} \quad x[n] = A\sin(\omega n + \phi)$$

for all n with real A

- Where ω is a discrete-time angular frequency in radians/sample and ϕ is a phase shift.

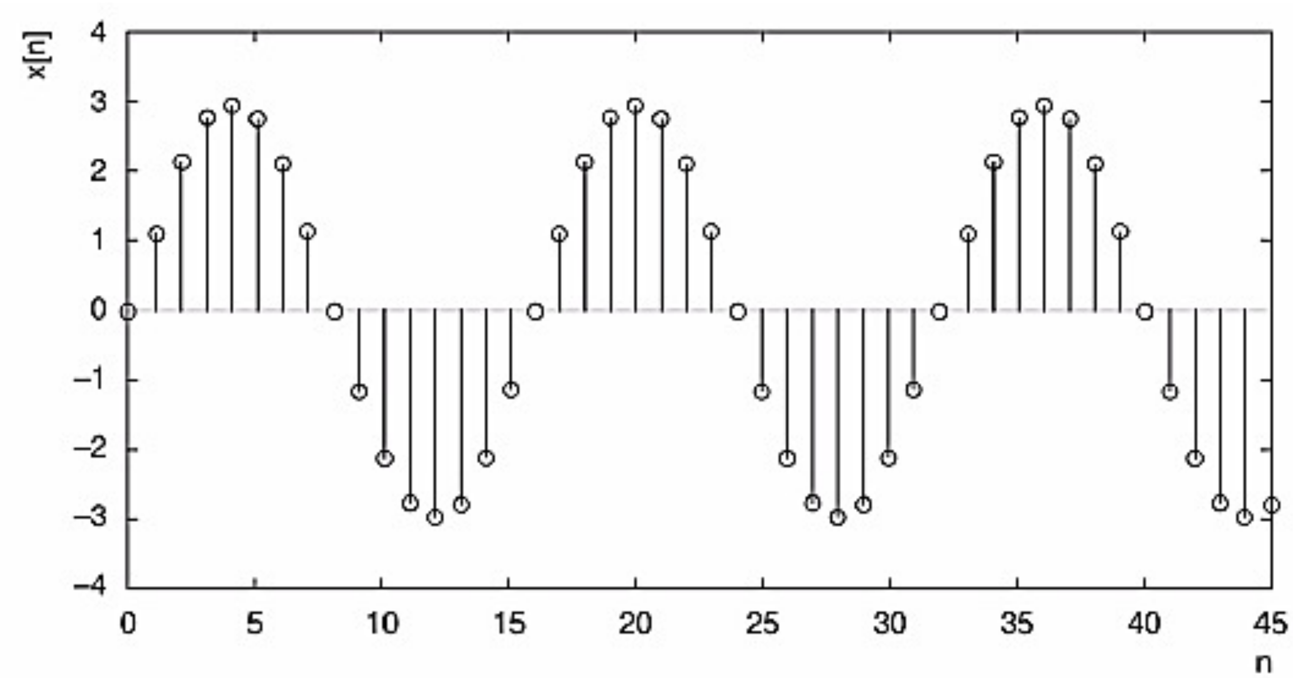
$$\omega = \Omega T = \frac{2\pi f}{F_s}$$



- Ω is the continuous-time frequency in radians/second
- f is continuous-time frequency in Hz and $\Omega = 2\pi f$
- T is the sampling period in seconds
- F_s is the sampling frequency and $F_s = \frac{1}{T}$

Sinusoidal Sequence Example 1

Plot of x[n] = 3sin(nπ/8)

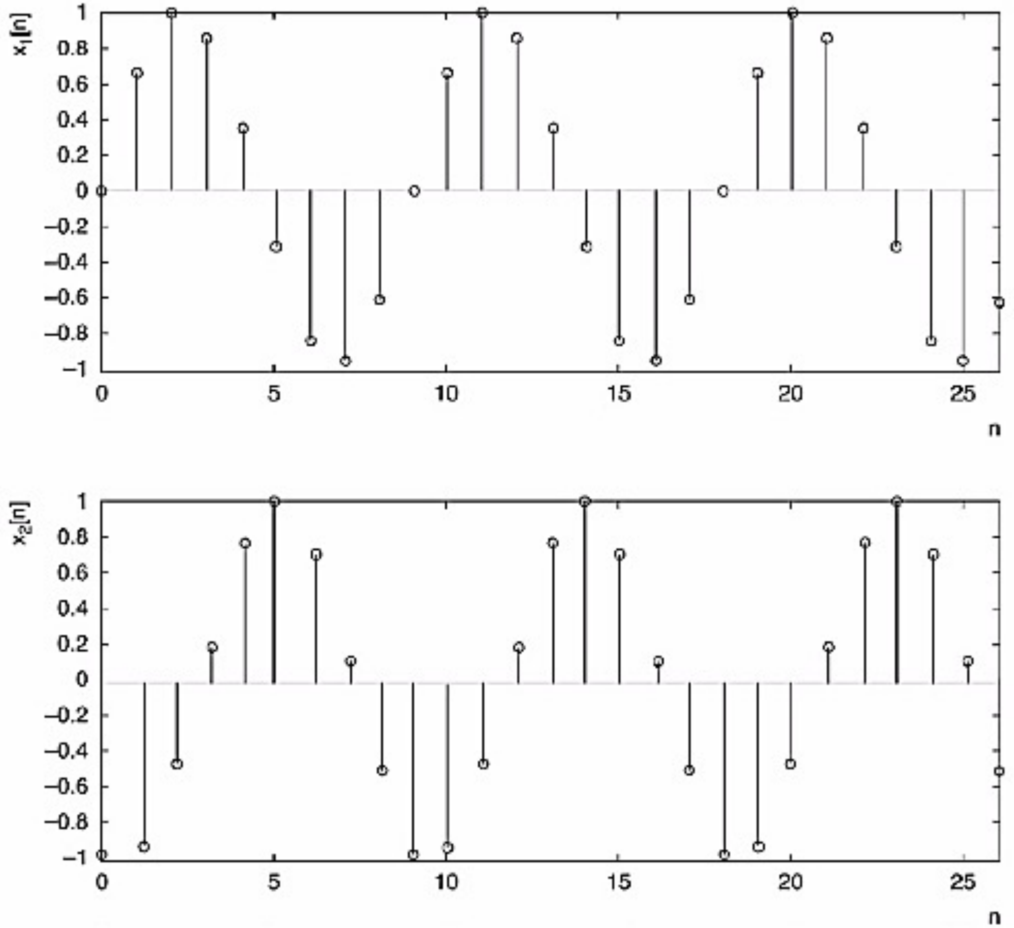


Phase Shifting a Digital Sinusoid

Plot the following sequences

$x_1[n] = \sin(n2\pi/9)$

$x_2[n] = \sin(n2\pi/9 - 3\pi/5)$

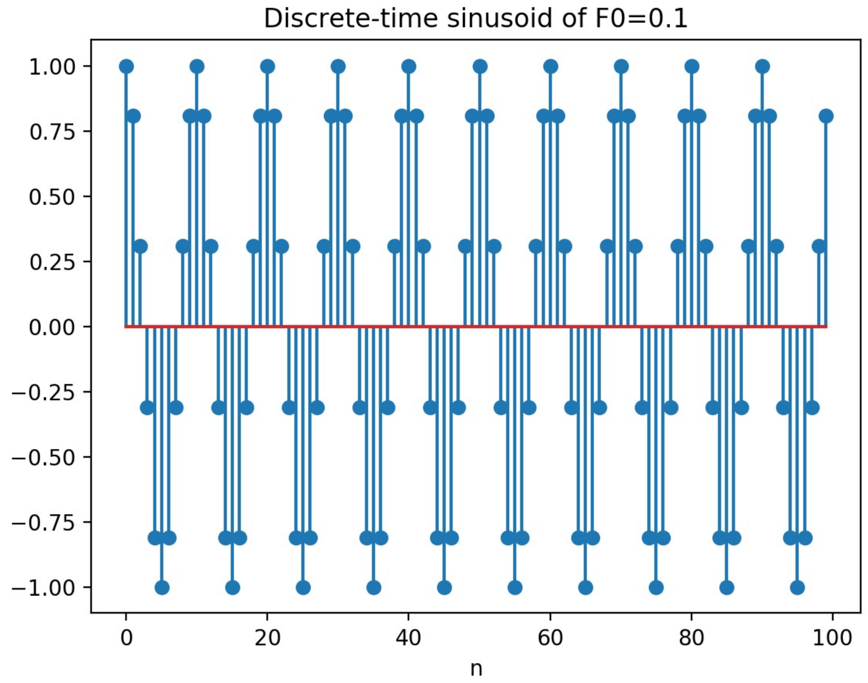


Sinusoidal Sequence: Python Code

Python Code

```
import matplotlib.pyplot as plt
import numpy as np
pi = np.pi

f = 0.1
L = 100
n = np.arange(L)
x = np.cos(2*pi*f*n)

plt.stem(x)
plt.title('Sinusoid Sequence of f=0.1')
plt.xlabel('n')
plt.show()
```

$$x[n] = \cos(2\pi(0.1)n)$$



https://xn--llions-yua.jutge.org/upc-python-cookbook/signal-processing/signals.html

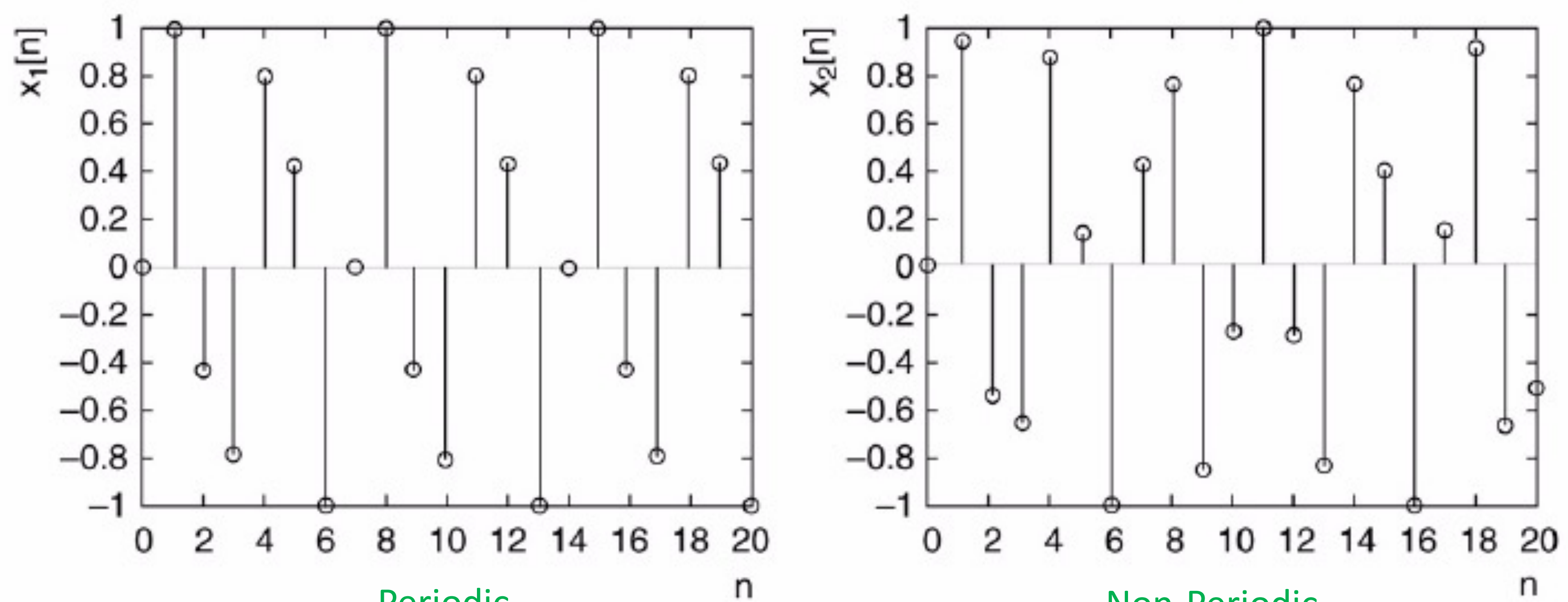
Periodicity of Sinusoidal Sequence

$$x[n] = A\cos(\omega n + \phi) \quad \text{or} \quad x[n] = A\sin(\omega n + \phi)$$

- Compared to analog counterpart, discrete-time sine and cosine signals are not always periodic sequences.
- Sinusoidal sequences are periodic only when $2\pi/\omega$ is a ratio of integers N/M.
- When $2\pi/\omega = N/M$, N is the number of samples in the discrete period, and M is the number of analog cycles that elapse while N samples are collected.
- An analog frequency f in Hz is related to its corresponding discrete-time angular frequency ω in radians with sampling rate of F_s through the equation

$$\omega = 2\pi\frac{f}{F_s}$$

Periodic and Non-Periodic Sinusoidal Examples



Periodic

(a) $x_1[n] = \sin\left(n\frac{4\pi}{7}\right)$

$$\frac{2\pi}{\omega} = \frac{2\pi}{\frac{4\pi}{7}} = \frac{7}{2}$$

Rational Number

Non-Periodic

(b) $x_2[n] = \sin\left(n\frac{13}{7}\right)$

$$\frac{2\pi}{\omega} = \frac{2\pi}{\frac{13}{7}} = \frac{14\pi}{13}$$

Irrational Number

Periodicity of Sinusoidal Sequence Example 1

A discrete-time signal is defined as $x[n] = \cos(2n)$

- Is this a periodic sequence?
- Find the first eight elements in the sequence.

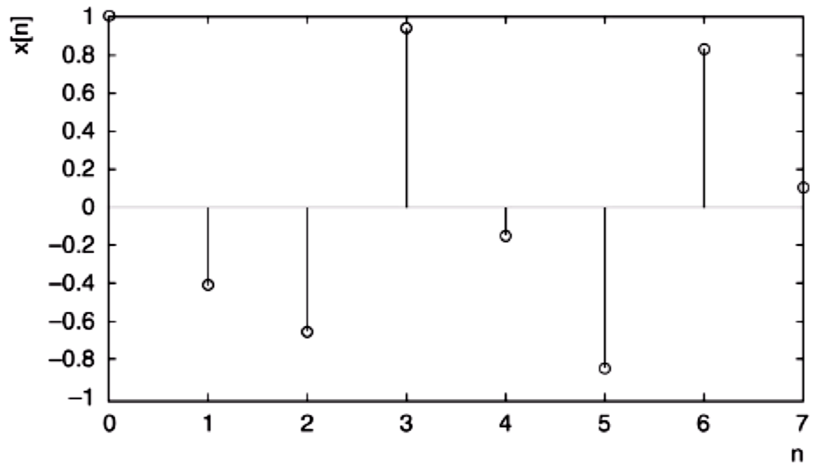
Periodicity of Sinusoidal Sequence Example 1

A discrete-time signal is defined as $x[n] = \cos(2n)$

- Is this a periodic sequence?
- Find the first eight elements in the sequence.

Solution:

- $\omega = 2$, then $2\pi/\omega = \pi$, this number is irrational and cannot be expressed in term of ratio of two integers.
- Therefore, this sinusoidal sequence is non-periodic.



Periodicity of Sinusoidal Sequence Example 2

A discrete-time signal is defined as $x[n] = \cos\left(\frac{4\pi}{5}n\right)$

- Is this a periodic sequence?
- Find the first eight elements in the sequence.

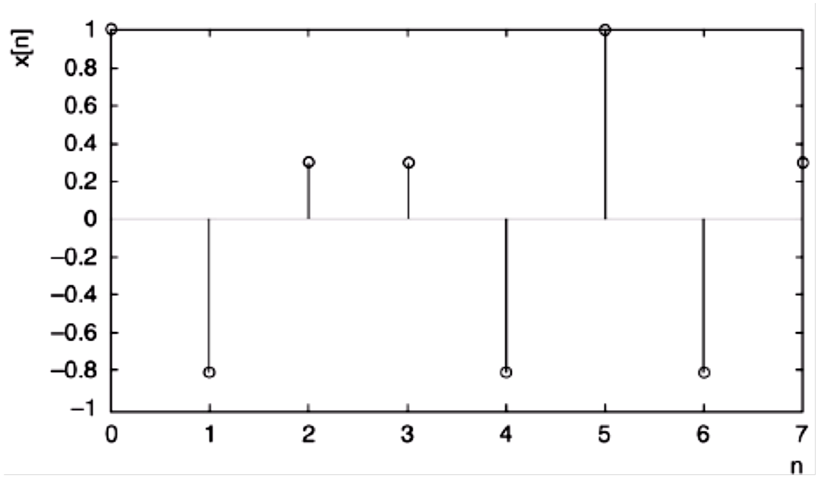
Periodicity of Sinusoidal Sequence Example 2

A digital signal is defined as $x[n] = \cos\left(\frac{4\pi}{5}n\right)$

- Is this a periodic sequence?
- Find the first eight elements in the sequence.

Solution:

- $\omega = 4\pi/5$, then $2\pi/\omega = 5/2$, this number is rational and can be expressed in term of ratio of two integers N/M.
- Therefore, this sinusoidal sequence is periodic.
- Where N=5 means sequence repeats every 5 samples, and M=2 means these 5 samples are collected over 2 complete cycles of the analog signal being sampled.



Composite Sequences (1)

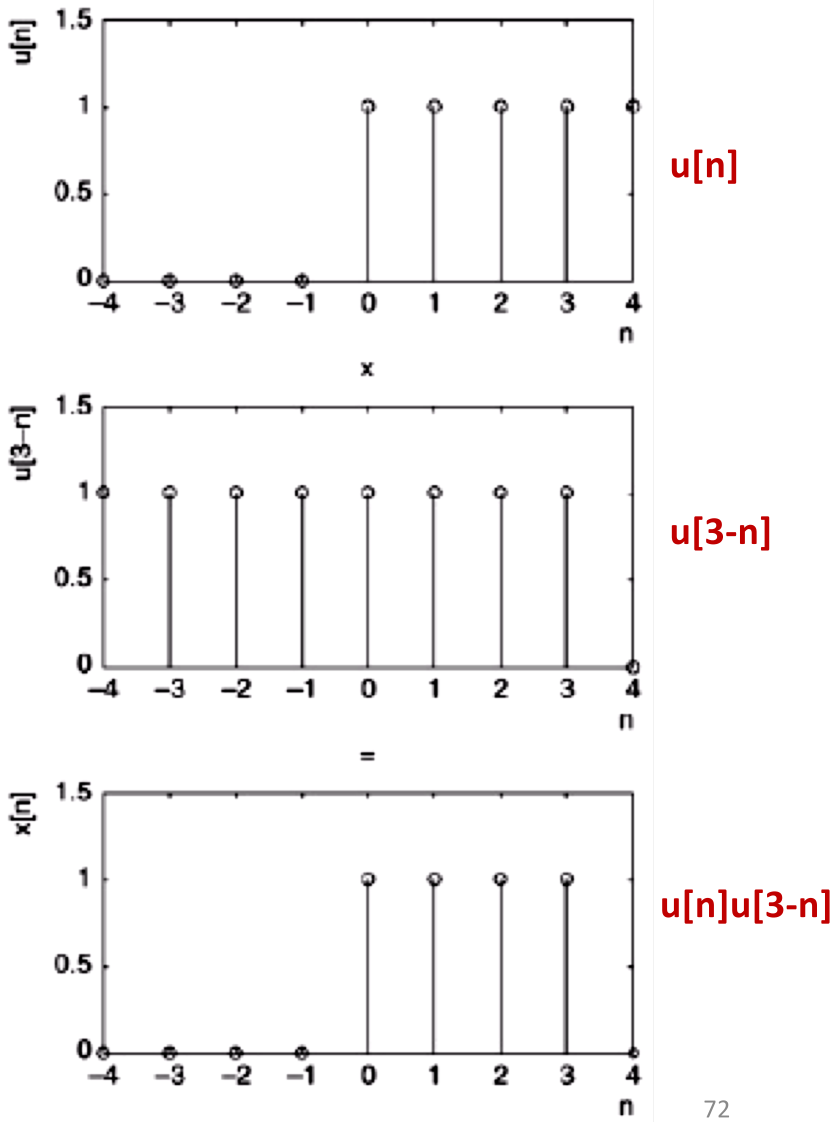
- They are the combinations of basic sequences.
- This give flexibility in defining discrete-time signals.
- To evaluate a composite sequence, each basic sequence (such as unit impulse, unit step, power, sinusoidal etc.) is constructed first, then the basic signals are multiplied, added or subtracted, as required.

Composite Sequences (2)

Draw a signal x[n] = u[n]u[3-n]

- The signal can also be expressed as a sum of Impulse functions.

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



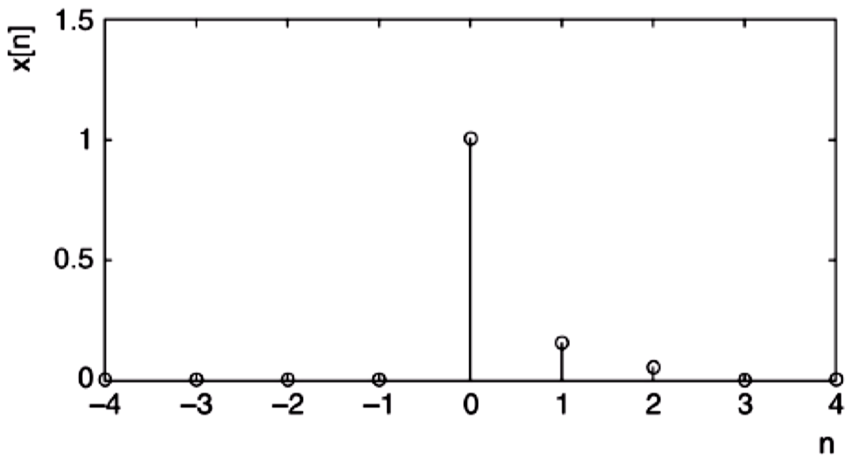
Composite Sequences (3)

Draw a signal $x[n] = e^{-2n}u[n]$

Composite Sequences (3)

Draw a signal $x[n] = e^{-2n}u[n]$

- First draw two basic signals (e^{-2n}, u[n]) and then multiply as shown in the figure.
- The u[n] has the effect of turning on the other function at n = 0.
- The u[n] is zero for n < 0, so x[n] is also for n < 0.
- The u[n] has a value of 1 for n ≥ 0, so x[n] is the same as e^{-2n} for n ≥ 0.



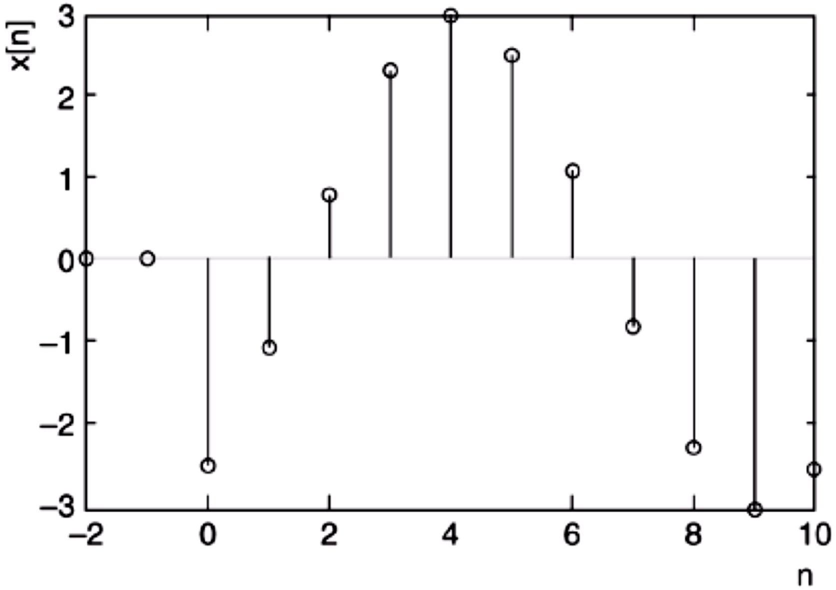
Right-Sized Exponential Sequence

L.M. Po

Composite Sequences (4)

Draw a signal x[n] = 3sin(nπ/5 - 1)u[n]

- First draw two basic signals and then multiply as shown in the figure.



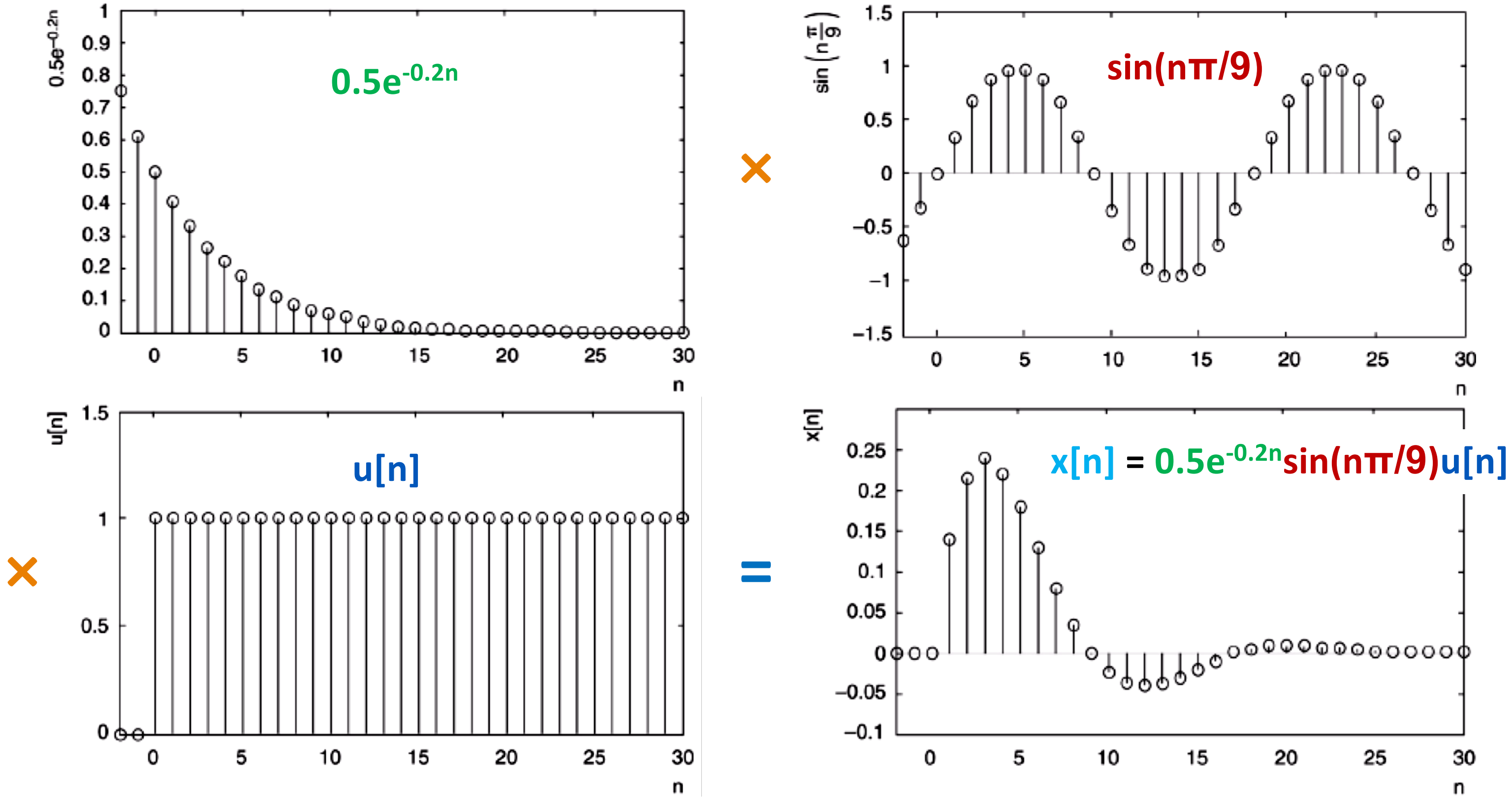
Right-Sized Periodic Sinusoidal Sequence with right shifted by one sample

L.M. Po

Composite Sequences (5)

Draw a signal $x[n] = 0.5e^{-0.2n}\sin(n\pi/9)u[n]$

- First draw three basic signals and then multiply to get the resultant damped



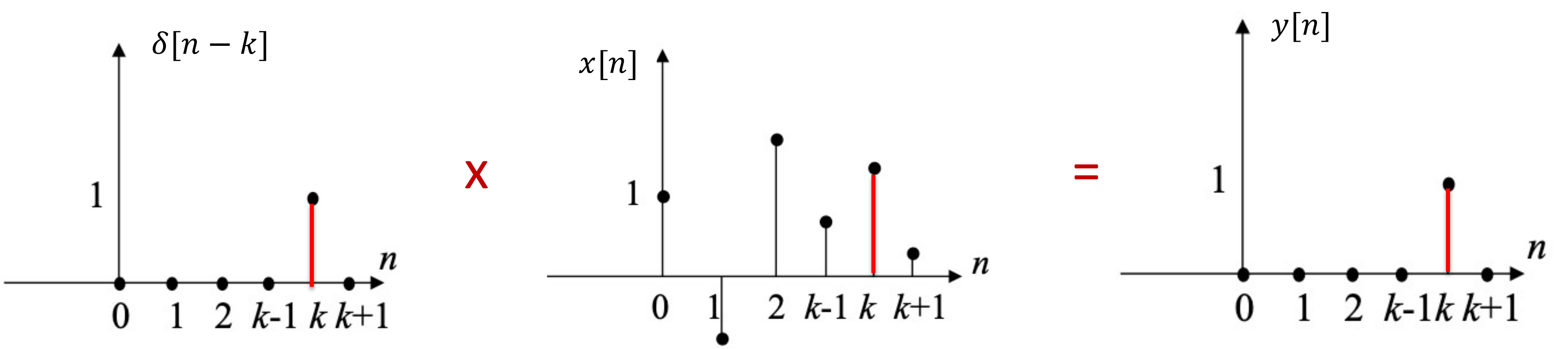
L.M. Po

Representation of Discrete-Time Signals by Sum of Scaled and Shifted Unit Impulse Signals (1)

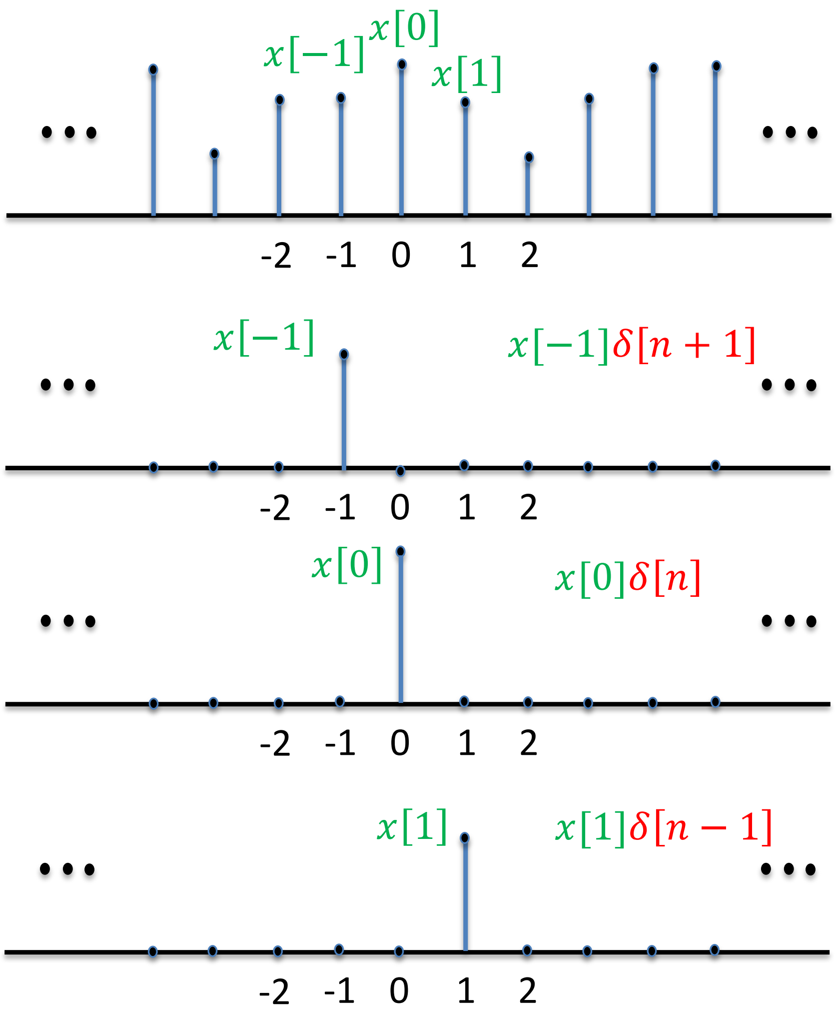
- A discrete-time signal, $x[n]$ may be shifted in time (delayed or advanced) by replacing the variables n with $(n-k)$ where $k>0$ is an integer
  - $x[n-k] \Rightarrow x[n]$ delayed by k samples
  - $x[n+k] \Rightarrow x[n]$ advanced by k samples
- For example, consider a shifted version of the unit impulse function. If we multiply an arbitrary signal $x[n]$ by this function, we obtain a signal that is zero everywhere, except at $n=k$.
  - $y[n] = x[n] \cdot \delta[n-k] = x[k] \cdot \delta[n-k]$

Representation of Discrete-Time Signals by Sum of Scaled and Shifted Unit Impulse Signals (2)

$$y[n] = x[k] \cdot \delta[n-k]$$



Unit Impulse based Composite Sequence Expression



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

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

Any discrete-time signals can be expressed in terms of summation of scaled and shifted unit sample sequence $x[k]\delta[n-k]$.

L.M. Po

Two-Dimensional Signals

(Optional)

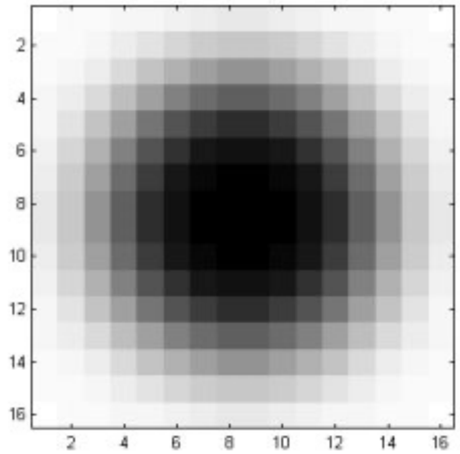
Two-Dimensional Digital Signals

- Digital images are two-dimensional digital signals.
- A digital image contains multiple rows and columns of pixels.
- Each pixel is assigned a color, and the combined effect of the colors of all the pixels is the desired image.
- For a gray scale image, each pixel is assigned a gray scale level that records the shading for that pixel.
- For an N-bit gray scale image, 2^N different shades are possible.
- Thus, 256 shades of gray are used in an 8-bit system.

Two-Dimensional Digital Signals

16 × 16 × 256 Digital Gray Scale Image

Row



Row

Column

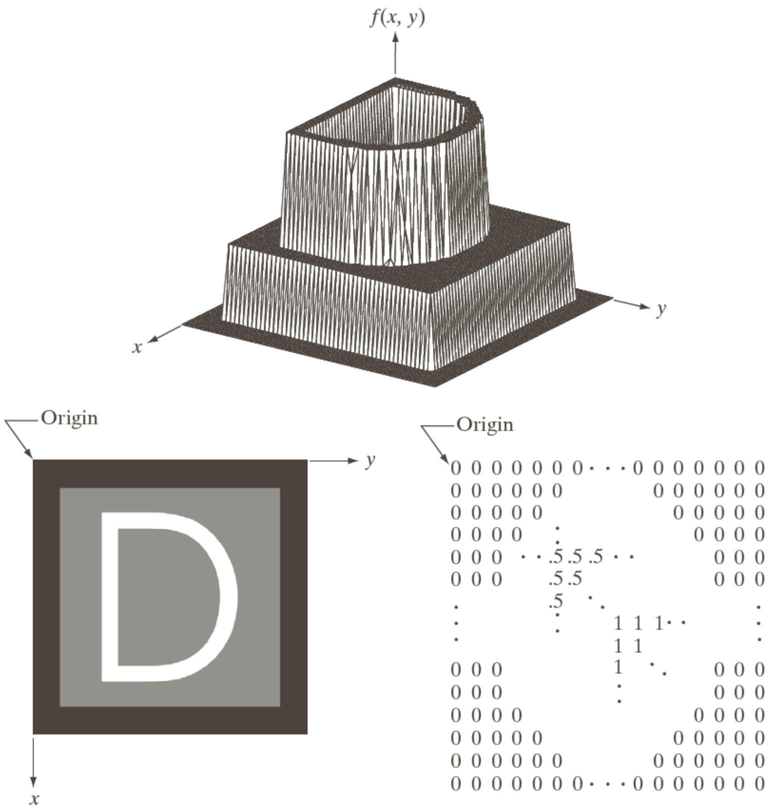
Gray Scale Values for 8-bit Digital Image

Column

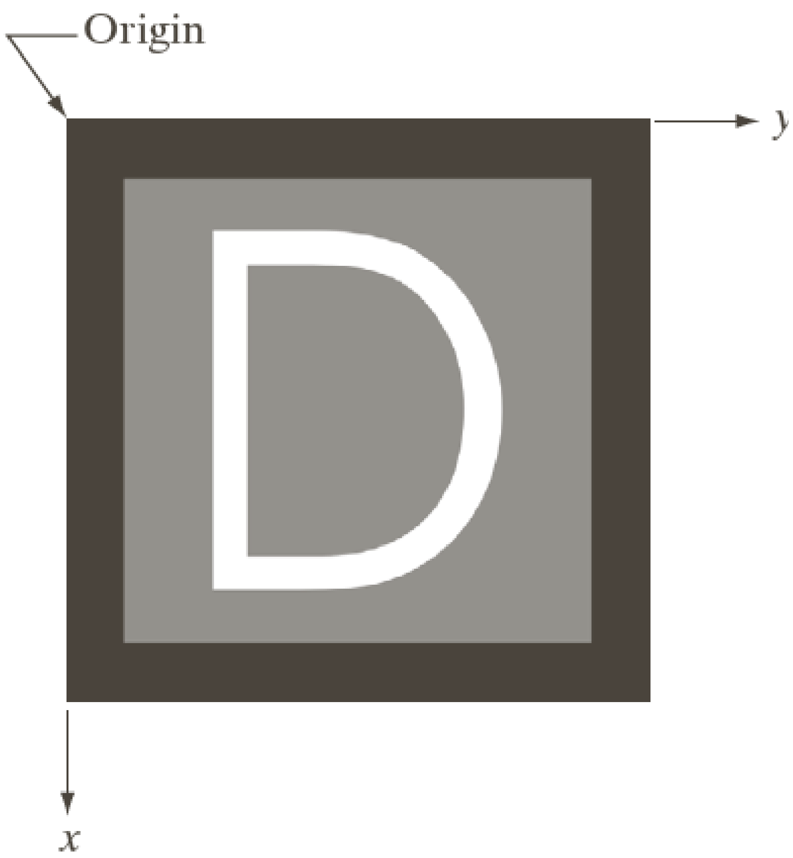
223	208	194	182	172	164	159	158	159	164	172	182	194	208	223	240
208	191	176	163	151	143	137	135	137	143	151	163	176	191	208	225
194	176	159	144	131	121	115	113	115	121	131	144	159	176	194	213
182	163	144	128	113	101	93	90	93	101	113	128	144	163	182	202
172	151	131	113	96	81	71	68	71	81	96	113	131	151	172	193
164	143	121	101	81	64	50	45	50	64	81	101	121	143	164	186
159	137	115	93	71	50	32	23	32	50	71	93	115	137	159	182
158	135	113	90	68	45	23	0	23	45	68	90	113	135	158	180
159	137	115	93	71	50	32	23	32	50	71	93	115	137	159	182
164	143	121	101	81	64	50	45	50	64	81	101	121	143	164	186
172	151	131	113	96	81	71	68	71	81	96	113	131	151	172	193
182	163	144	128	113	101	93	90	93	101	113	128	144	163	182	202
194	176	159	144	131	121	115	113	115	121	131	144	159	176	194	213
208	191	176	163	151	143	137	135	137	143	151	163	176	191	208	225
223	208	194	182	172	164	159	158	159	164	172	182	194	208	223	240
240	225	213	202	193	186	182	180	182	186	193	202	213	225	240	255

 82

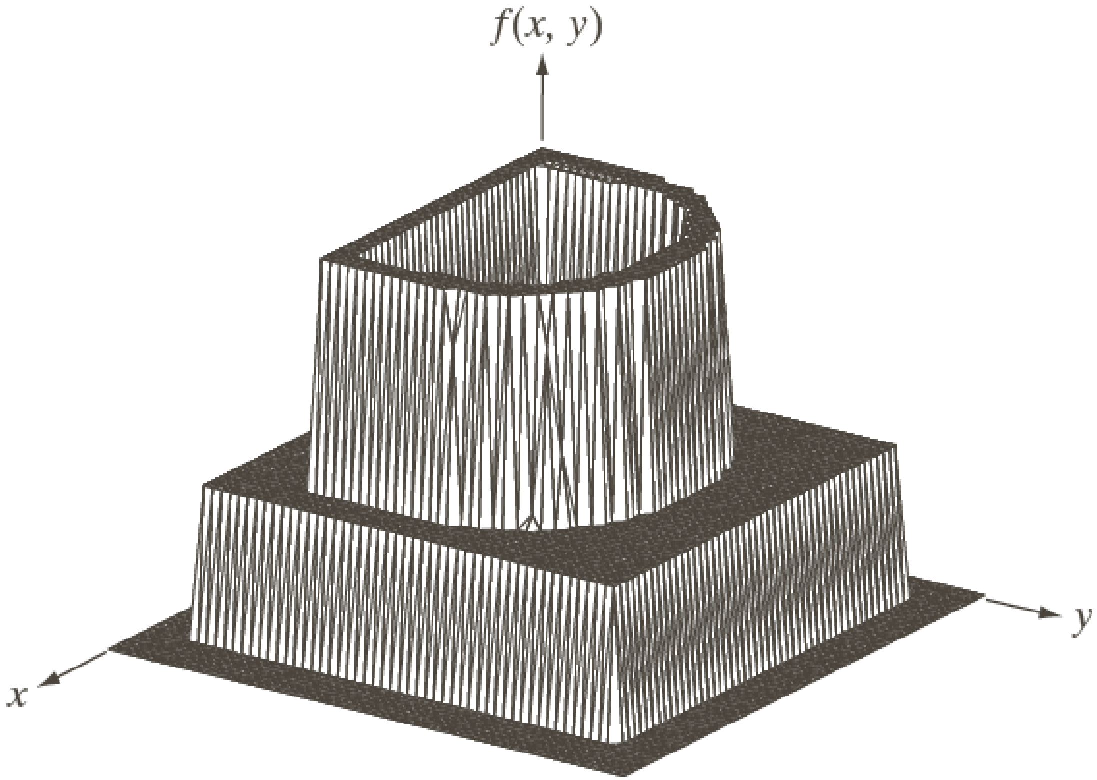
Image Representation



Gray Level Color Representation



3-D Plot Representation



Number Representation

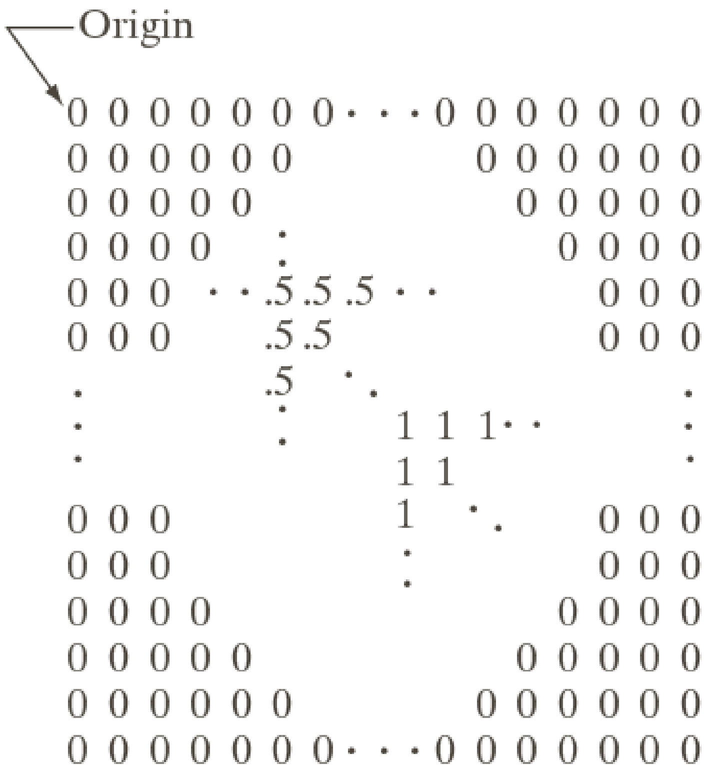
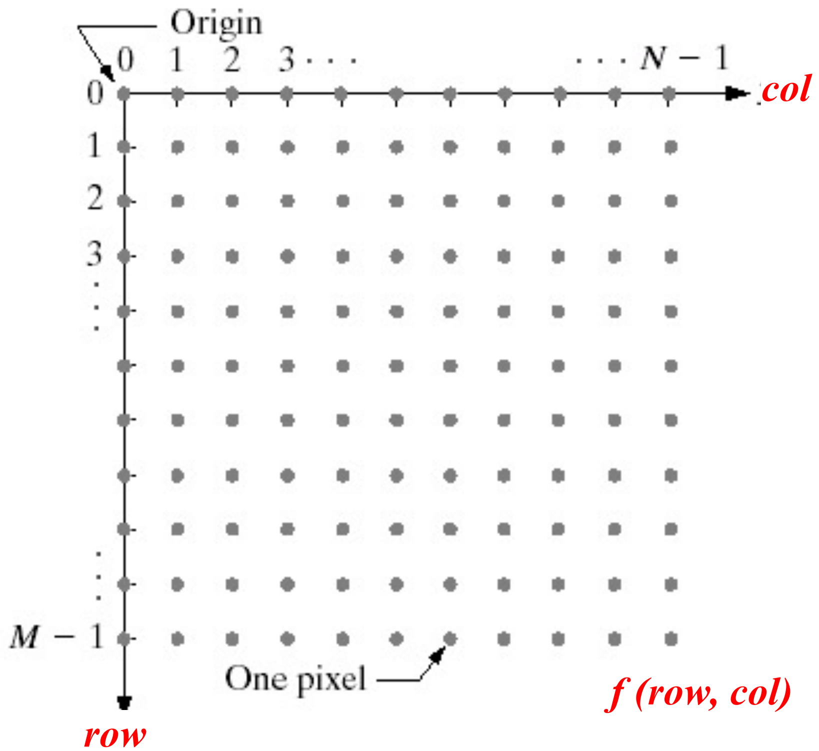


Image Coordination Convention

- An image is a function defined on a 2D coordinate f(*x*,*y*).
- The value of f(*x*,*y*) is the intensity.
- 3 such functions can be defined for a color image, each represents one color component.
- A digital image can be represented as a matrix.



Matrix Representation

- The representation of an M×N image as,

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \ldots & f(0,N-1) \\ f(1,0) & f(1,1) & \ldots & f(1,N-1) \\ \ldots & \ldots & \ldots & \ldots \\ f(M-1,0) & f(M-1,1) & \ldots & f(M-1,N-1) \end{bmatrix}$$

- Or the representation of an M×N numerical array as,

$$A=\begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \cdots & \cdots & \cdots & \cdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

Two-Dimensional Digital Signals

- The total number of bits needed to represent a digital image depends on the number of pixels and the number of bits assigned to each pixel. Specifically,

Total Number of Bits = (# Rows)(# Columns)(# Bits per Pixel)

Amount of Image Data

For an MxN image:

- Number of bits per pixel per channel: **k bits**
- Number of gray levels per pixel per channel: **L = 2^k levels**
- Pixel value dynamic range: **[0, L - 1]**
- Number of bits required to store an gray-scale image, **b:** **b = MNk bits**
- Number of bytes per image, B: **B = MNk/8 Bytes**
- Number of bytes with color image with RGB component: **B = 3*MNk/8 Bytes**

For Example, an 8-bits 1920x1080 Full HD gray level image,

- B = 1920 x 1080 x 8/8 = 2.0736 MB