جامعة ديالى
كلية الهندسة
www.engineering.uodiyala.edu.iq
University Of Diyala – College Of Engineering

Digital Signal Processing

Course Instructor
Lecturer: WARQAA SHAHER

Lecture No. 1: Introduction

Third Class
Department of Computer and Software Engineering

http://www.engineering.uodiyala.edu.iq/
https://www.facebook.com/Engineer.College1

Course Overview

- Signals and Systems.
- Classification of Discrete Digital Systems.
- Time and Frequency Domains Analysis.
- Signal Transformation Methods: Fourier, Wavelet and Z-transform.
- Digital Filter Types: FIR and IIR Filters.
- Digital Filter Design.
- Analog Filter Design.
- DSP Applications.

Books

- J.G. Proakis and D.G. Manolakis, **Digital Signal Processing,** *4rd edition*, Prentice-Hall , 2006.

- R.G Lyons, **Understanding Digital Signal processing**, *3rd edition*, Prentice-Hall, (Amazon's top-selling for five straight year) ,2011.

- Monsons Hays, **Schaums Outline of Digital Signal processing**, *2nd edition* ,McGraw-Hill Companies, 2012.

- Richard, **The Essential Guide to Digital Signal Processing,** *1st edition* Prentice-Hall ,ePUB, 2014.

- J.G. Proakis , **Digital Signal Processing Using MATLAB,** *3rd edition*, Cengage Learning , 2012.

What is Digital Signal Processing?

Signal: It can be broadly defined as any physical quantity that varies as a time and/or space and has the ability to convey information, examples of these signals are:

- **Electrical signals**: currents and voltages in AC circuits, radio communications signals, video signals etc.
- **Mechanical signals**: sound or pressure waves, vibrations in a structure, earthquakes, etc.
- **Biomedical signals**: electro-encephalogram, lung and heart monitoring, X-ray and other types of images.
- **Finance**: time variations of a stock value or a market index.

Digital Signal: operating by the use of discrete signals to represent data in the form of numbers.

What is Digital Signal Processing?

Processing: a series operations performed according to programmed instructions.

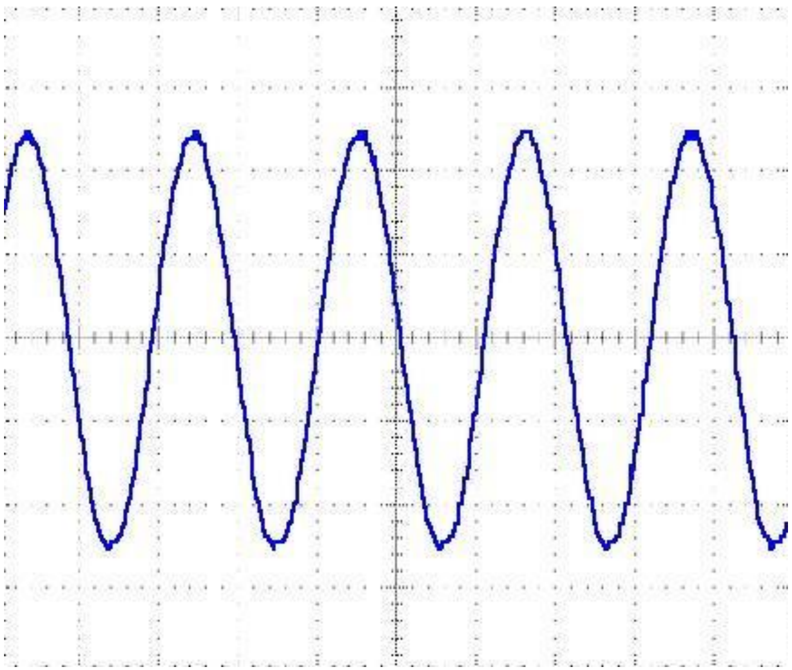
→ **changing or analysing information which is measured as discrete sequences of numbers**

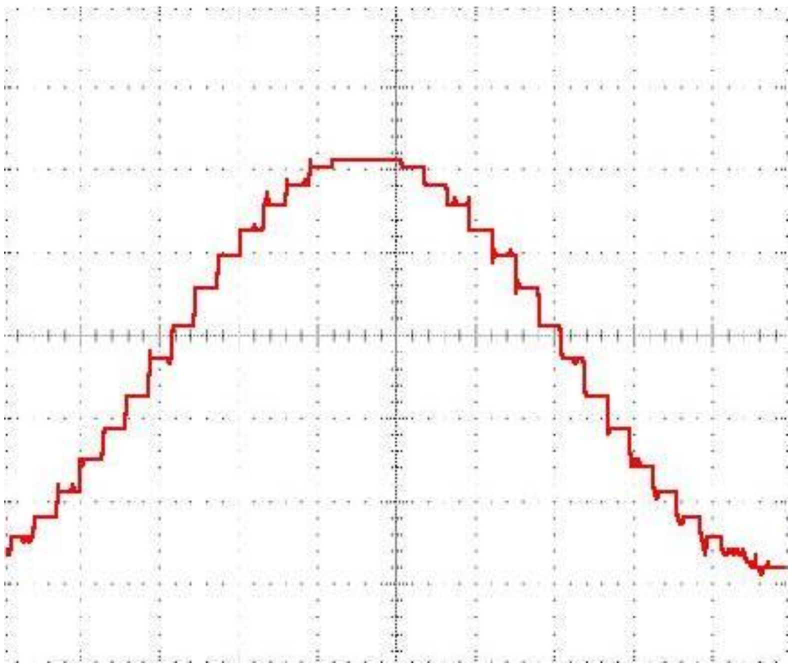
"Learning digital signal processing is not something you accomplish; it's a journey you take".

R.G Lyons, Understanding Digital Signal processing

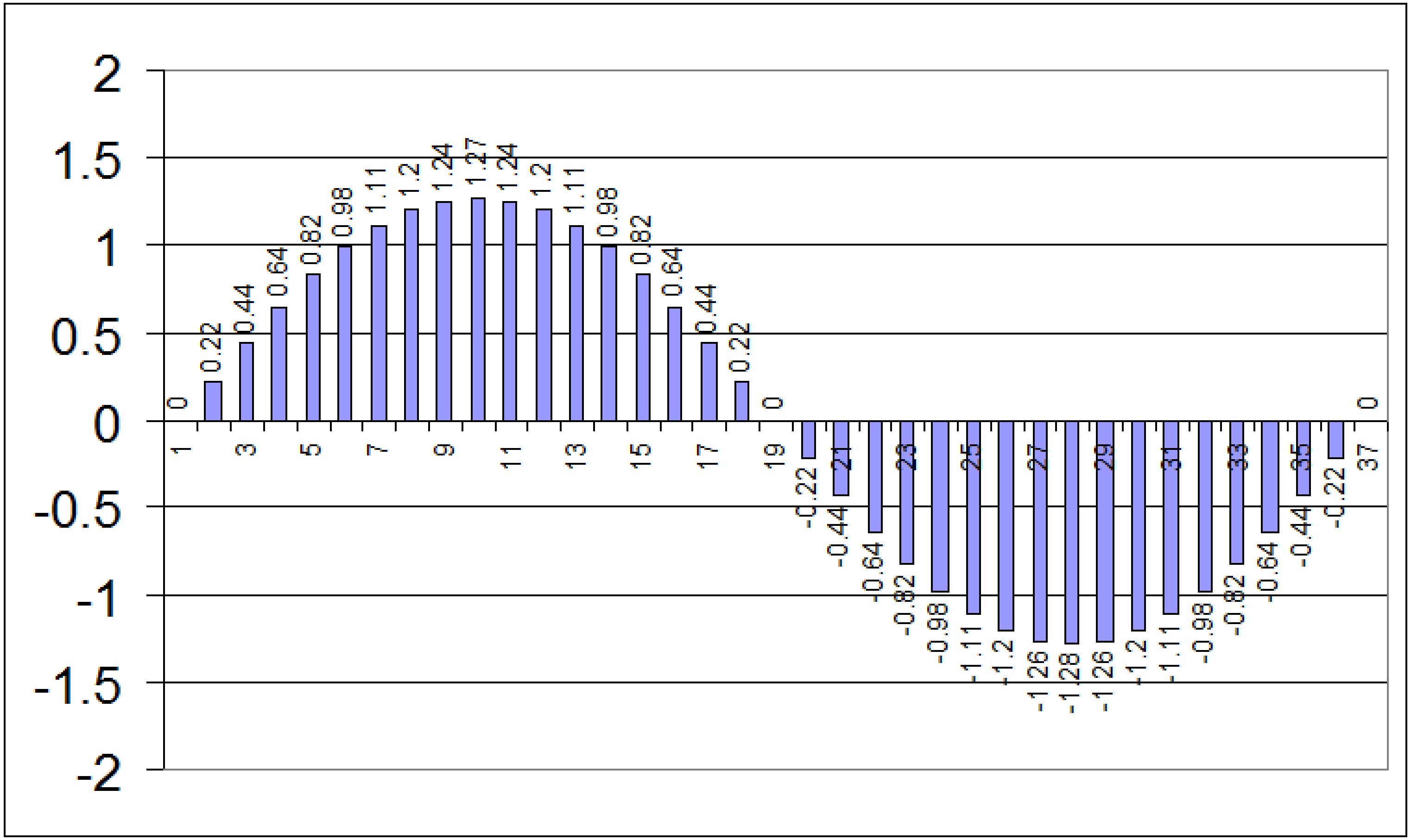
What is Digital Signal Processing?

Converting a continuously changing waveform (analog) into a series of discrete levels (digital)



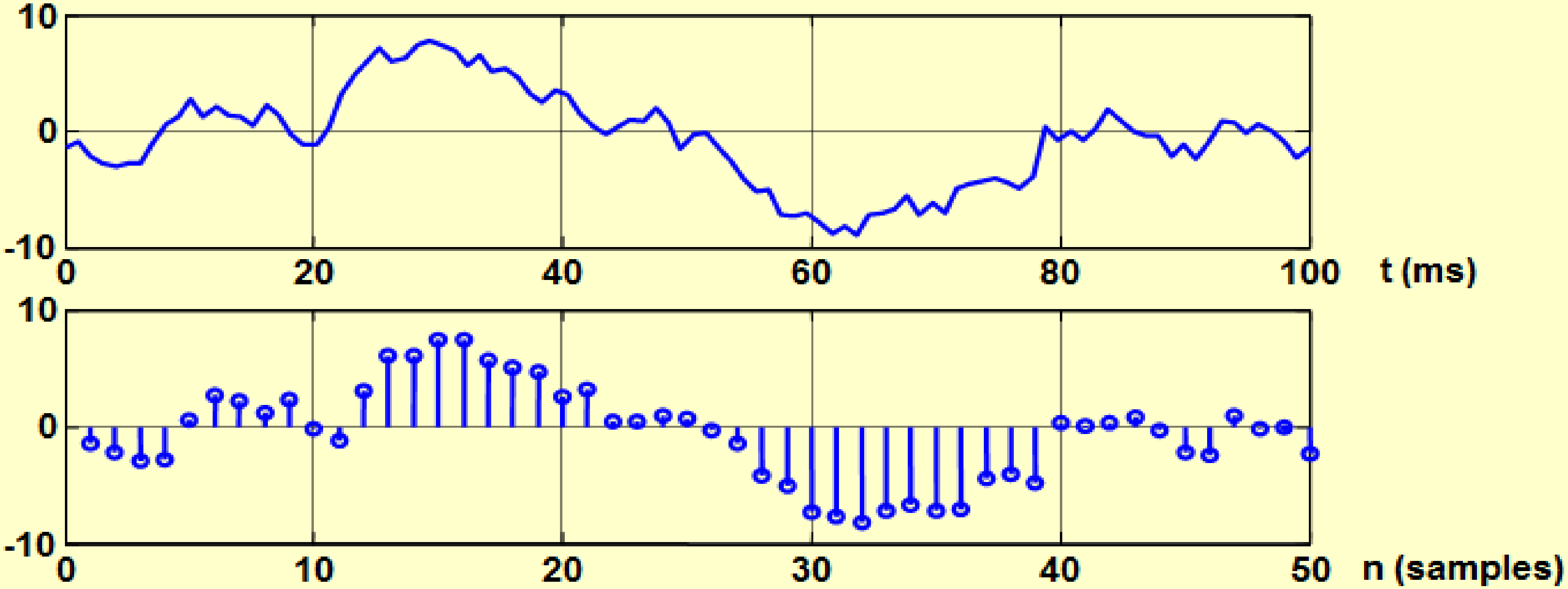


What is Digital Signal Processing?

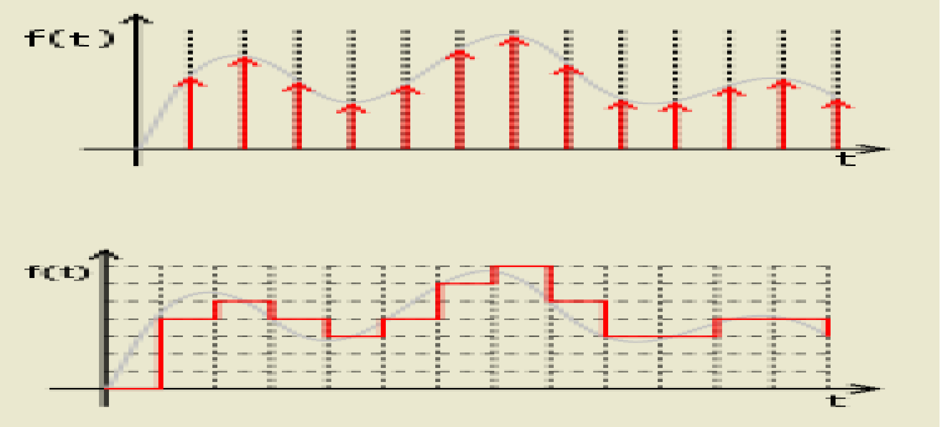


What is Digital Signal Processing?

- Discrete-time signals are represented by sequence of numbers
  - The n^{th} 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 at $x_c(nT)$
  - Where T is the sampling period



What is Digital Signal Processing?



discrete signal is discrete in time but continuous in amplitude.

digital signal is discrete in both time and amplitude.

The Concepts of Signals and Systems

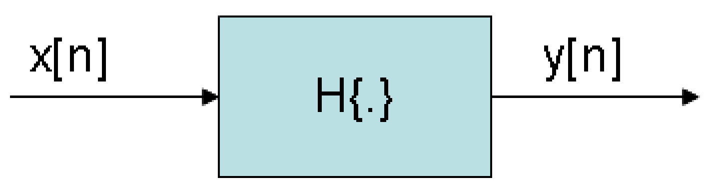
- The most convenient mathematical representation of a signal is via the concept of a function, say $\boldsymbol{x(t)}$. In this notation:

  $\boldsymbol{x}$ represents the dependent variable (e.g. voltage, pressure, etc.)
  $\boldsymbol{t}$ represents the independent variable (e.g. time, space, etc.).

- Depending on the nature of the independent and dependent variables, different types of signals can be identified such as:
  - ✓ Analog signals
  - ✓ Discrete signals
  - ✓ Digital signals
  - ✓ Multi-channel signals
  - ✓ Multi-dimensional signals

The Concepts of Signals and Systems

- **System**: A physical entity that operates on a set of primary signals (the inputs) to produce a corresponding set of resultant signals (the output).
- **The operations, or processing, may take several forms**: modification, combination, decomposition, filtering, extraction of parameters, etc.
- **System characterization**: a system can be represented mathematically as a transformation between two signal sets , as in $\boldsymbol{x[n]} \rightarrow \mathbf{y[n]}$:



The Concepts of Signals and Systems

- Depending on the nature of the signals on which the system operates, different basic types of systems may be identified:
  - Analog or continuous-time system: the input and output signals are analog in nature.
  - Discrete-time system: the input and output signals are discrete.
  - Digital system: the input and outputs are digital.
  - Mixed system: a system in which different types of signals (i.e.analog,discrete and/or digital) coexist.

Discussion:

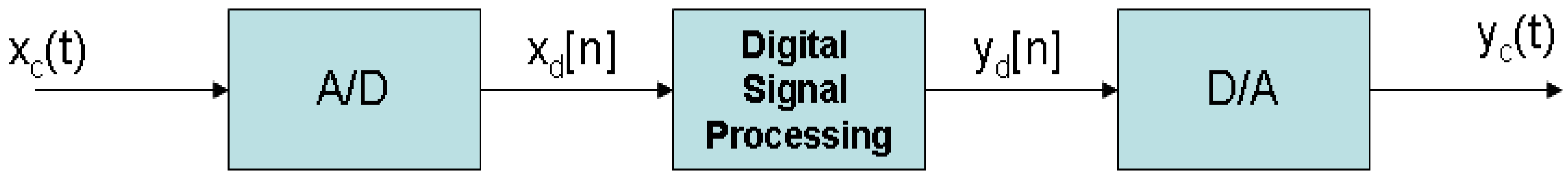
- Early education in engineering focuses on the use of calculus to analyze various systems and processes at the analog level:
  - motivated by the prevalence of the analog signal model
  - e.g.: circuit analysis using differential equations
- Yet, due to extraordinary advances made in micro-electronics, the most common/powerful processing devices today are digital in nature.
- Thus, there is a strong, practical motivation to carry out the processing of analog real-world signals using such digital devices.
- This has lead to the development of an engineering discipline know as digital signal processing DSP.

Digital Signal Processing:

- In its most general form, DSP refers to the processing of analog signals by means of discrete-time operations implemented on digital hardware.

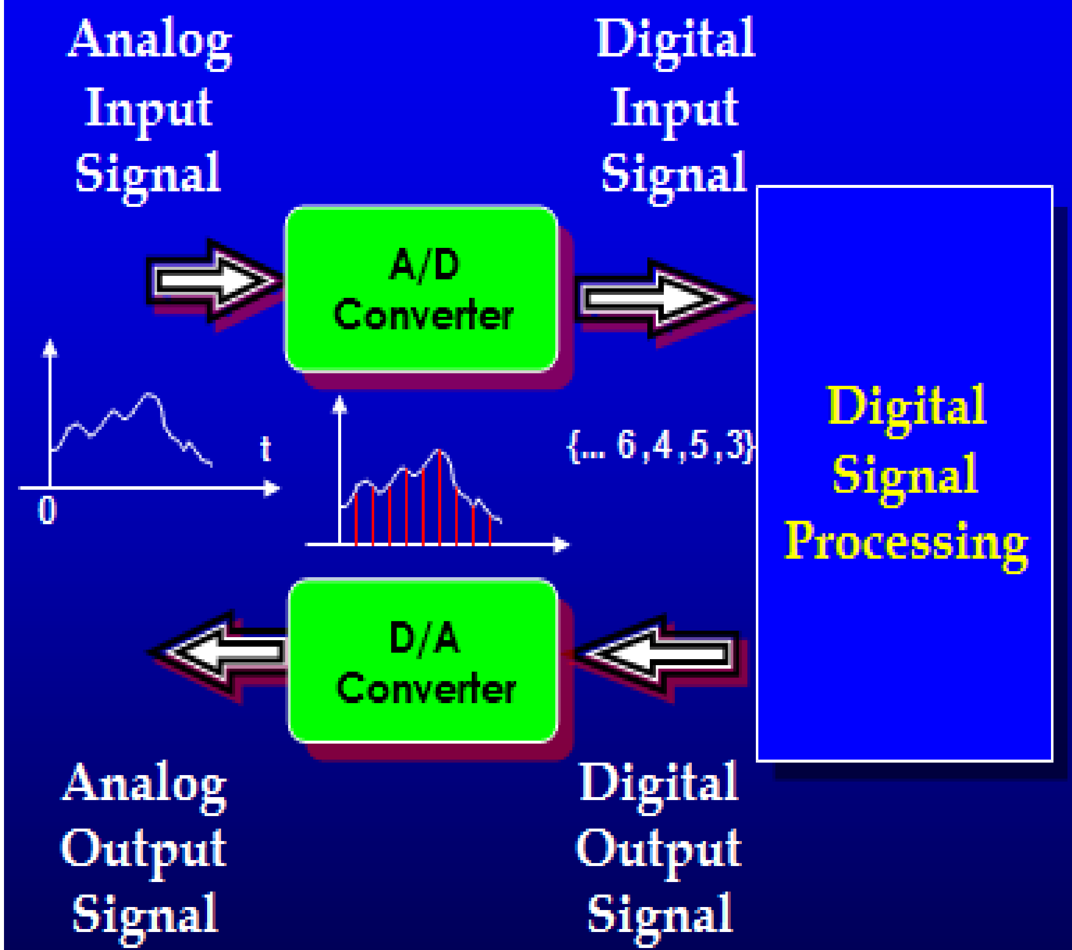
Basic components of a DSP System

- In its most general form, a DSP system will consist of three main components, as illustrated below:



- The analog-to-digital (A/D) converter transforms the analog signal $x_c(t)$ at the system input into a digital signal $x_d[n]$.
- The digital system performs the desired operations on the digital signal $x_d[n]$ and produces a corresponding output $y_d[n]$ also in digital form.
- The digital-to-analog (D/A) converter transforms the digital output $y_d[n]$ into an analog signal $y_c(t)$ suitable for interfacing with the outside world.

Basic components of a DSP System



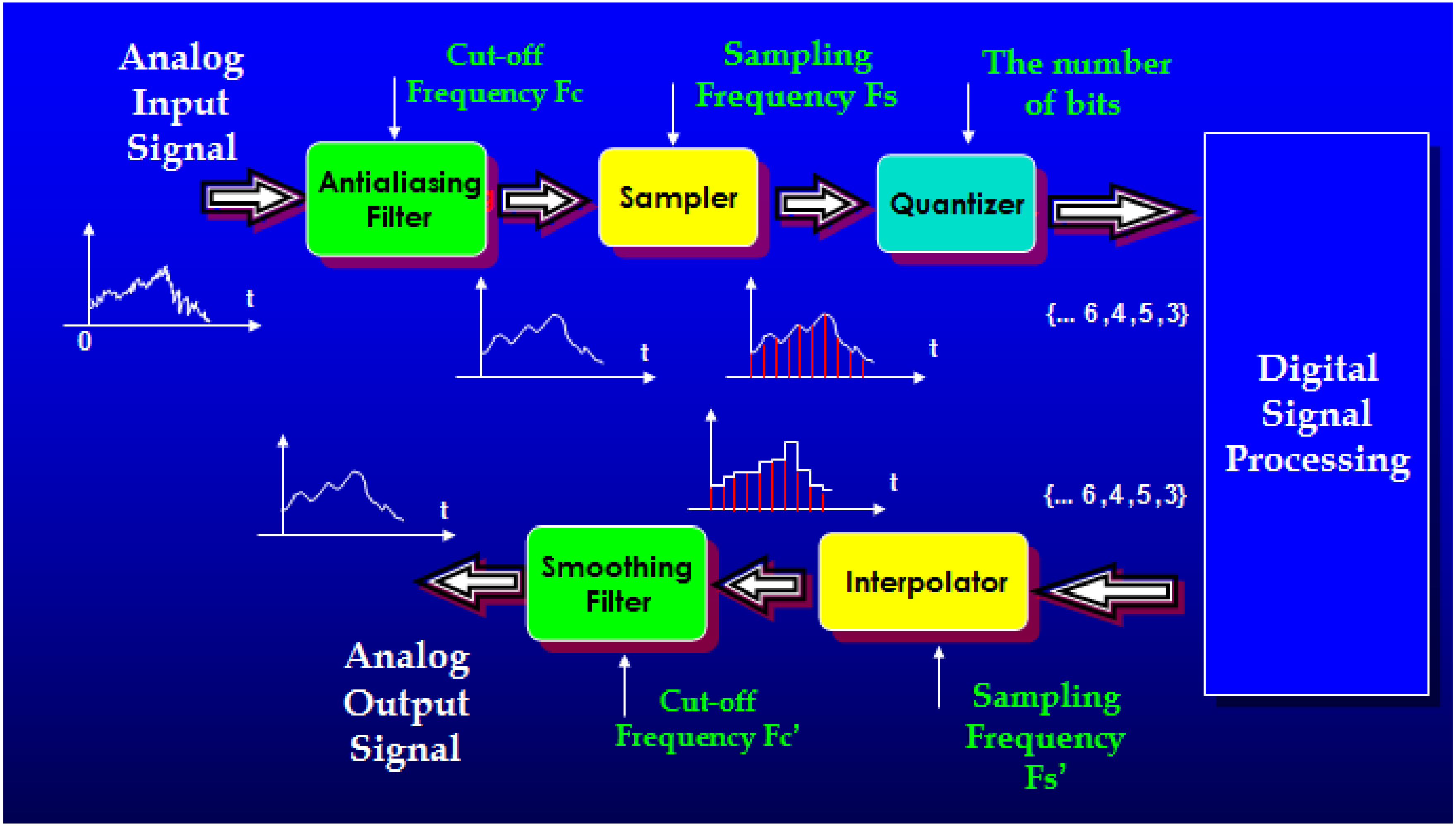
A/D Converter

Converts an analog signal into a sequence of digits

D/A Converter

Converts a sequence of digits into an analog signal

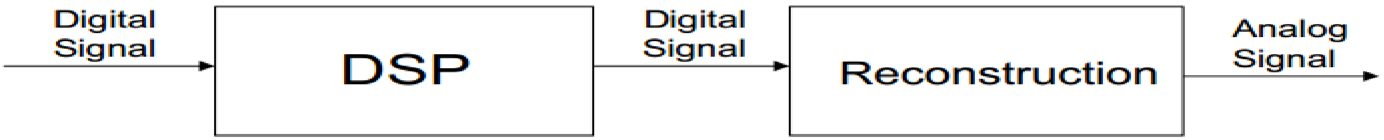
Basic components of a DSP System



DSP Implementation –Analog/Digital Conversion

To implement DSP we must be able to:

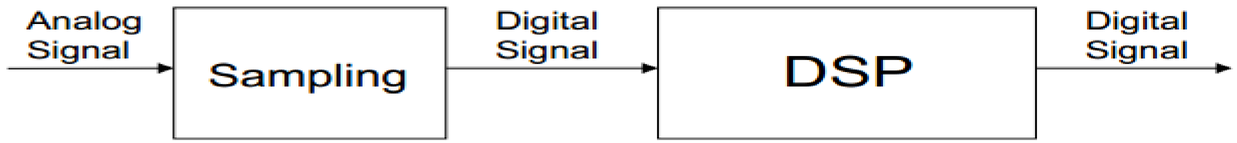
(1) perform numerical operations including, for example, additions, multiplications, data transfers and logical operations either using computer or special-purpose hardware.



(2) convert the digital information, after being processed back to an analog signal – involves digital-to- analog conversion and reconstruction .
e.g. text-to-speech signal (characters are used to generate artificial sound)

DSP Implementation –Analog/Digital Conversion

To implement DSP we must be able to:



3) convert analog signals into the digital information - sampling & involves analog-to-digital conversion.

e.g. Touch-Tone system of telephone dialling (when button is pushed two sinusoid signals are generated (tones) and transmitted, a digital system determines the frequencies and uniquely identifies the button – digital (1 to 12) output

perform both A/D and D/A conversions

e.g. digital recording and playback of music (signal is sensed by microphones, amplified, converted to digital, processed, and converted back to analog to be played.

DSP Chips : Special Purpose Hardware

Introduction of the microprocessor in the late 1970's and early 1980's meant DSP techniques could be used in a much wider range of applications.

DSP chip – a *programmable* device, with its own native instruction code

designed specifically to meet numerically-intensive requirements of DSP

capable of carrying out millions of floating point operations per second

Bluetooth headset

Household appliances

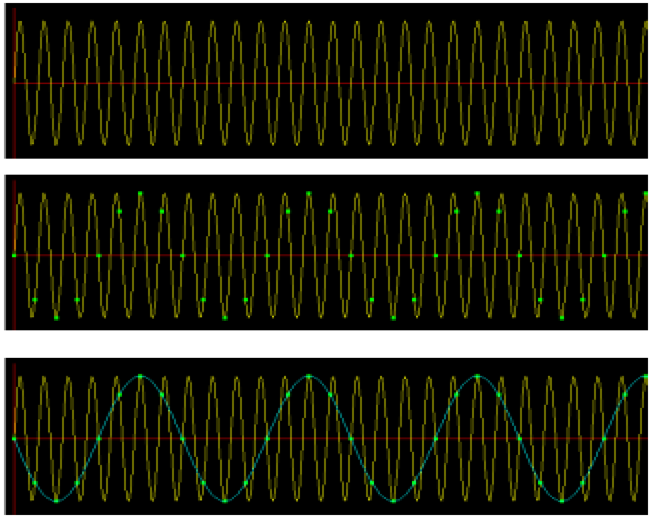
Home theatre system

Limitations of DSP-Aliasing

Most signals are analog in nature, and have to be sampled.

➡ loss of information

we only take samples of the signals at intervals and don't know what happens in between



Gjendemsjø, A. Aliasing Applet, Connexions, http://cnx.org/content/m11448/1.14

➡ **aliasing**

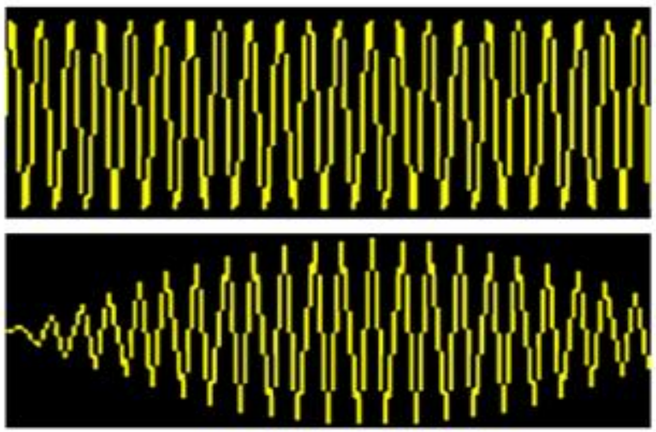
cannot distinguish between higher and lower frequencies

(recall from 1B Signal and Data Analysis)

Sampling theorem: to avoid aliasing, sampling rate must be at least twice the maximum frequency component (`bandwidth') of the signal

Limitations of DSP - Antialiasing Filter

- Sampling theorem says there is enough information to reconstruct the signal, which does not mean sampled signal looks like original one



Each sample is taken at a slightly earlier part of a cycle

➡ correct reconstruction is not just connecting samples with straight lines

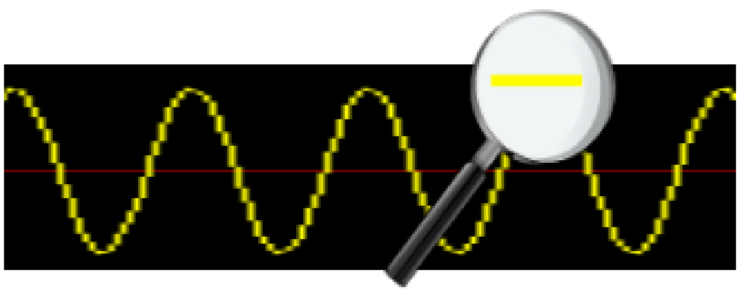
needs antialias filter (to filter out all high frequency components before sampling) and the same for reconstruction – it does remove information though

Limitations of DSP – Frequency Resolution

Most signals are analog in nature, and have to be sampled

→ loss of information

- we only take samples for a limited period of time



→ **limited frequency resolution**

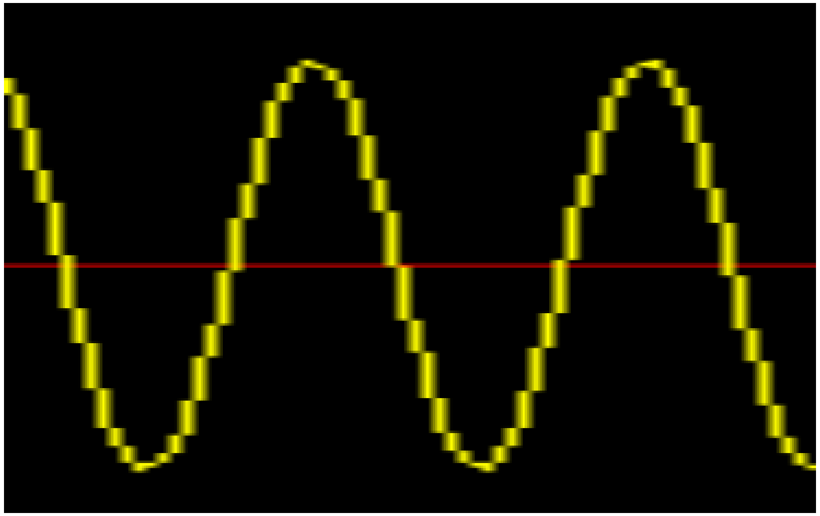
does not pick up "relatively" slow changes

Limitations of DSP – Quantization Error

Most signals are analog in nature, and have to be sampled

➡ loss of information

- limited (by the number of bits available) precision in data storage and arithmetic



➡ **quantisation error**

smoothly varying signal represented by "stepped" waveform

Why still do it?

- Digital system can be simply reprogrammed for other applications / ported to different hardware / duplicated (Reconfiguring analog system means hardware redesign, testing, verification)
- DSP provides better control of accuracy requirements (Analog system depends on strict components tolerance, response may drift with temperature)
- Digital signals can be easily stored without deterioration (Analog signals are not easily transportable and often can't be processed off-line)
- More sophisticated signal processing algorithms can be implemented

(Difficult to perform precise mathematical operations in analog form)

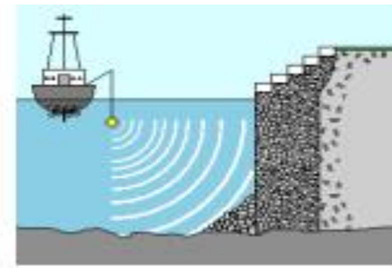
Advantages of Digital over Analog Signal Processing

Advantages:

- Robustness:
- Signal levels can be regenerated.
- Precision not affected by external factors
- Storage capability:
- DSP system can be interfaced to low-cost devices for lasting storage
- allows for off-line computations
- Flexibility:
- Easy control of system accuracy via changes in sampling rate and number of representation bits.
- Software programmable → reconfiguring the DSP operations simply by changing the program.
- Structure:
- Easy interconnection of DSP blocks (no loading problem)
- Possibility of sharing a processor between several tasks

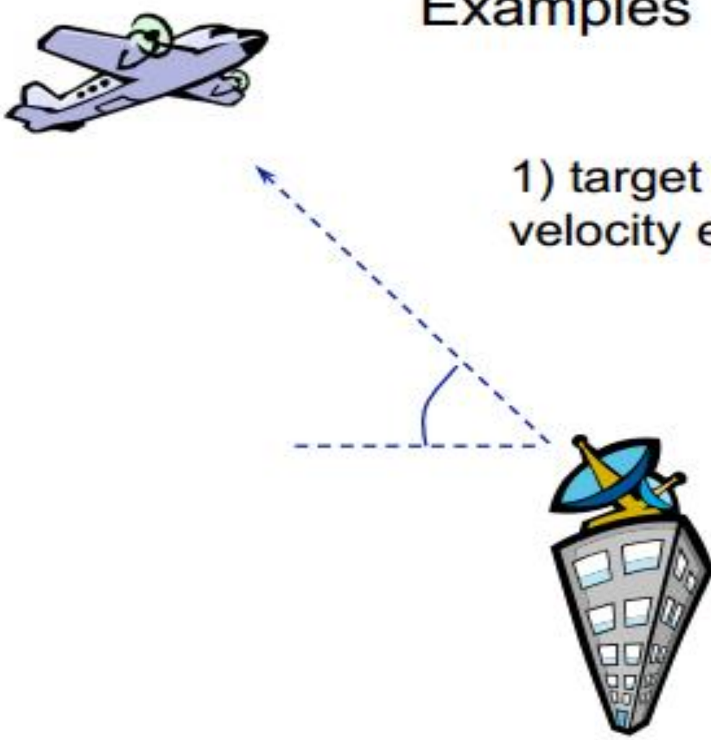
Disadvantages:

- Cost/complexity added by A/D and D/A conversion.
- Input signal bandwidth is technology limited.
- Quantization effects.



Radar and Sonar:

Examples



1) target detection – position and velocity estimation

2) tracking

Applications of DSP: Biomedical

Biomedical: analysis of biomedical signals, diagnosis, patient monitoring, preventive health care, artificial organs

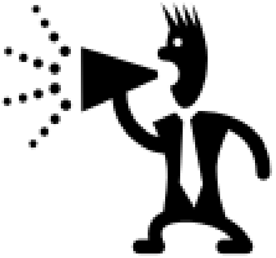
Examples:

1) electrocardiogram (ECG) signal – provides doctor with information about the condition of the patient's heart

2) electroencephalogram (EEG) signal – provides Information about the activity of the brain

Applications of DSP: Speech

Speech applications:

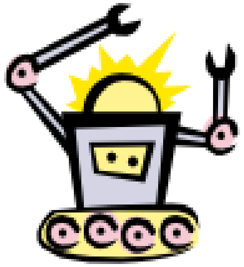


Examples

1) noise reduction – reducing background noise
in the sequence produced by a sensing device (microphone)

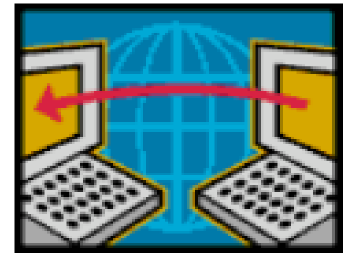
2) speech recognition – differentiating between various speech sounds

3) synthesis of artificial speech – text to speech systems for blind



Applications of DSP: Communications

Communications:



Examples

1) telephony – transmission of information in digital form via telephone lines, modem technology, mobile phones

2) encoding and decoding of the information sent over a physical channel (to optimise transmission or to detect or correct errors in transmission)

Applications of DSP: Image Processing

Image Processing:

INFORMATION RETRIEVAL

Examples

1) content based image retrieval – browsing, searching and retrieving images from database

2) image enhancement

2) compression - reducing the redundancy in the image data to optimise transmission / storage

Applications of DSP: Music

Music Applications:

Examples:

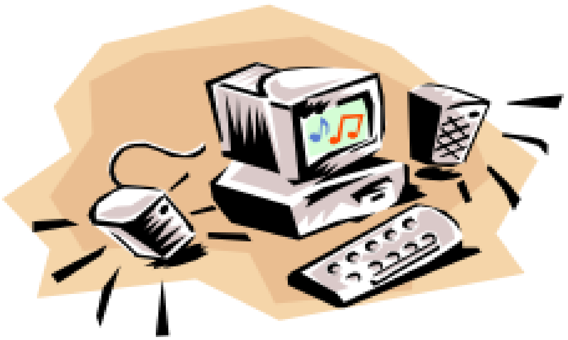
1) Recording

2) Playback

3) Manipulation (mixing, special effects)

Applications of DSP: Multimedia

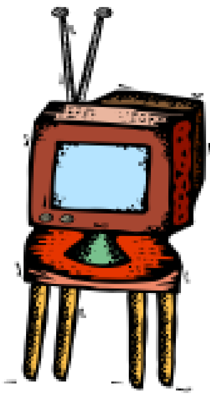
Multimedia:



generation storage and transmission of sound, still images, motion pictures

Examples:

1) digital TV



2) video conferencing

Applications of DSP

DSP

SPACE
- Space photograph enhancement
- Data compression
- Intelligent sensory analysis

MEDICAL
- Diagnostic imaging (MRI, CT)
- Electrocardiogram Analysis
- Medical image storage and retrieval

COMMERCIAL
- Image and sound compression for multimedia presentation.
- Movie special effects
- Video conference calling

TELEPHONE
- Video and data compression
- echo reduction
- signal multiplexing
- filtering

MILITARY
- Radar
- Sonar
- Ordnance Guidance
- Secure communication

INDUSTRIAL
- Oil and mineral prospecting
- Process monitoring and control
- Non-destructive testing
- CAD and design tools

SCIENTIFIC
- Earth quick recording and analysis
- Data acquisition
- Spectral Analysis
- Simulation and Modeling

جامعة ديالى
كلية الهندسة
www.engineering.uodiyala.edu.iq
University Of Diyala – College Of Engineering

Digital Signal Processing

Course Instructor
Lecturer: WARQAA SHAHER

Lecture No. 2: Classification of DSP Systems

Third Class
Department of Computer and Software Engineering

http://www.engineering.uodiyala.edu.iq/
https://www.facebook.com/Engineer.College1

Lecture Outline

- **Classification of Signals**
- **Basic Types of Digital Signals:**
  1) Unit Step
  2) Impulse
  3) Ramp
  4) Exponential
  5) Cosine
- **Classification of DSP Systems:**
  1) Causality
  2) linearity
  3) Time Invariant
  4) Stability
- **Characterization of Digital Filters:**

  (1) Recursive (2) Non-Recursive

Classification of Signals

- **Multichannel and Multidimensional Signals**

$$s_1(t) = A \sin 3\pi t$$

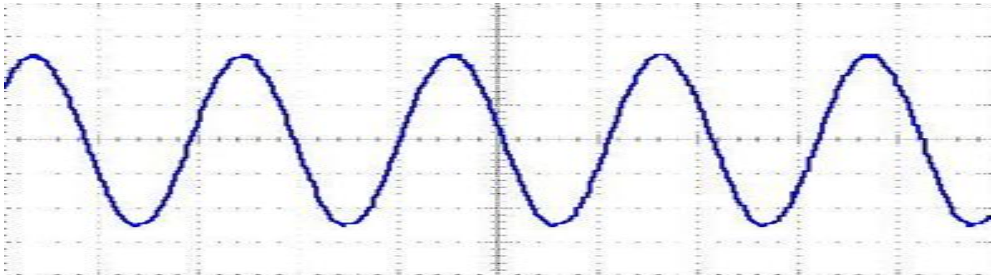
$$s_2(t) = Ae^{j3\pi t} = A \cos 3\pi t + jA \sin 3\pi t$$

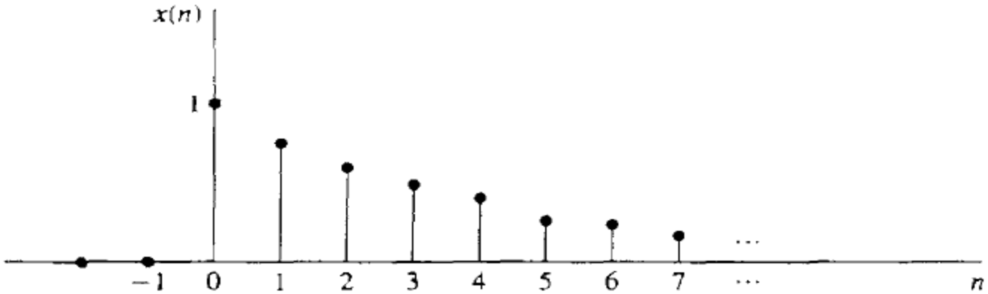
$$\mathbf{S}_3(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \end{bmatrix}$$

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

Classification of Signals

- **Continuous-Time and Discrete-Time Signals**



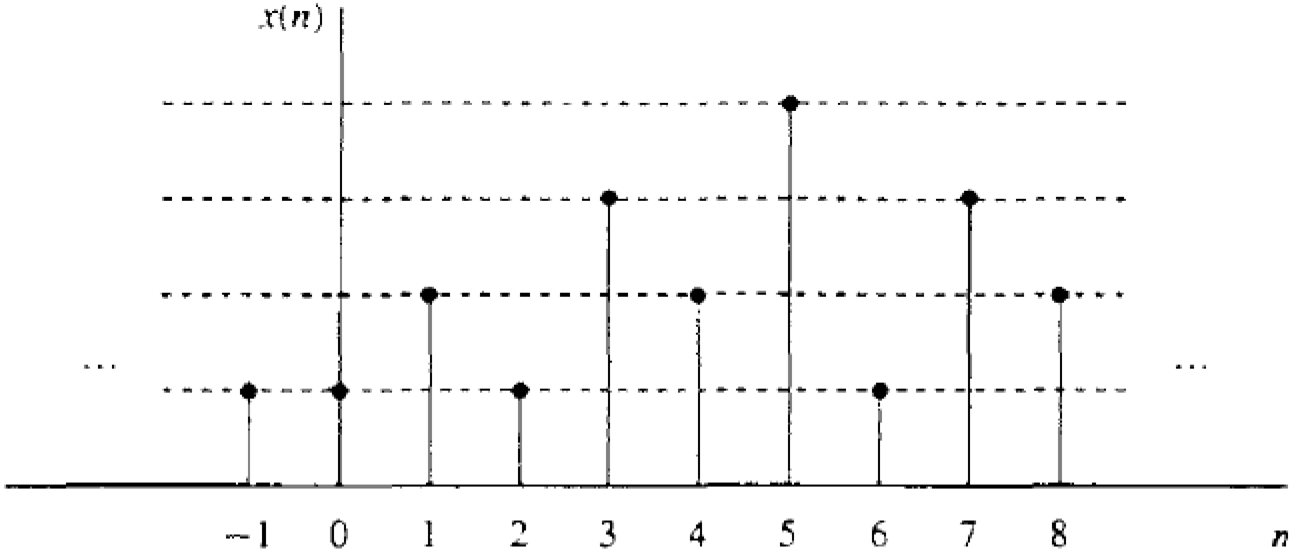


$$x(t) = 0.8^t,\ t \geq 0 \text{ and } x(t) = 0,\ t < 0$$

$$x(n) = \begin{cases} 0.8^n, & \text{if } n \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

Classification of Signals

- **Continuous-Valued and Discrete-Valued Signals**

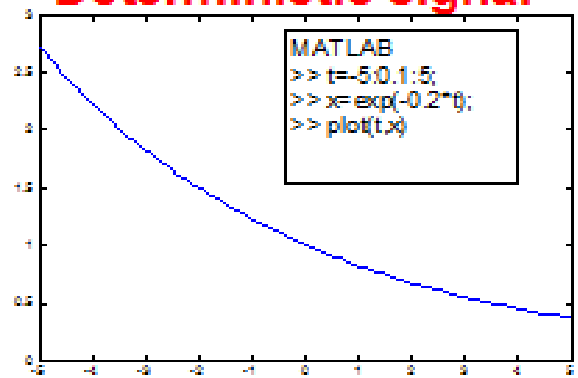


Classification of Signals

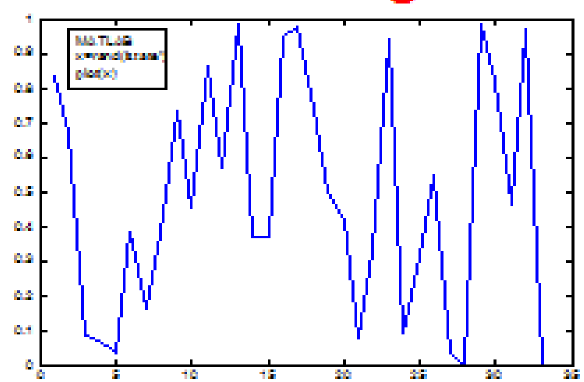
- **Deterministic and Random Signals**

  - Deterministic Vs Random
    - A **deterministic** signal is a signal in which each value of the signal is fixed and can be determined by a mathematical expression. The past, present and future of a deterministic signal are known with certainty. Because of this the future values of the signal can be calculated from past values with complete confidence.
      - Example: $x(t) = e^{-2t}$ is a deterministic signal.
    - A **random** or stochastic signal has a lot of uncertainty about its behavior. The future values of a random signal can't be accurately predicted. The random signal can be modeled using statistical information about the signal.
      - Examples: some common examples of random signals are speech and music.

Deterministic signal

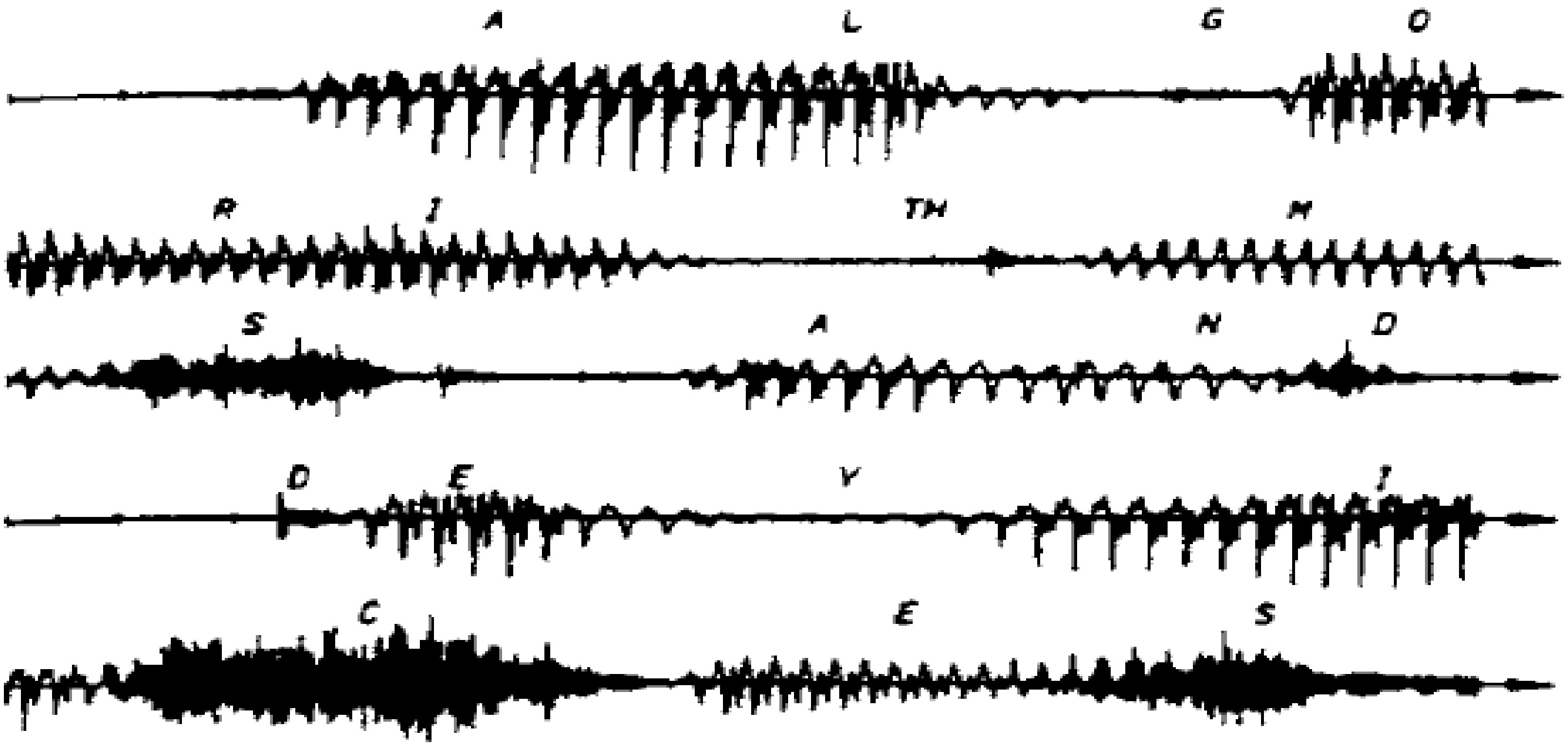


Random signal



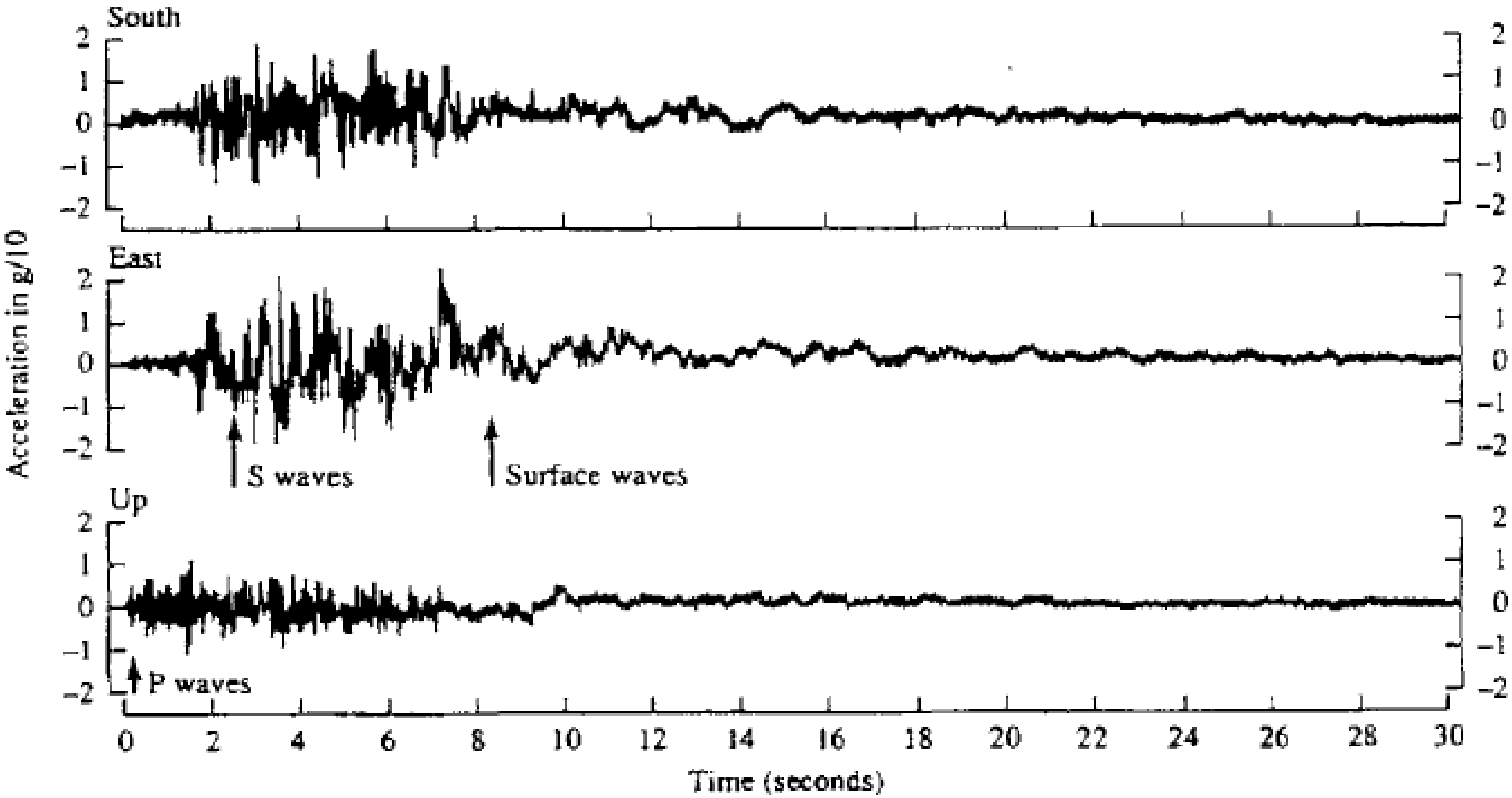
Classification of Signals

- **Deterministic and Random Signals**



Classification of Signals

- **Deterministic and Random Signals**



Classification of Signals

- **Peroidic and Aperoidic Signals**

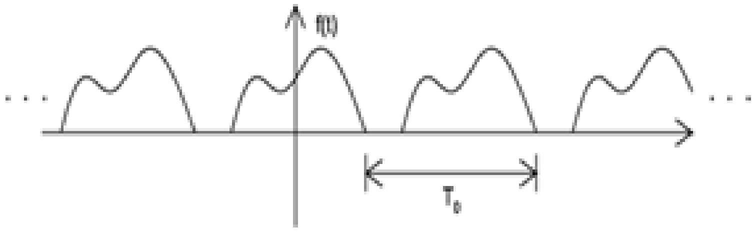
Periodic VS Aperiodic

- Periodic signals repeat with some period T, while aperiodic, or nonperiodic, signals do not. We can define a periodic function through the following mathematical expression, where t can be any number and T is a positive constant

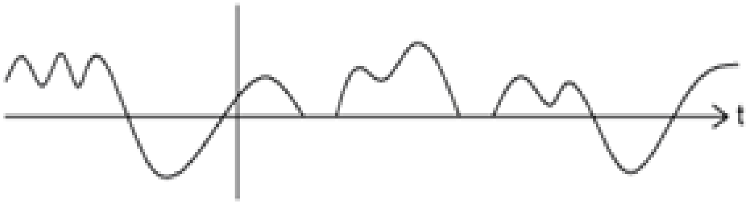
$$x(t) = x(t+T)$$

- The fundamental period of our function, x(t), is the smallest value of T that allows above Equation to be true.

Periodic signals



Aperiodic signals



Classification of Signals

- **Peroidic and Aperoidic Signals**

Defining Periodicity of a discrete-time signal:

- For any *continuous-time* sinusoidal function $x(t) = A\cos(\Omega_0 t + \theta)$ then it is always periodic with period $T = 2\pi / \Omega_0$.

Example 1: Show that x(t)=x(t+T)= $e^{j\Omega_0 t}$

Solution 1: x(t+T)$e^{j\Omega_0(t+T)} = e^{j\Omega_0 t}.e^{j\Omega_0 T} = e^{j\Omega_0 t}.e^{j\Omega_0(\frac{2\pi}{\Omega_0})} = e^{j\Omega_0 t}.e^{j2\pi}$

Recall that $e^{j2\pi} = \cos(2\pi) + j\sin(2\pi) = 1 + 0 = 1$

Hence, $= e^{j\Omega_0 t} = x(t)$ Proved.

- For a *discrete-time* sinusoid, it may or may not be periodic!
- *So how can we say if a discrete function is periodic or not????*

Classification of Signals

- **Peroidic and Aperoidic Signals**

  - To decide if a discrete function is periodic or not, lets assume, $x(n)=\cos(n\omega_0+\theta)$ is a periodic signal such that , $x(n)=x(n+N)$ then:

    $$\cos(n\omega_0+\theta)=\cos([n+N]\omega_0+\theta)=\cos(n\omega_0+N\omega_0+\theta)=\cos(n\omega_0+\theta+N\omega_0)$$

  - According to our assumption $x(n)$ is a periodic signal, therefore, $N\omega_0$ must be equal to the integer multiple of 2π, thus:

  - $N\omega_0=l2\pi$ where l is the integer > 0.

Therefore, $\omega_0=\dfrac{l}{N}2\pi$

  - So for $x(n)=\cos(n\omega_0+\theta)$ to be periodic, ω_0 must be a rational multiple of 2π
  - The periodicity of $x(n)$ is N, where $\omega_0=\dfrac{l}{N}2\pi$, and l and N are the smallest possible integers.

Continuous-time Sinusoids

To find the period $T > 0$ of a general continuous-time sinusoid $x(t) = A\cos(\omega t + \phi)$:

$$\begin{aligned} x(t) &= x(t+T) \\ A\cos(\omega t + \phi) &= A\cos(\omega(t+T) + \phi) \\ A\cos(\omega t + \phi + 2\pi k) &= A\cos(\omega t + \phi + \omega T) \\ \therefore \quad 2\pi k &= \omega T \\ T &= \frac{2\pi k}{\omega} \end{aligned}$$

where $k \in \mathbb{Z}$. Note: when k is the same sign as ω, $T > 0$.

Therefore, there exists a $T > 0$ such that $x(t) = x(t+T)$ and therefore $x(t)$ is $\boxed{\text{periodic}}$.

Periodicity

Recall if a signal $x(t)$ is periodic, then there exists a $T > 0$ such that

$$x(t) = x(t + T)$$

If no $T > 0$ can be found, then $x(t)$ is non-periodic.

Discrete-time Sinusoids

To find the integer period $N > 0$ (i.e., ($N \in \mathbb{Z}^+$) of a general discrete-time sinusoid $x[n] = A\cos(\Omega n + \phi)$:

$$\begin{aligned} x[n] &= x[n+N] \\ A\cos(\Omega n + \phi) &= A\cos(\Omega(n+N) + \phi) \\ A\cos(\Omega n + \phi + 2\pi k) &= A\cos(\Omega n + \phi + \Omega N) \\ \therefore \quad 2\pi k &= \Omega N \\ N &= \frac{2\pi k}{\Omega} \end{aligned}$$

where $k \in \mathbb{Z}$.

Note: there may not exist a $k \in \mathbb{Z}$ such that $\frac{2\pi k}{\Omega}$ is an integer.

Discrete-time Sinusoids

Example i: $\Omega = \frac{37}{11}\pi$

$$N = \frac{2\pi k}{\Omega} = \frac{2\pi k}{\frac{37}{11}\pi} = \frac{22}{37}k$$

$$N_0 = \frac{22}{37}k = \boxed{22} \text{ for } k = 37;\ x[n] \text{ is periodic.}$$

Example ii: $\Omega = 2$

$$N = \frac{2\pi k}{\Omega} = \frac{2\pi k}{2} = \pi k$$

$$N \in \mathbb{Z}^+ \quad \text{does not exist for any } k \in \mathbb{Z};\ x[n] \text{ is non-periodic.}$$

Example iii: $\Omega = \sqrt{2}\pi$

$$N = \frac{2\pi k}{\Omega} = \frac{2\pi k}{\sqrt{2}\pi} = \sqrt{2}k$$

$$N \in \mathbb{Z}^+ \quad \text{does not exist for any } k \in \mathbb{Z};\ x[n] \text{ is not periodic.}$$

Discrete-time Sinusoids

$$
\begin{aligned}
N &= \frac{2\pi k}{\Omega} \\
\Omega &= \frac{2\pi k}{N} = 2\pi \frac{k}{N} = \pi \cdot \underbrace{\boxed{\frac{2k}{N}}}_{RATIONAL}
\end{aligned}
$$

Therefore, a discrete-time sinusoid is periodic if its radian frequency Ω is a rational multiple of π.

Otherwise, the discrete-time sinusoid is non-periodic.

Example 1: $\Omega = \pi/6 = \pi \cdot \boxed{\frac{1}{6}}$

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

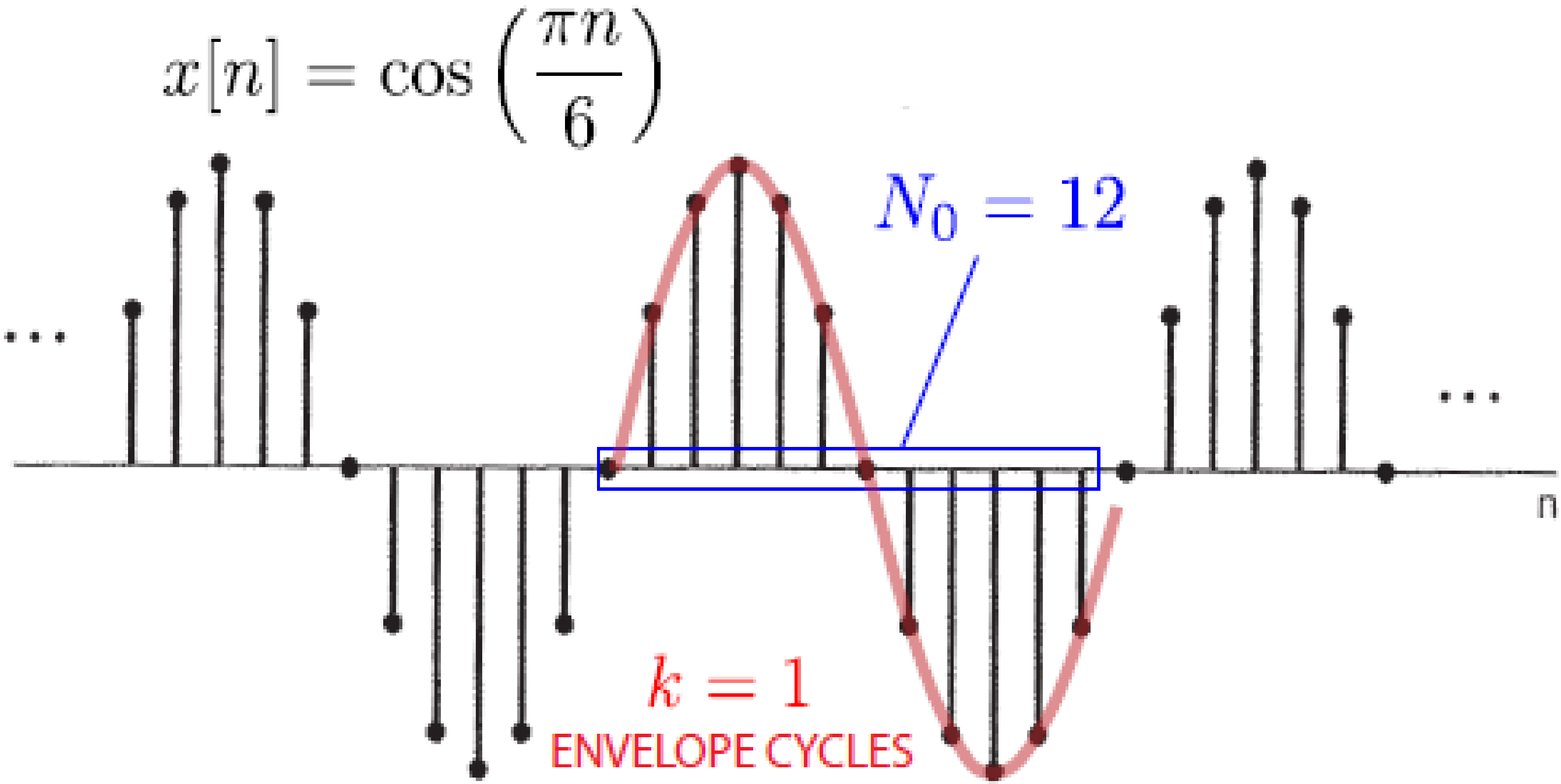
$$N = \frac{2\pi k}{\Omega} = \frac{2\pi k}{\pi \frac{1}{6}} = 12k$$

$$N_0 = 12 \quad \text{for } k = 1$$

The fundamental period is 12 which corresponds to $k = 1$ envelope cycles.

Classification of Signals

Discrete Time Sinusoids



Classification of Signals

Discrete Time Sinusoids

Example 2: $\Omega = 8\pi/31 = \pi \cdot \boxed{\frac{8}{31}}$

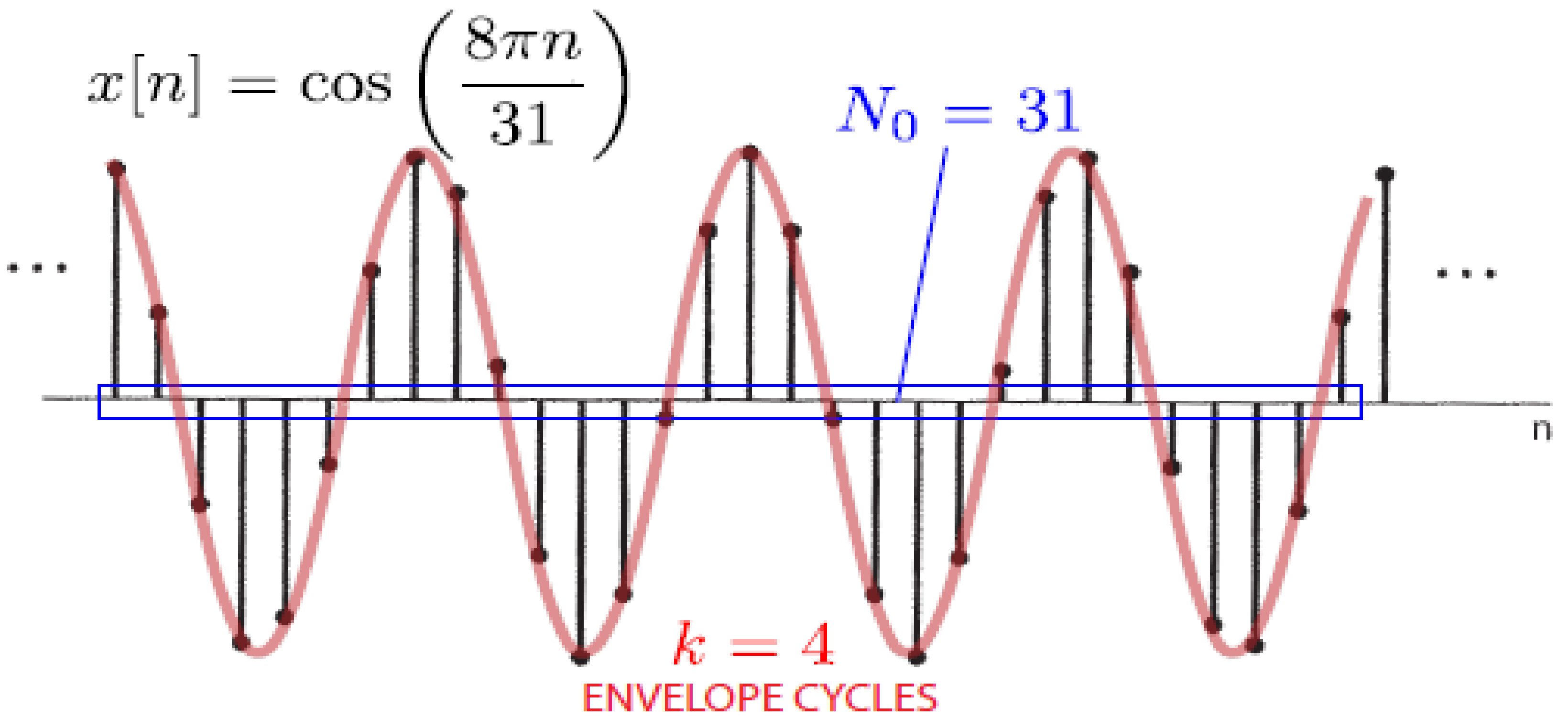
$$x[n] = \cos\left(\frac{8\pi n}{31}\right)$$

$$\begin{aligned} N &= \frac{2\pi k}{\Omega} = \frac{2\pi k}{\pi \frac{8}{31}} = \frac{31}{4} k \\ N_0 &= 31 \quad \text{for } k = 4 \end{aligned}$$

The fundamental period is 31 which corresponds to $k = 4$ envelope cycles.

Classification of Signals

Discrete Time Sinusoids



Example 3: $\Omega = 1/6 = \pi \cdot \boxed{\frac{1}{6\pi}}$

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

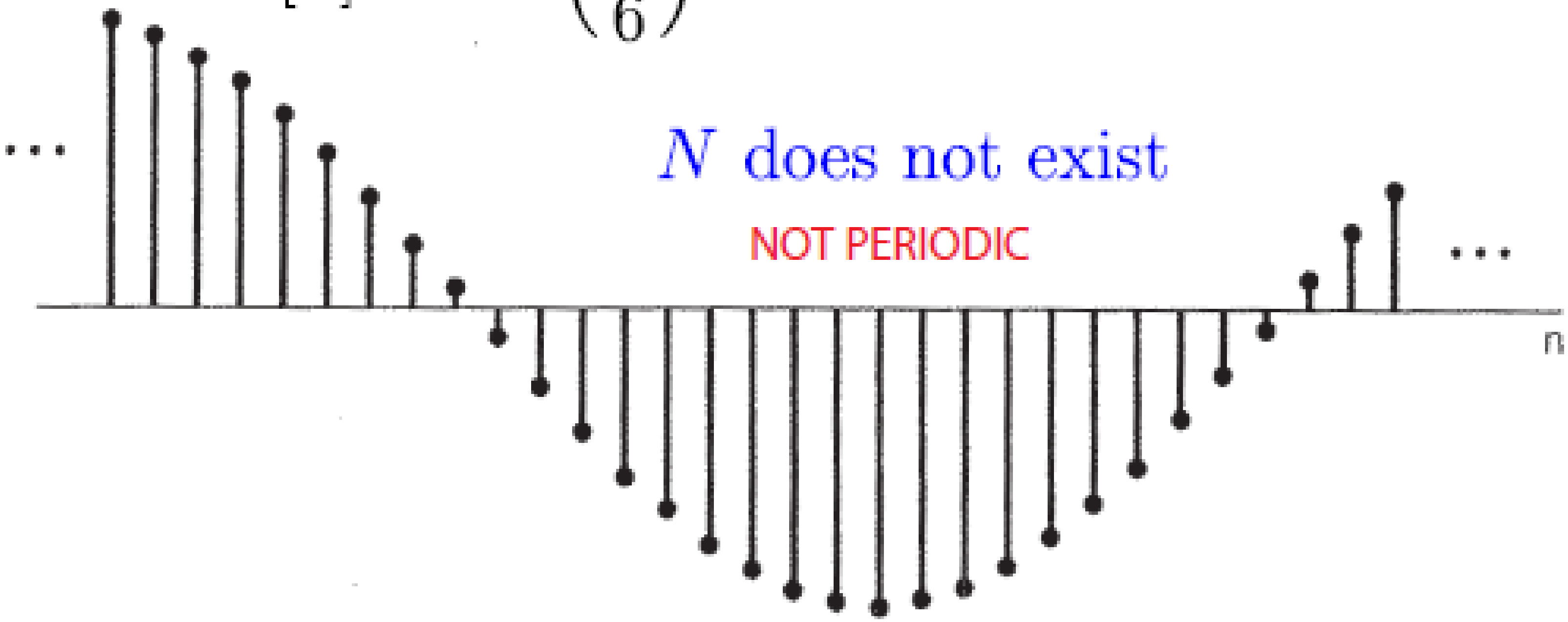
$$N = \frac{2\pi k}{\Omega} = \frac{2\pi k}{\frac{1}{6}} = 12\pi k$$

$N \in \mathbb{Z}^+$ does not exist for any $k \in \mathbb{Z}$; $x[n]$ is non-periodic.

Classification of Signals

Discrete Time Sinusoids

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



Continuous-Time Sinusoids: Frequency and Rate of Oscillation

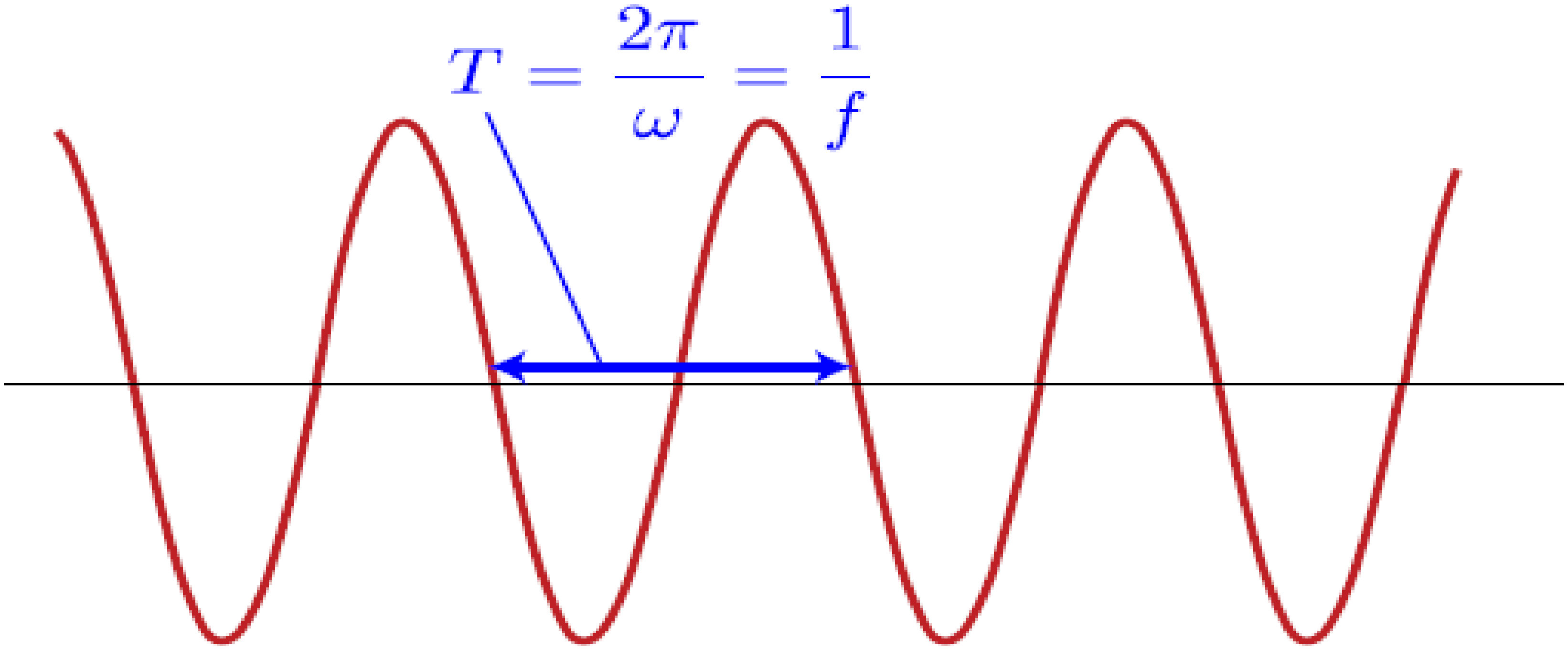
$$x(t) = A\cos(\omega t + \phi)$$

$$T = \frac{2\pi}{\omega} = \frac{1}{f}$$

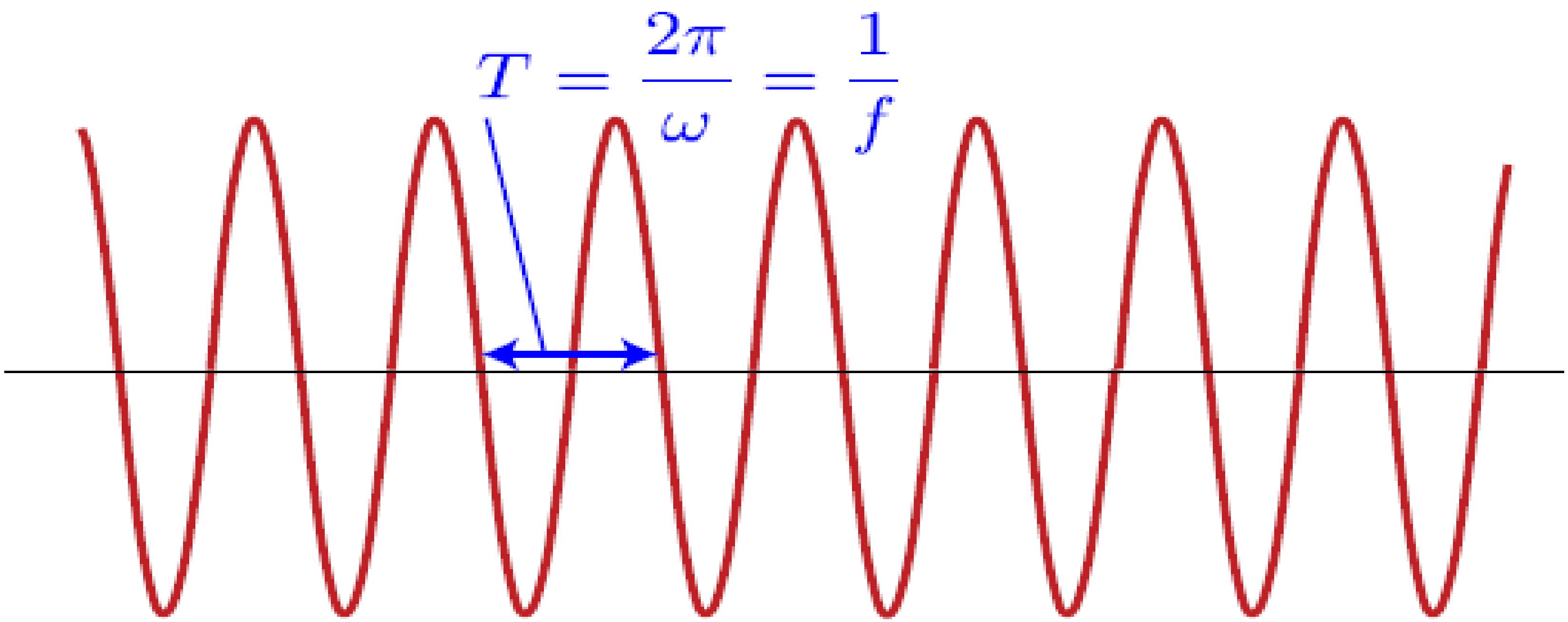
Rate of oscillation increases as ω increases (or T decreases).

Discrete Time Sinusoids

ω smaller



ω larger, rate of oscillation higher



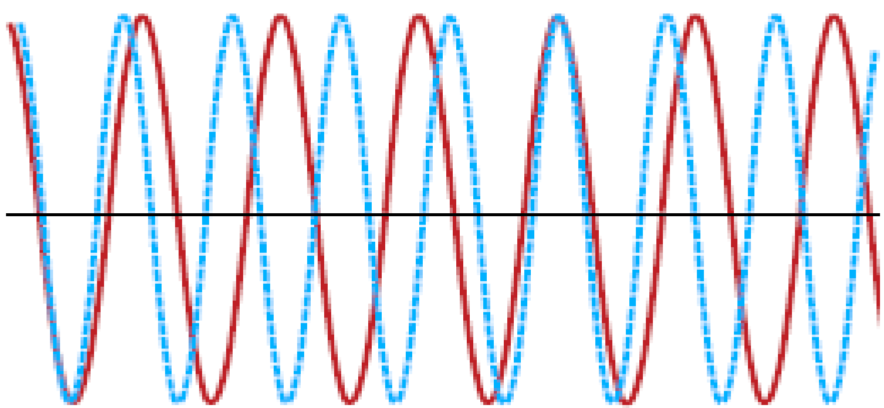
Discrete Time Sinusoids

Continuous-Time Sinusoids: Frequency and Rate of Oscillation

Also, note that $x_1(t) \neq x_2(t)$ for all t for

$$x_1(t) = A\cos(\omega_1 t + \phi) \quad \text{and} \quad x_2(t) = A\cos(\omega_2 t + \phi)$$

when $\omega_1 \neq \omega_2$.



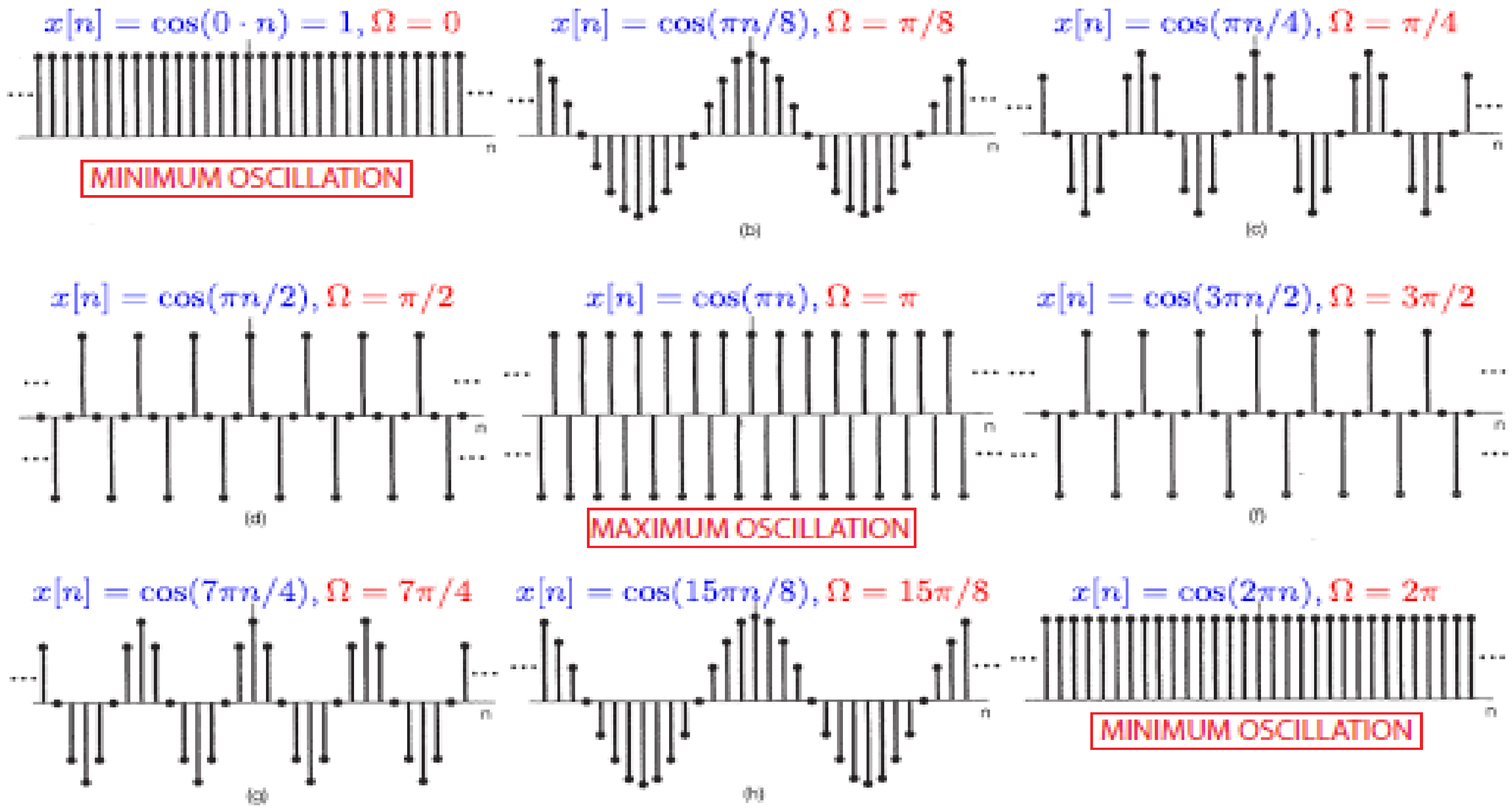
Discrete-Time Sinusoids: Frequency and Rate of Oscillation

$$x[n] = A\cos(\Omega n + \phi)$$

Rate of oscillation increases as Ω increases UP TO A POINT then decreases again and then increases again and then decreases again ....

Classification of Signals

Discrete Time Sinusoids



Discrete-Time Sinusoids: Frequency and Rate of Oscillation

$$x[n] = A\cos(\Omega n + \phi)$$

Discrete-time sinusoids repeat as Ω increases!

Discrete-Time Sinusoids: Frequency and Rate of Oscillation

Let

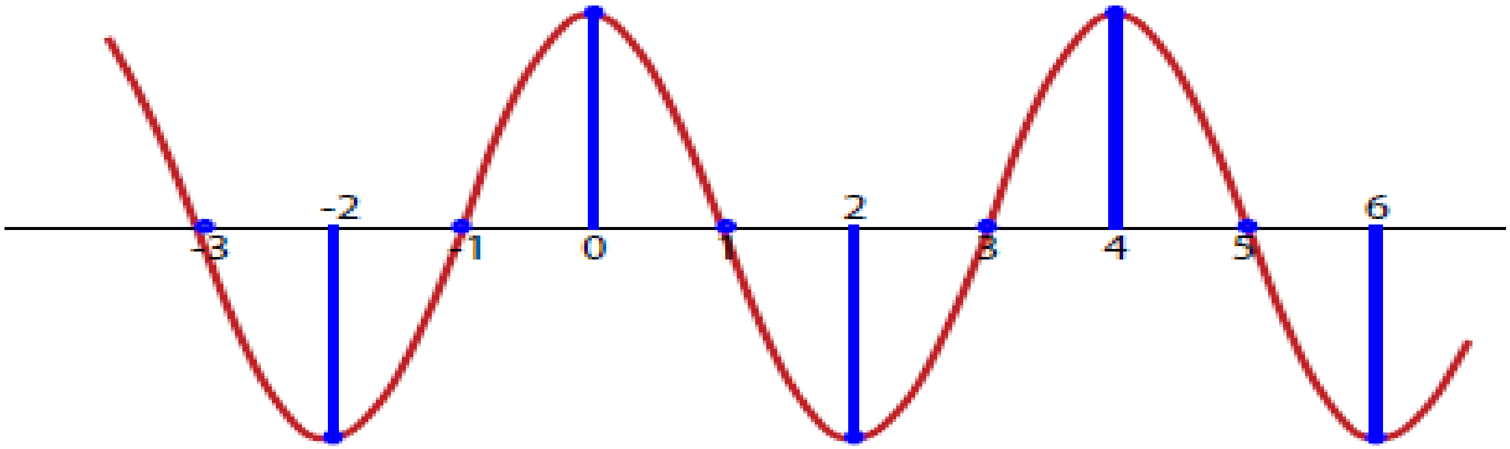
$$x_1[n] = A\cos(\Omega_1 n + \phi) \quad \text{and} \quad x_2[n] = A\cos(\Omega_2 n + \phi)$$

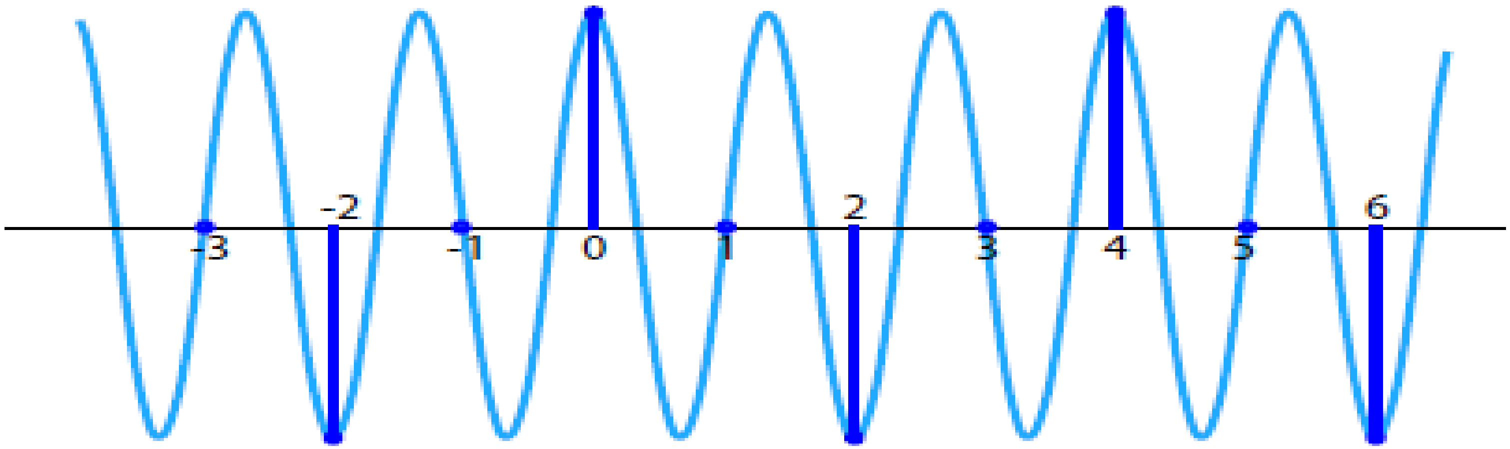
and $\Omega_2 = \Omega_1 + 2\pi k$ where $k \in \mathbb{Z}$:

$$\begin{aligned} x_2[n] &= A\cos(\Omega_2 n + \phi) \\ &= A\cos((\Omega_1 + 2\pi k)n + \phi) \\ &= A\cos(\Omega_1 n + 2\pi kn + \phi) \\ &= A\cos(\Omega_1 n + \phi) = x_1[n] \end{aligned}$$

Classification of Signals

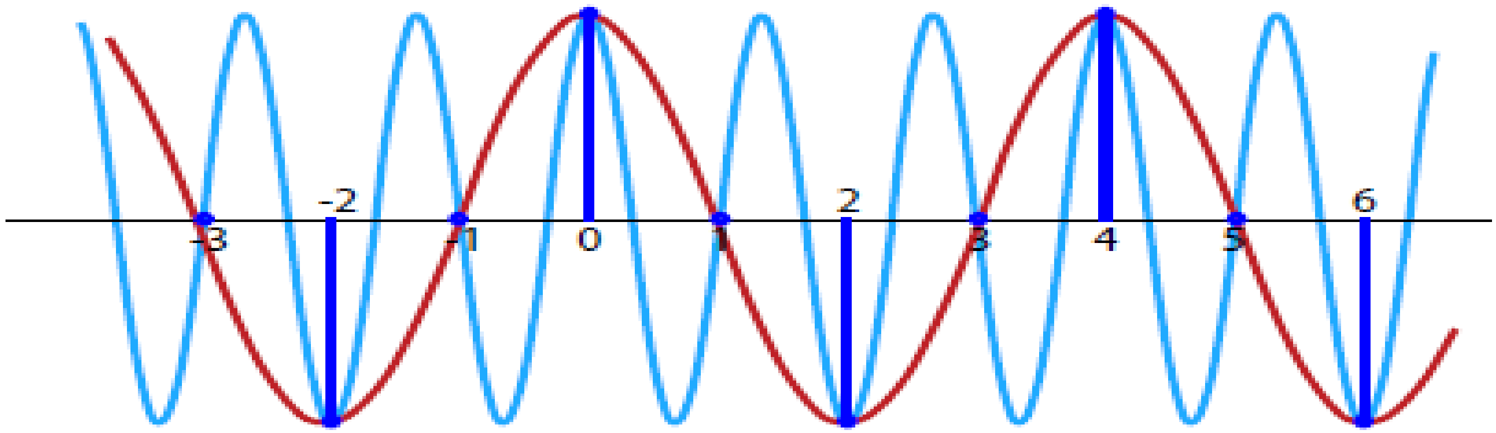
Discrete Time Sinusoids

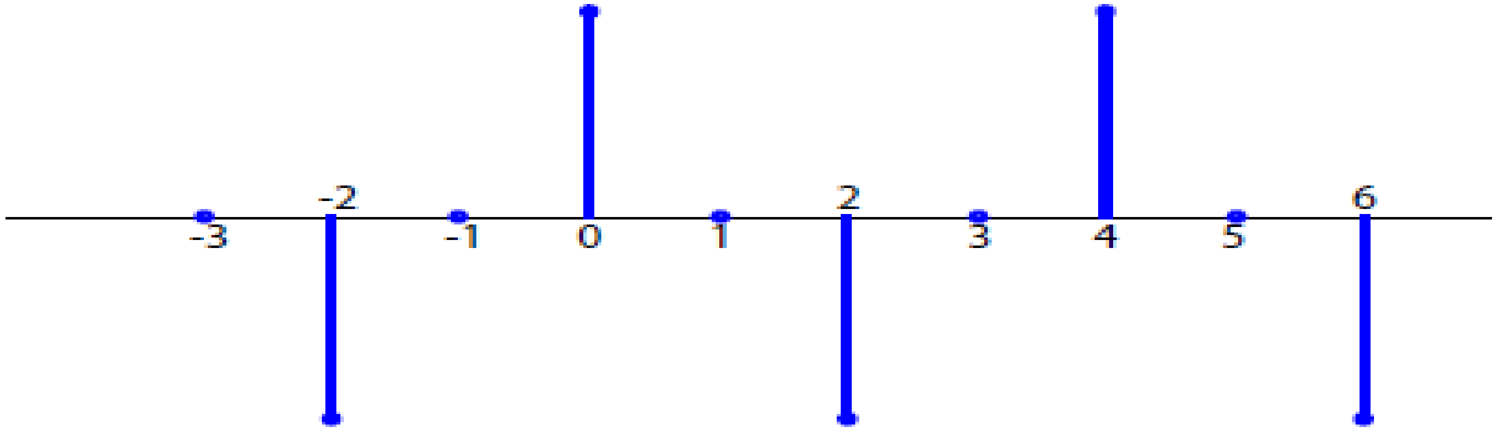




Classification of Signals

Discrete Time Sinusoids





Discete-Time Sinusoids: Frequency and Rate of Oscillation

$$x[n] = A\cos(\Omega n + \phi)$$

can be considered a sampled version of

$$x(t) = A\cos(\Omega t + \phi)$$

at integer time instants.

As Ω increases, the samples miss the faster oscillatory behavior.

- **Peroidic and Aperoidic Signals**

Example 2: Determine which of the sinusoids are periodic and compute their fundamental period.

(a) $\cos 0.01\pi n$

Solution 2:

$$\cos(0.01\pi n) = \cos\left(2\pi \times \frac{0.01}{2} n\right) = \cos\left(2\pi \frac{1}{200} n\right)$$

which means that the signal is periodic with f = 1/200 and fundamental period N = 200.

(b) $\cos(\pi 30n/105)$

Solution:

$$\cos\left(\pi \frac{30}{105} n\right) = \cos\left(2\pi \frac{30}{105 \times 2} n\right) = \cos\left(2\pi \frac{1}{7} n\right)$$

i.e. the signal is periodic with f = 1/7 and fundamental period = 7.

Classification of Signals

- **Peroidic and Aperoidic Signals**

Tutorials 1:

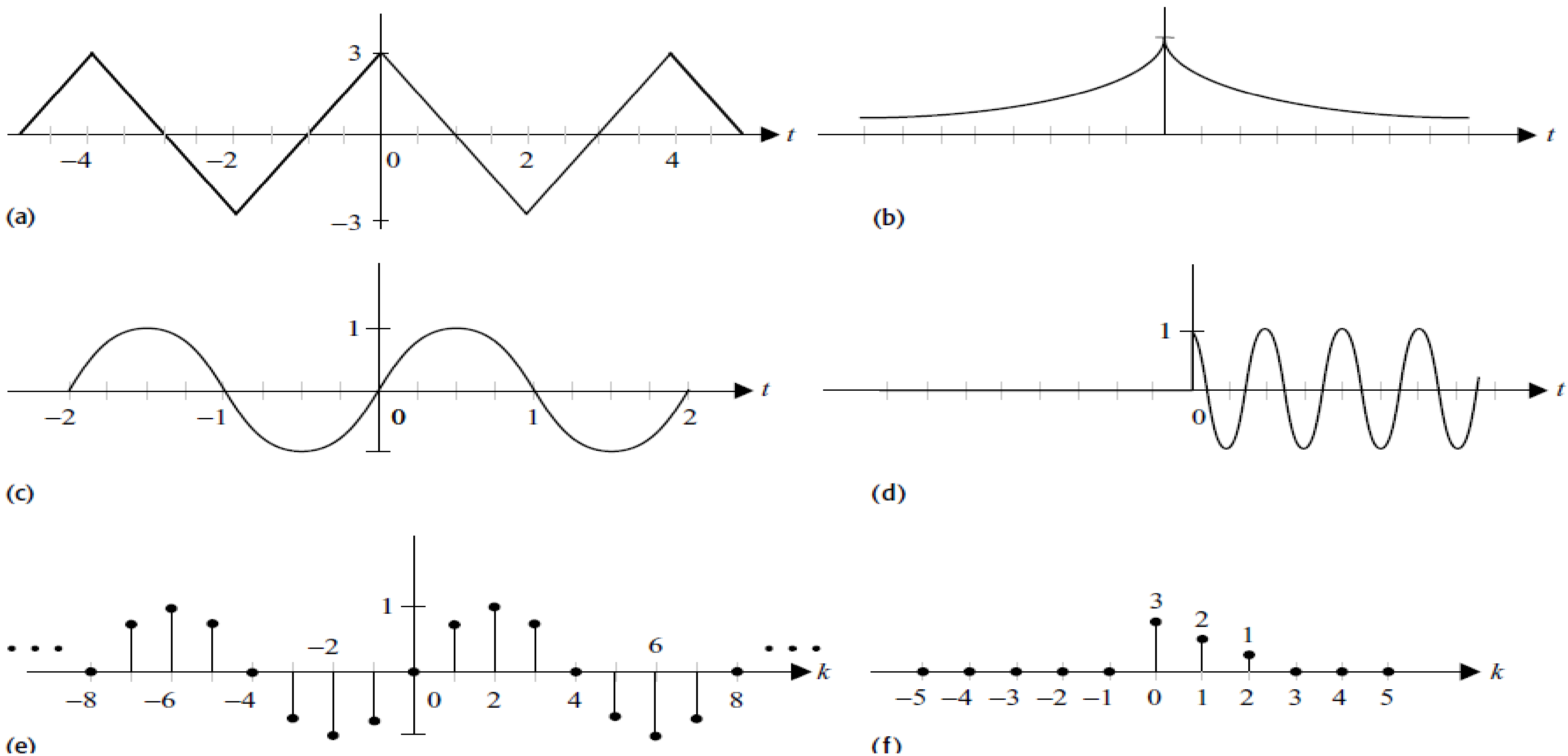
(a) $\cos(3n)$

(b) $3\cos(5n + \pi/6)$

(c) $x[n] = \cos(\pi n/2) - \sin(\pi n/8) + 3\cos(\pi n/4 + \pi/3)$

Classification of Signals

- **Peroidic and Aperoidic Signals**



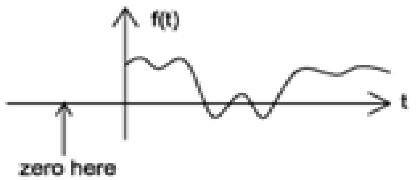
Classification of Signals

- **Causal Vs. Anticausal Vs. Noncausal**

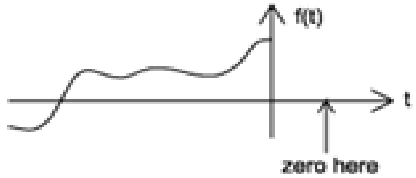
Causal Vs Anticausal Vs Noncausal

- **Causal signals** are signal that are zero for all negative time.
- **Anticausal** are signals that are zero for all positive time.
- **Noncausal** are signals that have nonzero values in both positive & negative time.

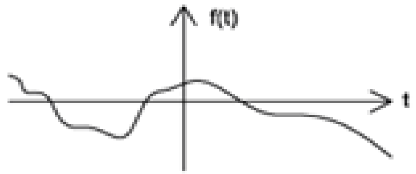
A causal signal



An anticausal signal



A noncausal signal



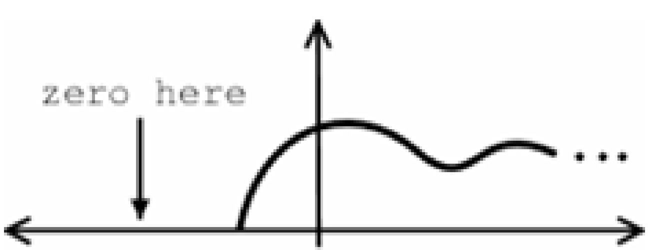
Classification of Signals

- **Right Handed Vs. Left Handed**

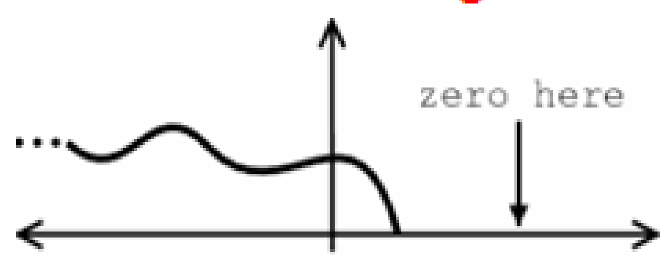
Right Handed Vs Left Handed

- **Right handed** signal is defined as any signal where $x(n) = 0$ for $n<N<\infty$.
- **Left handed** signal is defined as any signal where $x(n) = 0$ for $n>N>\infty$.

Right-handed signal



Left-handed signal

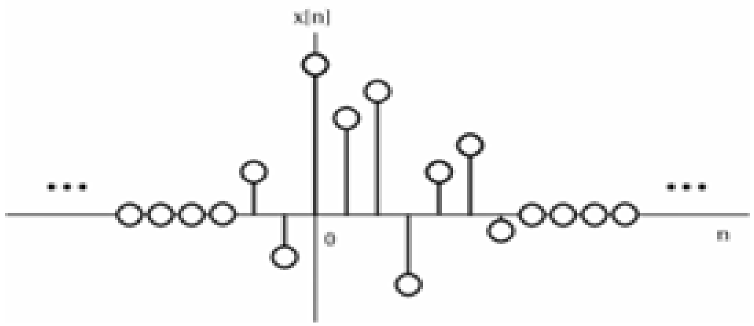


Classification of Signals

- **Finite Vs. Infinite Length**

Finite Vs Infinite Length

- Signals can be characterized as to whether they have finite or infinite length set of values.
- Most finite length signals are used when dealing with discrete time signals or a given sequence of length.
- Mathematically speaking, x(t) is a finite length signal if it is nonzero over a finite interval $t_1 < x(t) < t_2$ where $t_1 > -\infty$ & $t_2 < \infty$
- Infinite length signal, x(t), is defined as non zero over all real numbers.



Classification of Signals

- **Even Vs. Odd**

Even Vs Odd

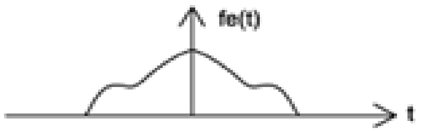
- An **even** or symmetric signal (discrete or continuous) is any signal such that x (-t) = x(t) or x [-n] = x[n]
  - Even signals can be easily spotted as they are symmetric around the vertical axis.
- An **Odd** signal (discrete or continuous), on the other hand, is a signal such that x (-t)= -x(t) or x [-n]= -x[n]
- An odd signal is anti-symmetric!
- Any signal can be written as:

$$x(n)= x_e(n) + x_o(n)$$

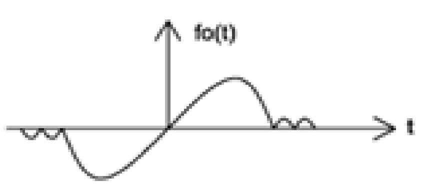
Where,

$$\begin{cases} x_e(n) = \frac{1}{2}\left[x(n) + x(-n)\right] \\ x_o(n) = \frac{1}{2}\left[x(n) - x(-n)\right] \end{cases}$$

Even signal



Odd signal

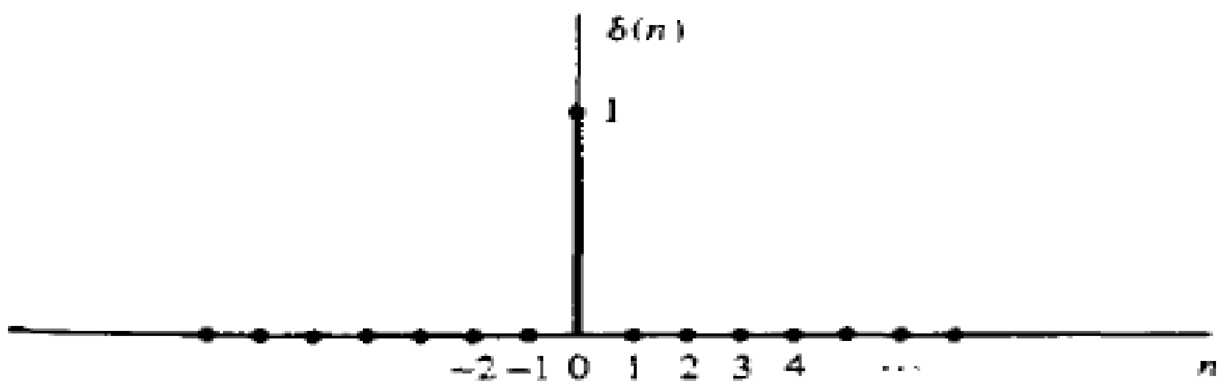


Basic Digital Sequences (Signals)

Unit impulse (unit sample)

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

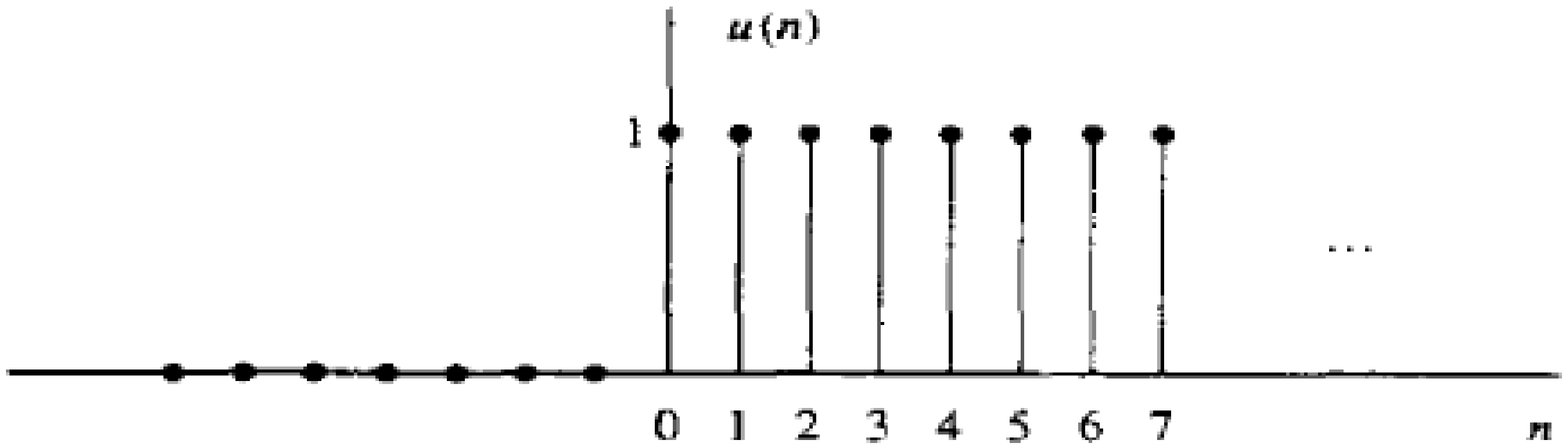
$$\begin{aligned} u(n) &= \sum_{m=-\infty}^{n} \delta(m) \quad \text{summing}, \\ \delta(n) &= u(n) - u(n-1) \quad \text{differencing}. \end{aligned}$$



Basic Digital Sequences (Signals)

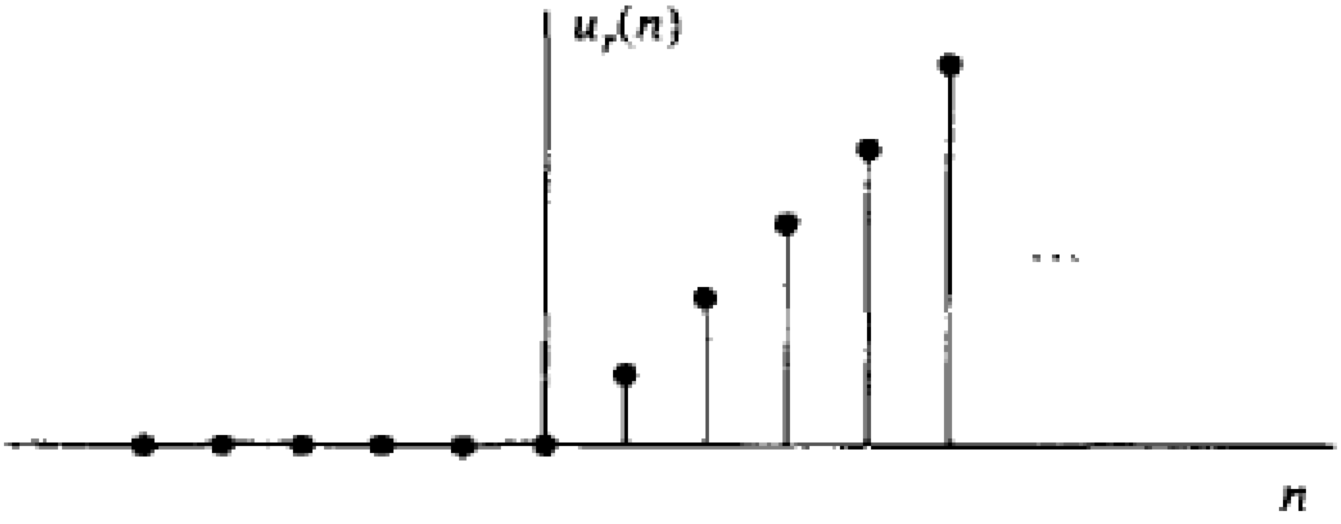
Unit step signal

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



Unit Ramp Signal

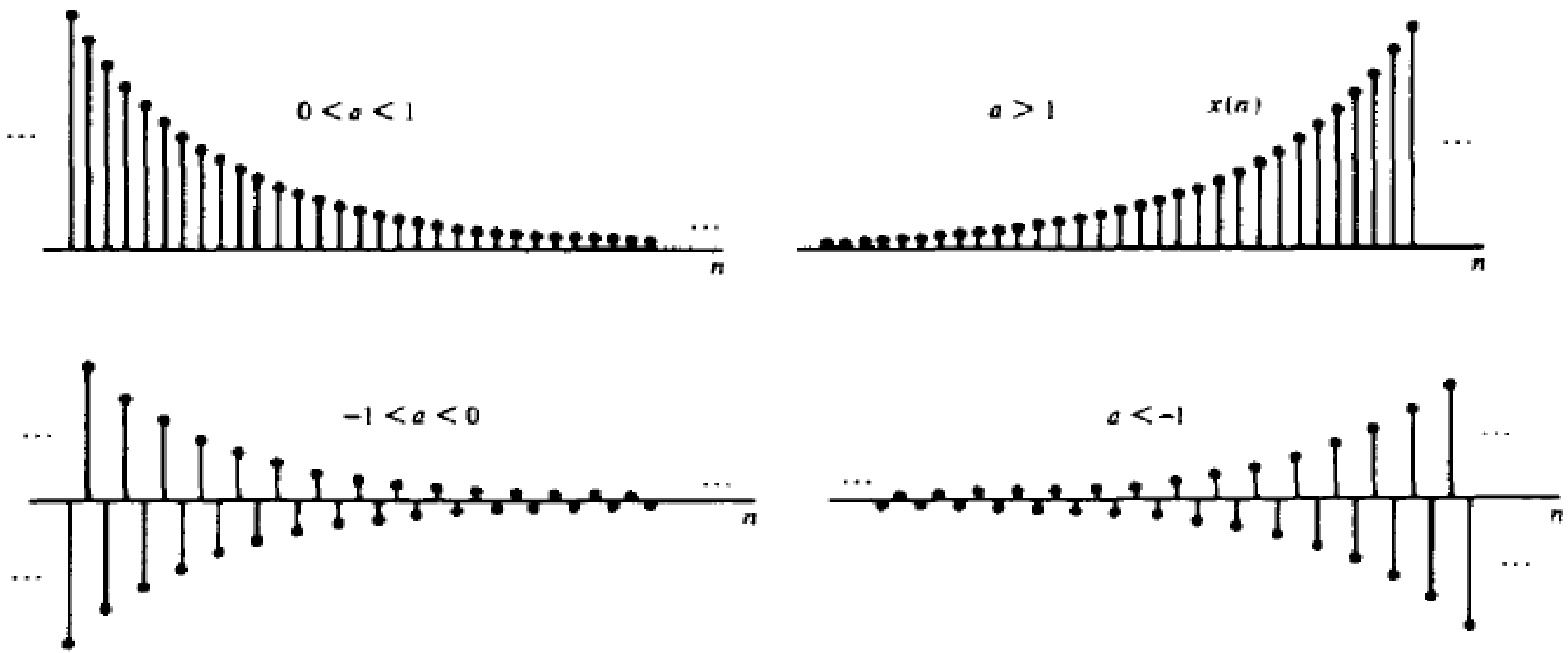
$$u_r(n) = \begin{cases} n, & \text{for } n \geq 0 \\ 0, & \text{for } n < 0 \end{cases}$$



Exponential Signal

$$x(n) = a^n \qquad \text{for all } n$$

If the parameter a is real, then $x(n)$ is a real signal.



Exponential Signal

$$x(n) = a^n \qquad \text{for all } n$$

When the parameter a is complex valued, it can be expressed as

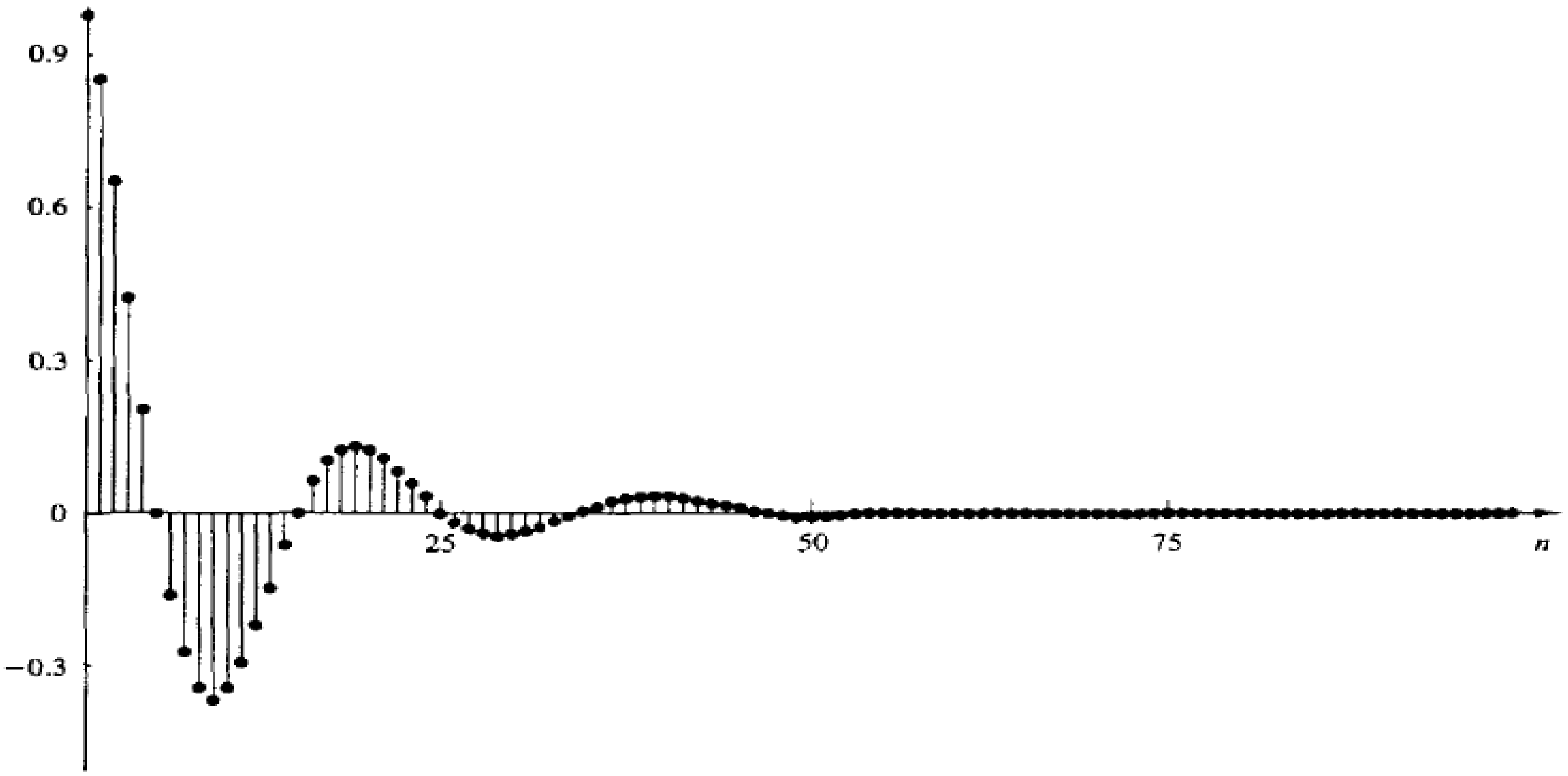
$$a \equiv re^{j\theta}$$

where r and θ are now the parameters. Hence we can express $x(n)$ as

$$\begin{aligned} x(n) &= r^n e^{j\theta n} \\ &= r^n(\cos\theta n + j\sin\theta n) \end{aligned}$$

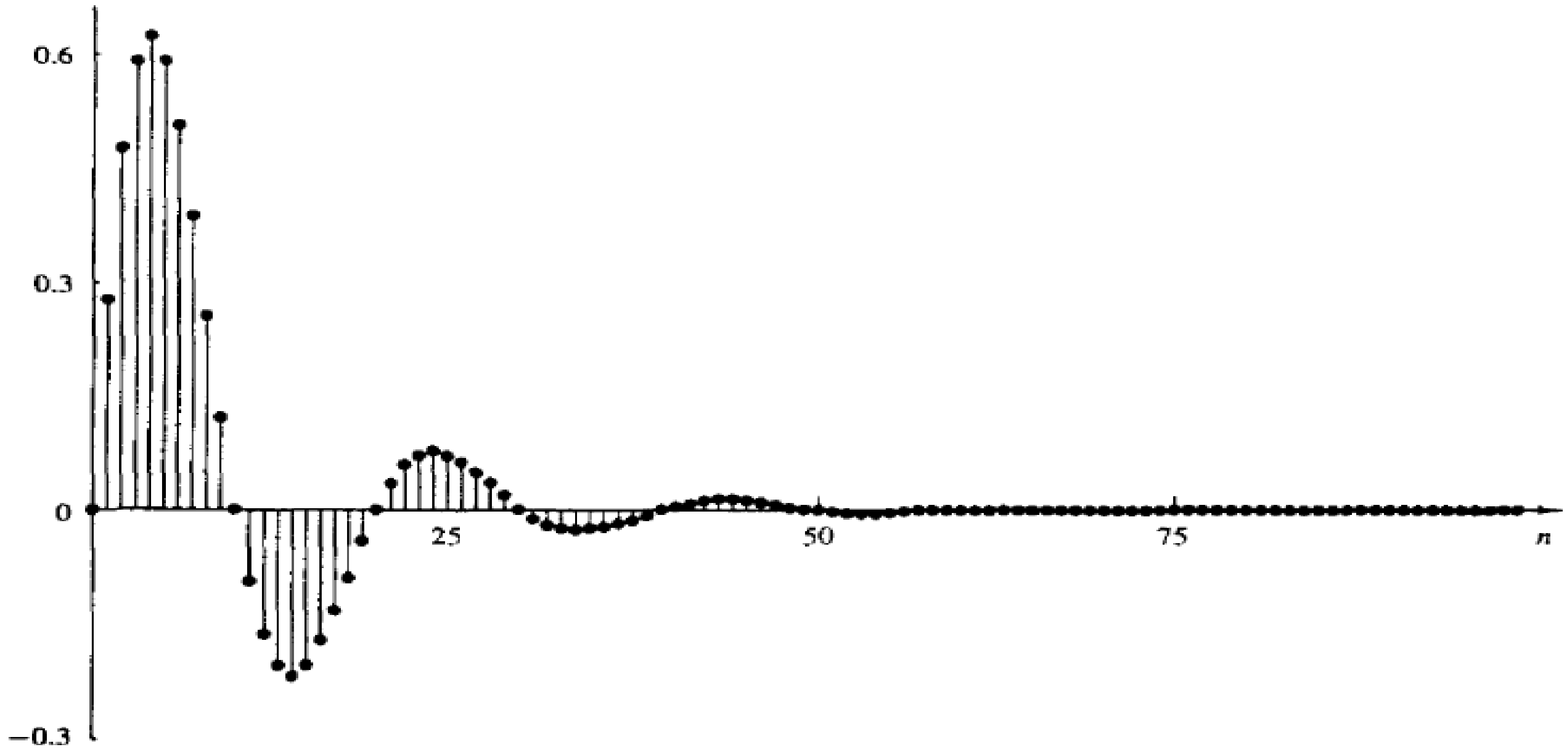
Basic Digital Sequences (Signals)

Exponential Signal



(a) Graph of $x_R(n) = (0.9)^n \cos \frac{\pi n}{10}$

Exponential Signal



(b) Graph of $x_f(n) = (0.9)^n \sin \frac{\pi n}{10}$

Basic Digital Sequences (Signals)

Sinusoids Signal

Sinusoids

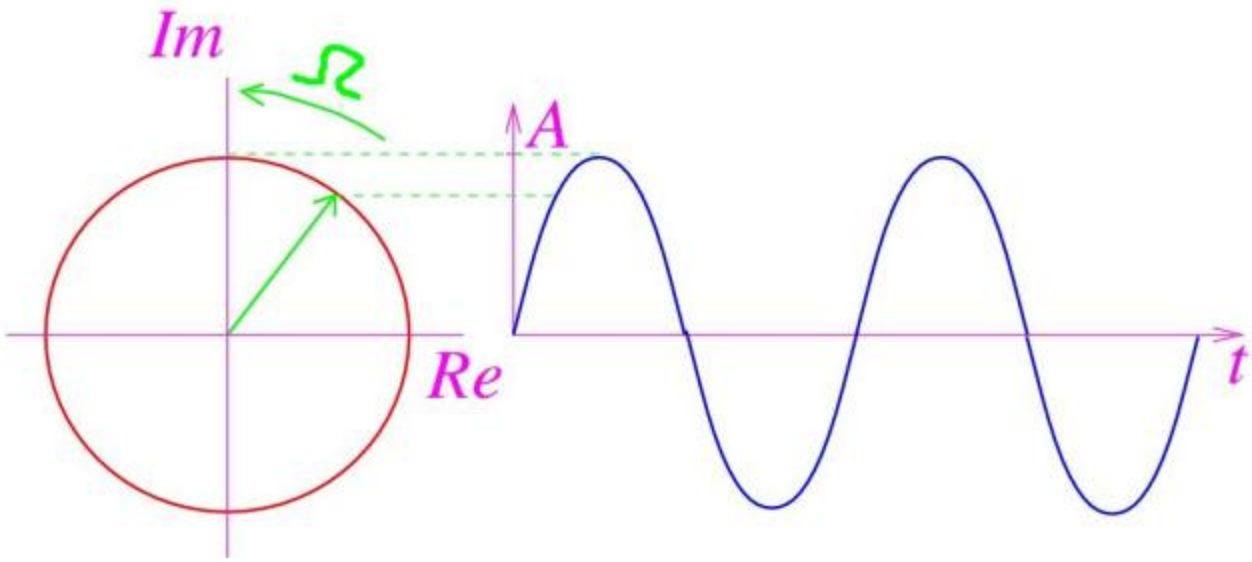
$$x(n) = A\sin(\omega n + \theta)$$

Useful properties:

$$\begin{aligned}
\exp[j(\omega n + \theta)] &= \cos(\omega n + \theta) + j\sin(\omega n + \theta), \\
\cos(\omega n + \theta) &= \frac{\exp[j(\omega n + \theta)] + \exp[-j(\omega n + \theta)]}{2}, \\
\sin(\omega n + \theta) &= \frac{\exp[j(\omega n + \theta)] - \exp[-j(\omega n + \theta)]}{2j}.
\end{aligned}$$

Sinusoids Signal

A sine wave as the projection of a complex phasor onto the imaginary axis:



Classification of DSP Systems

Linear Vs. Non-linear Systems

A **linear** system is any system that obeys the properties of scaling (homogeneity) and superposition (additivity), while a **nonlinear** system is any system that does not obey at least one of these.

To show that a system H obeys the scaling property is to show that

$$H(kf(t)) = kH(f(t))$$

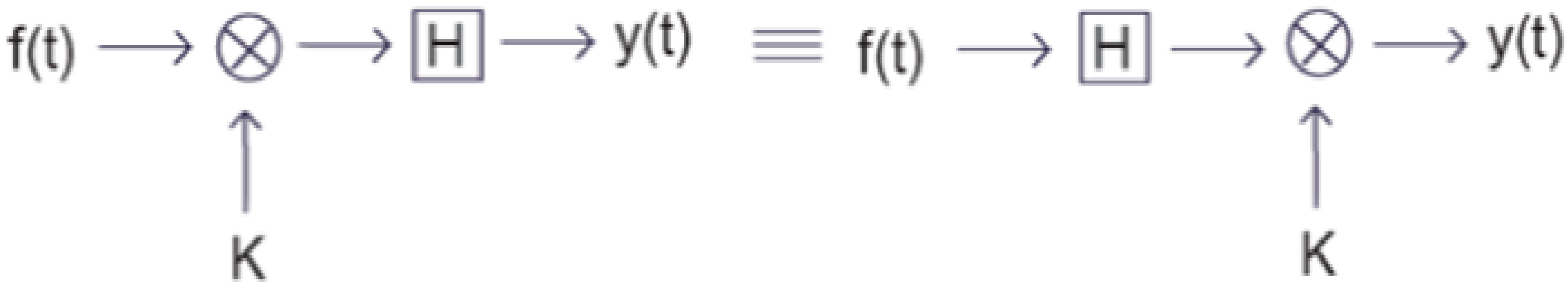
To demonstrate that a system H obeys the superposition property of linearity is to show that

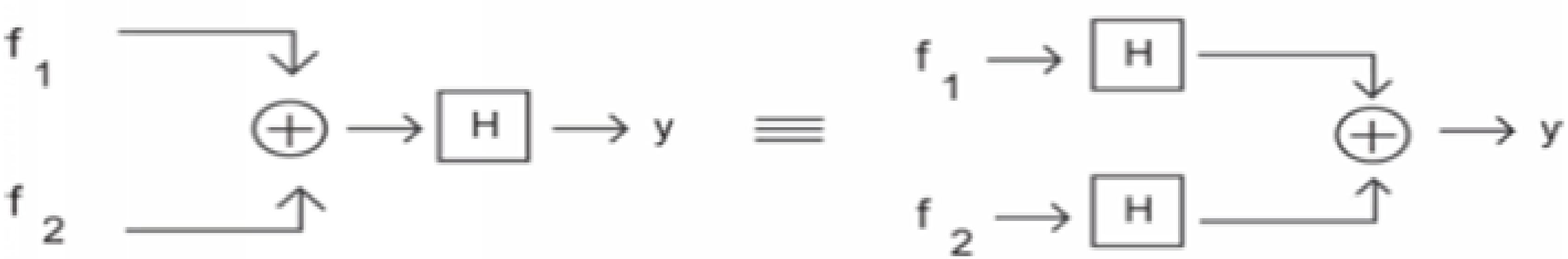
$$H(f_1(t) + f_2(t)) = H(f_1(t)) + H(f_2(t))$$

It is possible to check a system for linearity in a single (though larger) step. To do this, simply combine the first two steps to get

$$H(k_1 f_1(t) + k_2 f_2(t)) = k_2 H(f_1(t)) + k_2 H(f_2(t))$$

Linear vs. Non-linear Systems





Classification of DSP Systems

Linear vs. Non-linear Systems

- Linear System: A system is linear if and only if

$$T\{x_1[n] + x_2[n]\} = T\{x_1[n]\} + T\{x_2[n]\} \quad \text{(additivity)}$$

and

$$T\{ax[n]\} = aT\{x[n]\} \quad \text{(scaling)}$$

- Examples
  - Ideal Delay System

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

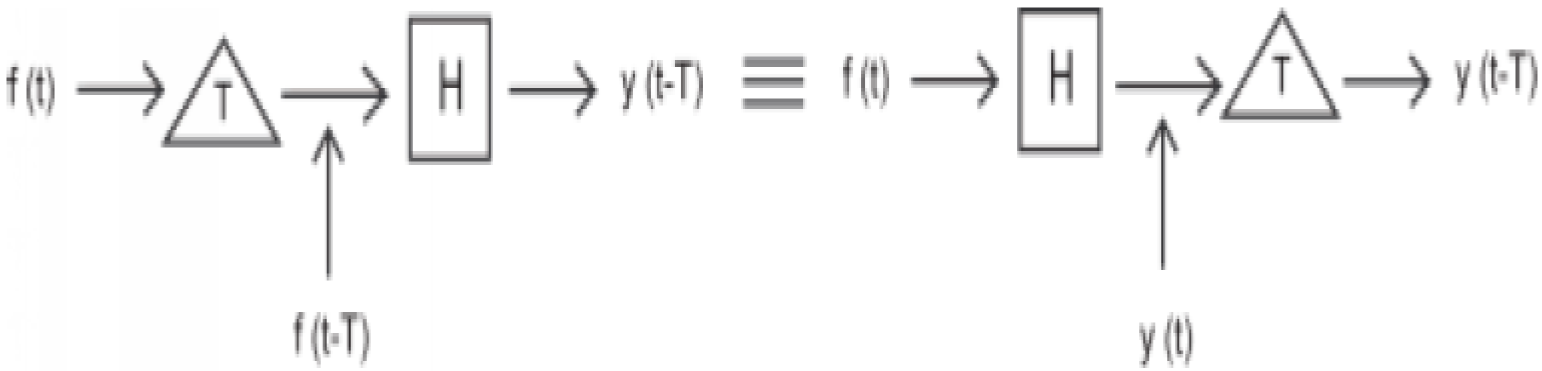
$$\begin{aligned} T\{x_1[n] + x_2[n]\} &= x_1[n - n_o] + x_2[n - n_o] \\ T\{x_2[n]\} + T\{x_1[n]\} &= x_1[n - n_o] + x_2[n - n_o] \\ T\{ax[n]\} &= ax_1[n - n_o] \\ aT\{x[n]\} &= ax_1[n - n_o] \end{aligned}$$

Classification of DSP Systems

Time Invariant vs. Time Variant

A **time invariant** system is one that does not depend on when it occurs: the shape of the output does not change with a delay of the input. That is to say that for a system H where $H(f(t)) = y(t)$, H is time invariant if for all T

$$H(f(t-T)) = y(t-T)$$



When this property does not hold for a system, then it is said to be **time variant** , or time-varying.

Classification of DSP Systems

Time Invariant vs. Time Variant

- Time-Invariant (shift-invariant) Systems
  - A time shift at the input causes corresponding time-shift at output

$$y[n] = T\{x[n]\} \Rightarrow y[n - n_o] = T\{x[n - n_o]\}$$

- Example
  - Square

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

Delay the input the output is $y_1[n] = (x[n - n_o])^2$

Delay the output gives $y[n - n_o] = (x[n - n_o])^2$

- Counter Example
  - Compressor System

$y[n] = x[Mn]$

Delay the input the output is $y_1[n] = x[Mn - n_o]$

Delay the output gives $y[n - n_o] = x[M(n - n_o)]$

Casual vs. Noncasual

A **causal** system is one that is **nonanticipative** ; that is, the output may depend on current and past inputs, but not future inputs. All "realtime" systems must be causal, since they can not have future inputs available to them.

One may think the idea of future inputs does not seem to make much physical sense; however, we have only been dealing with time as our dependent variable so far, which is not always the case. Imagine rather that we wanted to do image processing. Then the dependent variable might represent pixels to the left and right (the "future") of the current position on the image, and we would have a **noncausal** system.

Causality

- A system is causal it's output is a function of only the current and previous samples

Examples

- Backward Difference

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

Counter Example

- Forward Difference

$$y[n] = x[n+1] + x[n]$$

Stable vs. Nonstable

A **stable** system is one where the output does not diverge as long as the input does not diverge. A bounded input produces a bounded output. It is from this property that this type of system is referred to as **bounded input-bounded output (BIBO)** stable.

Representing this in a mathematical way, a stable system must have the following property, where $x(t)$ is the input and $y(t)$ is the output. The output must satisfy the condition

$$|y(t)| \leq M_y < \infty$$

when we have an input to the system that can be described as

$$|x(t)| \leq M_x < \infty$$

M_x and M_y both represent a set of finite positive numbers and these relationships hold for all of t.

If these conditions are not met, i.e. a system's output grows without limit (diverges) from a bounded input, then the system is **unstable** .

Classification of DSP Systems

Stable vs. Nonstable

Stability (in the sense of bounded-input bounded-output BIBO)

- A system is stable if and only if every bounded input produces a bounded output

$$|x[n]| \le B_x < \infty \Rightarrow |y[n]| \le B_y < \infty$$

Example

- Square

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

if input is bounded by $|x[n]| \le B_x < \infty$

output is bounded by $|y[n]| \le B_x^2 < \infty$

Counter Example

- Log

$$y[n] = \log_{10}(|x[n]|)$$

even if input is bounded by $|x[n]| \le B_x < \infty$

output not bounded for $x[n] = 0 \Rightarrow y[0] = \log_{10}(|x[n]|) = -\infty$

Classification of DSP Systems

Memoryless System

- Memoryless System
  - A system is memoryless if the output y[n] at every value of n depends only on the input x[n] at the same value of n

- Example Memoryless Systems
  - Square

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

  - Sign

$$y[n] = \text{sign}\{x[n]\}$$

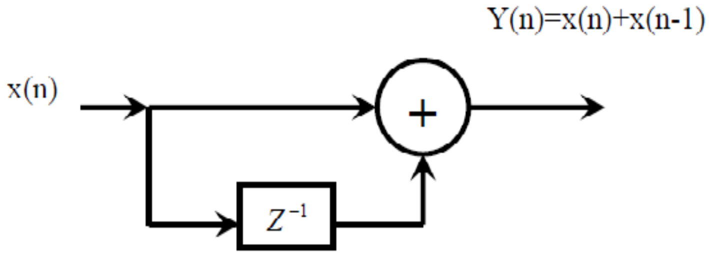
- Counter Example
  - Ideal Delay System

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

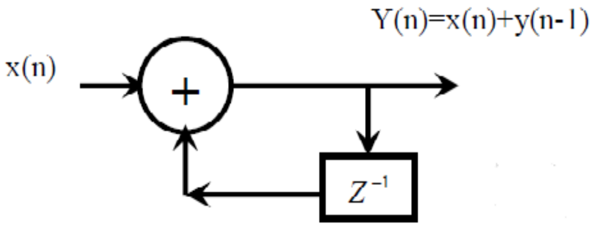
Characterization of Digital Filters

Recursive and Nonrecursive Digital Filters

A recursive system is one in which the output y(n) is dependent on one or more of its past outputs (y(n-1), y(n-2)G) while a non recursive system is one in which the output is independent of any past outputs .e.g. feedforward system having no feedback is a non recursive system.



Non Recursive System



Recursive System

Digital Signal Processing

Course Instructor
Lecturer: WARQAA SHAHER

Lecture No. 3: Discrete Time Systems

Third Class
Department of Computer and Software Engineering

http://www.engineering.uodiyala.edu.iq/
https://www.facebook.com/Engineer.College1

Lecture Outline

- **Classification of Discrete Time Systems (DTS)**
- **Basic Operations on Signals**
- **Describing Digital Signals with Impluse Function**

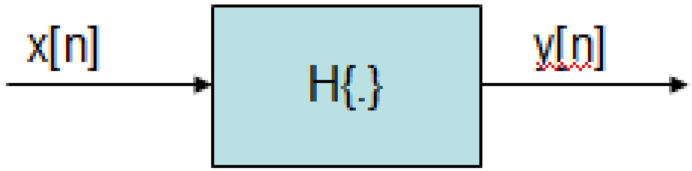
Discrete Time Systems (DTS)

- A discrete time system is a device or algorithm that operates on a discrete time signal x[n], called the <u>input</u> or <u>excitation</u>, according to some well defined rule, to produce another discrete time signal y[n] called the <u>output</u> or <u>response</u> of the system.
- We express the general relationship between x[n] and y[n] as

$$y[n] = H\{x[n]\}$$

where the symbol H denotes the transformation (also called an *operator*), or processing performed by the system on x[n] to produce y[n].

Discrete-time System



Static Versus Dynamic:

- Static System = memory less = the output doesn't depend on the past future values of the input.
- Dynamic system = having either finite or infinite memory.

Example 1:

$$y[n] = x^2[n]$$ Static or memory-less System

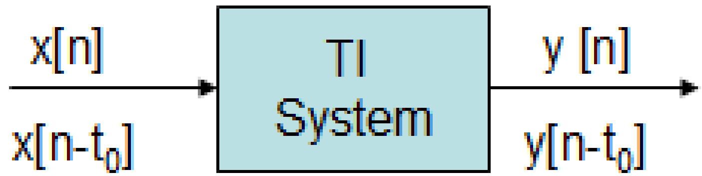
$$y[n] = \sum_{k=0}^{N} x[n-k]$$ Dynamic-finite

$$y[n] = \sum_{k=0}^{\infty} x[n-k]$$ Dynamic-infinite

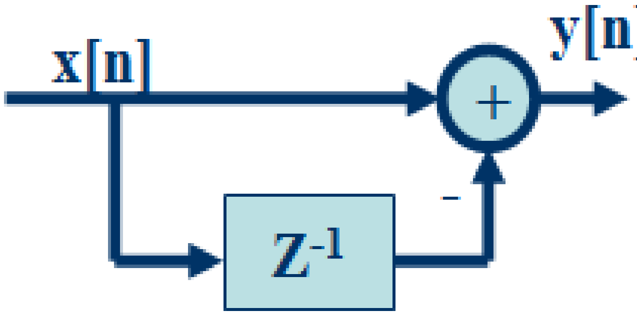
Classification of Discrete Time Systems (DTS)

Time Invariant versus Time Variant Systems:

- A system is time invariant if
  - When the input is shifted in time, then its output is shifted by the same amount
  - This must hold for all possible shifts.
  - Stated in another way, *a system is called time invariant if its input-output characteristics do not change with time*. Otherwise the system is said to be time variant.
- If a shift in input $x[n]$ by t_0 causes a shift in output $y[n]$ by t_0 for all real-valued t_0, then system is time-invariant:



Example 2: Determine if the system shown in the figure is time invariant or time variant.



Solution 2: $y[n] = x[n] - x[n-1]$

Now if the input is delayed by k units in time and applied to the system, the output is

$y[n,k] = n[n-k] - x[n-k-1]$ (1)

On the other hand, if we delay y[n] by k units in time, we obtain

$y[n-k] = x[n-k] - x[n-k-1]$ (2)

(1) and (2) show that the system is **time invariant**.

Examples

Example 3: Determine if the following systems are time invariant or time variant.

(a) $y[n] = nx[n]$ (b) $y[n] = x[n]\cos w_0 n$

Solution 3:

(a) The response to this system to $x[n-k]$ is

$$y[n,k] = nx[n-k] \qquad (3)$$

Now if we delay $y[n]$ by k units in time, we obtain

$$y[n-k] = (n-k)x[n-k]$$
$$= nx[n-k] - kx[n-k] \qquad (4)$$

which is different from (3). This means the system is time-variant.

(b) The response of this system to $x[n-k]$ is

$$y[n,k] = x[n-k]\cos w_0 n \qquad (5)$$

If we delay the output $y[n]$ by k units in time, then

$$y[n-k] = x[n-k]\cos w_0[n-k]$$

which is different from that given in (5), hence the system is time variant.

Classification of Discrete Time Systems (DTS)

Tutorials:

Q4: Determine whether the following systems are time invariant or time variant.

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

(b) $y[n] - (y[n-2])/n = 2x[n]$

Linearity

- A system is linear if it is both
  - *Homogeneous*: If we scale the input, then the output is scaled by the same amount:

$$f\left(a\,x(t)\right) = a\,f\left(x(t)\right)$$

  - *Additive*: If we add two input signals, then the output will be the sum of their respective outputs

$$f\left(x_1(t) + x_2(t)\right) = f\left(x_1(t)\right) + f\left(x_2(t)\right)$$

- A system that is both linear and time invariant is called <u>Linear Time-Invariant</u> (LTI) system.

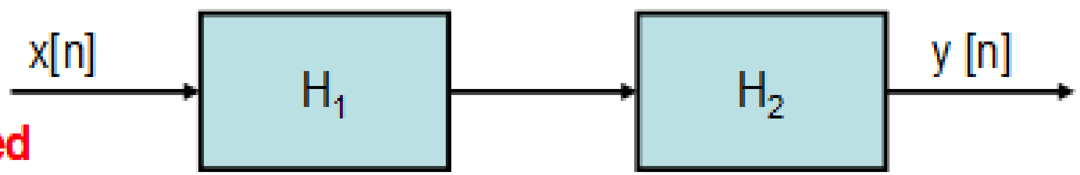
Classification of Discrete Time Systems (DTS)

Linear versus Non-linear Systems:

A system H is linear if and only if

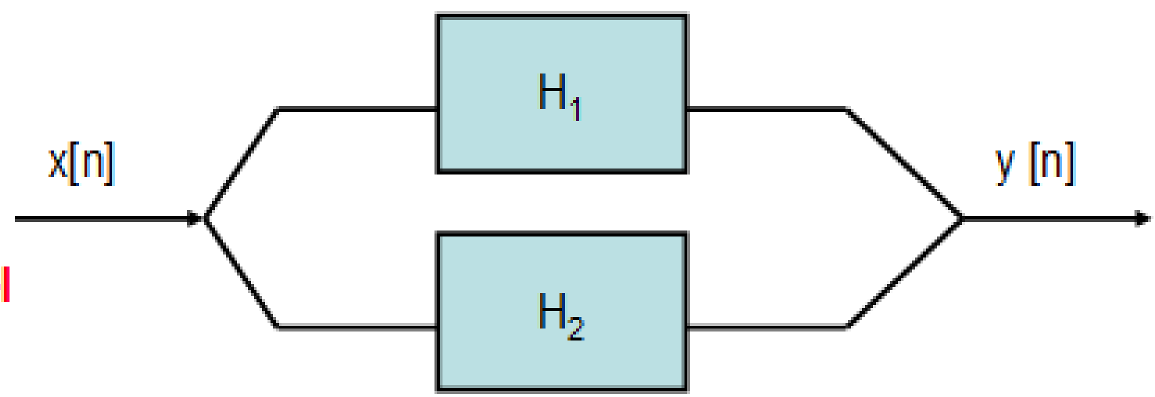
$$H[a_1x_1[n] + a_2x_2[n]] = a_1H[x_1[n]] + a_2H[x_2[n]]$$

for any arbitrary input sequences $x_1[n]$ and $x_2[n]$, and any arbitrary constants a_1 and a_2.



Cascaded

$H=H_1 \cdot H_2=H_2 \cdot H_1$ for Linear systems only!



Parallel

$H=H_1+H_2$

for Linear systems only!

Examples

Example 4: Determine if the following systems are linear or nonlinear.

(a) $y[n] = n\,x[n]$ (b) $y[n] = A\,x[n] + B$

Solution 4:

(a) $y[n] = n\,x[n]$

For two input sequences $x_1[n]$ and $x_2[n]$, the corresponding outputs are

$$y_1[n] = nx_1[n] \quad \text{and } y_2[n] = nx_2[n]$$

A linear combination of the two input sequences results in the output

$$H[a_1x_1[n] + a_2x_2[n]] = n[a_1x_1[n] + a_2x_2[n]] = na_1x_1[n] + na_2x_2[n] \quad (1)$$

On the other hand, a linear combination of the two outputs results in the out

$$a_1y1[n] + a_2y_2[n] = a_1nx_1[n] + a_2nx_2[n] \qquad (2)$$

Since the right hand sides of (1) and (2) are identical, the system is linear.

Examples

(b) $y[n] = A\,x[n] + B$

Assuming that the system is excited by $x_1[n]$ and $x_2[n]$ separately, we obtain the corresponding outputs

$$y_1[n] = Ax_1[n] + B \quad \text{and } y_2 = Ax_2[n] + B$$

A linear combination of $x_1[n]$ and $x_2[n]$ produces the output

$$\begin{aligned} y_3[n] &= H[a_1x_1[n] + a_2x_2[n]] = A[a_1x_1[n] + a_2x_2[n]] + B \\ &= Aa_1x_1[n] + Aa_2x_2[n] + B \qquad (3) \end{aligned}$$

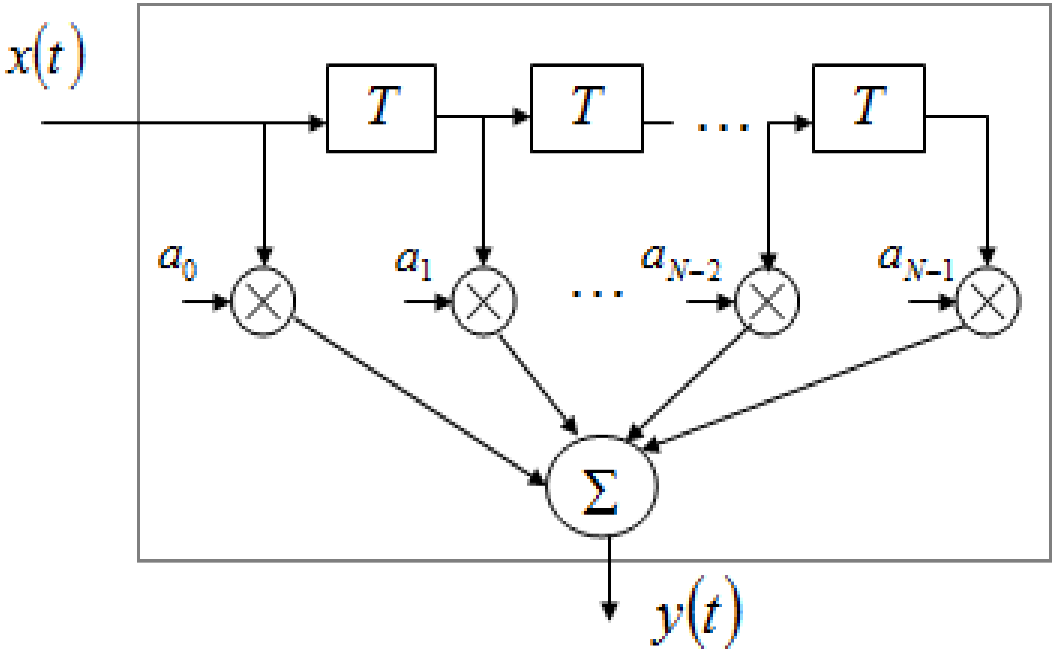
On the other hand, if the system were linear, its output to the linear combination of $x_1[n]$ and $x_2[n]$ would be a linear combination of $y_1[n]$ and $y_2[n]$, that is,

$$a_1y_1[n] + a_2y_2[n] = a_1Ax_1[n] + a_1B + a_2Ax_2[n] + a_2B \qquad (4)$$

Clearly, (3) and (4) are different and hence the system is nonlinear. Under what conditions would it be linear?

Examples

- Tapped delay line



Each T represents a delay of T time units

There are N-1 delays

$$y(t) = a_0\, x(t) + a_1\, x(t-T) + \cdots + a_{N-1}\, x(t-(N-1)T) = \sum_{k=0}^{N-1} a_k\, x(t-kT)$$

Tutorial

Tutorial

Q5: Determine whether following systems are linear or non linear.

(a) Squarer $$y(t) = x^2(t)$$

(b) Differentiation $$y(t) = \frac{d}{dt}x(t)$$

$$x(t) \longrightarrow \boxed{\frac{d}{dt}(\bullet)} \longrightarrow y(t)$$

(c) Integration $$y(t) = \int_{-\infty}^{t} x(u)\,du$$

$$x(t) \longrightarrow \boxed{\int_{-\infty}^{t}(\bullet)\,dt} \longrightarrow y(t)$$

Causal versus Noncausal Systems

- A system is said to be causal if the output of the system at any time n [i.e. y[n]) depends only on present and past inputs [i-e x[n], x[n-1],....]but does not depend on future inputs [i.e. x[n+1], x[n+2]...]. If the system does not satisfy this definition, it is called noncausal.

Example: Determine if the systems described by the following input-output equations are causal or noncausal.

(a) $y[n] = x[n] - x[n-1]$ (b) $y[n] = ax[n]$

(c) $y[n] = x[n] + 3x[n+4]$ (d) $y[n] = x[n^2]$

(e) $y[n] = x[-n]$

Solution:

The systems (a), (b) are causal, all others are non-causal. $y[n]=x[-n]$ is non-causal because $y(-1)=x(1)$! Thus the o/p at n=-1 depends on the i/p at n=1, which is two units of time into the future.

Future value!

Stable versus unstable Systems:

- A system is stable if any bounded input produces bounded output (BIBO).
- Otherwise, it is unstable!
- The condition that the i/p sequence x(n) & the o/p sequence y(n) are bounded is translated mathematically to mean that there exist some finite numbers.

Say M_x & M_y, such that

$|x(n)| <= M_x < inf \quad |y(n)| <= M_y < inf$ for all n.

If, for some reason bounded i/p sequence x(n), the o/p is unbounded (infinite), system is unstable.

Classification of Discrete Time Systems (DTS)

Invertibility:

- A system is said to be *invertible* if the input to the system may be uniquely determined from the output.
  - In order for a system to be invertible, it is necessary for distinct inputs to produce distinct outputs.
  - In other words, given any two inputs $x_1(n)$ and $x_2(n)$ with $x_1(n) \neq x_2(n)$, it must be true that $y_1(n) \neq y_2(n)$.
- This property is important in applications such as channel equalization and deconvolution is *invertibility*.

Example: The system defined by

$$y(n) = x(n)g(n)$$

Solution: is invertible if and only if $g(n) \neq 0$ for all n. In particular, given y(n) with g(n) nonzero for all n, x(n) may be recovered from y(n) as follows:

$$x(n) = \frac{y(n)}{g(n)}$$

Summary

- If several causes are acting on a linear system, then the total effect is the sum of the responses from each cause
- In time-invariant systems, system parameters do not change with time
- For memoryless systems, the system response at any instant *n* depends only on the present value of the input (value at *n*)
- If a system response at n depends on future input values (beyond n), then the system is noncausal.

Operations performed on dependant variables:

- **Amplitude scaling**

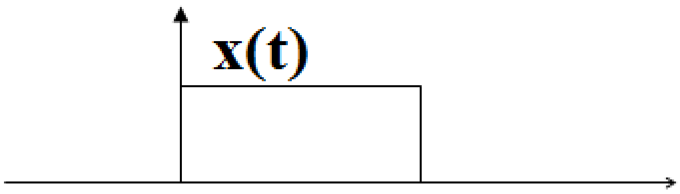
Let x(t) denote a continuous time signal. The signal y(t) resulting from amplitude scaling applied to x(t) is defined by

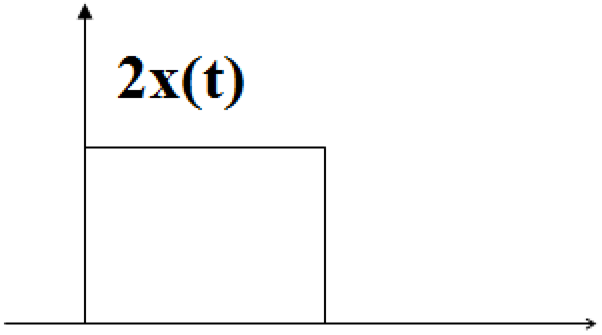
$$y(t) = c\,x(t)$$

where c is the scale factor.

In a similar manner to the above equation, for discrete time signals we can write

$$y[nT] = c\,x[nT]$$





Basic Operations on Signals

Addition

Let $x_1[n]$ and $x_2[n]$ denote a pair of discrete time signals. The signal $y[n]$ obtained by the addition of $x_1[n] + x_2[n]$ is defined as

$$y[n] = x_1[n] + x_2[n]$$

Example: Audio mixer

Multiplication

Let $x_1[n]$ and $x_2[n]$ denote a pair of discrete-time signals. The signal $y[n]$ resulting from the multiplication of the $x_1[n]$ and $x_2[n]$ is defined by

$$y[n] = x_1[n].x_2[n]$$

Example: AM Radio Signal

Basic Operations on Signals

Operations performed on independant variables:

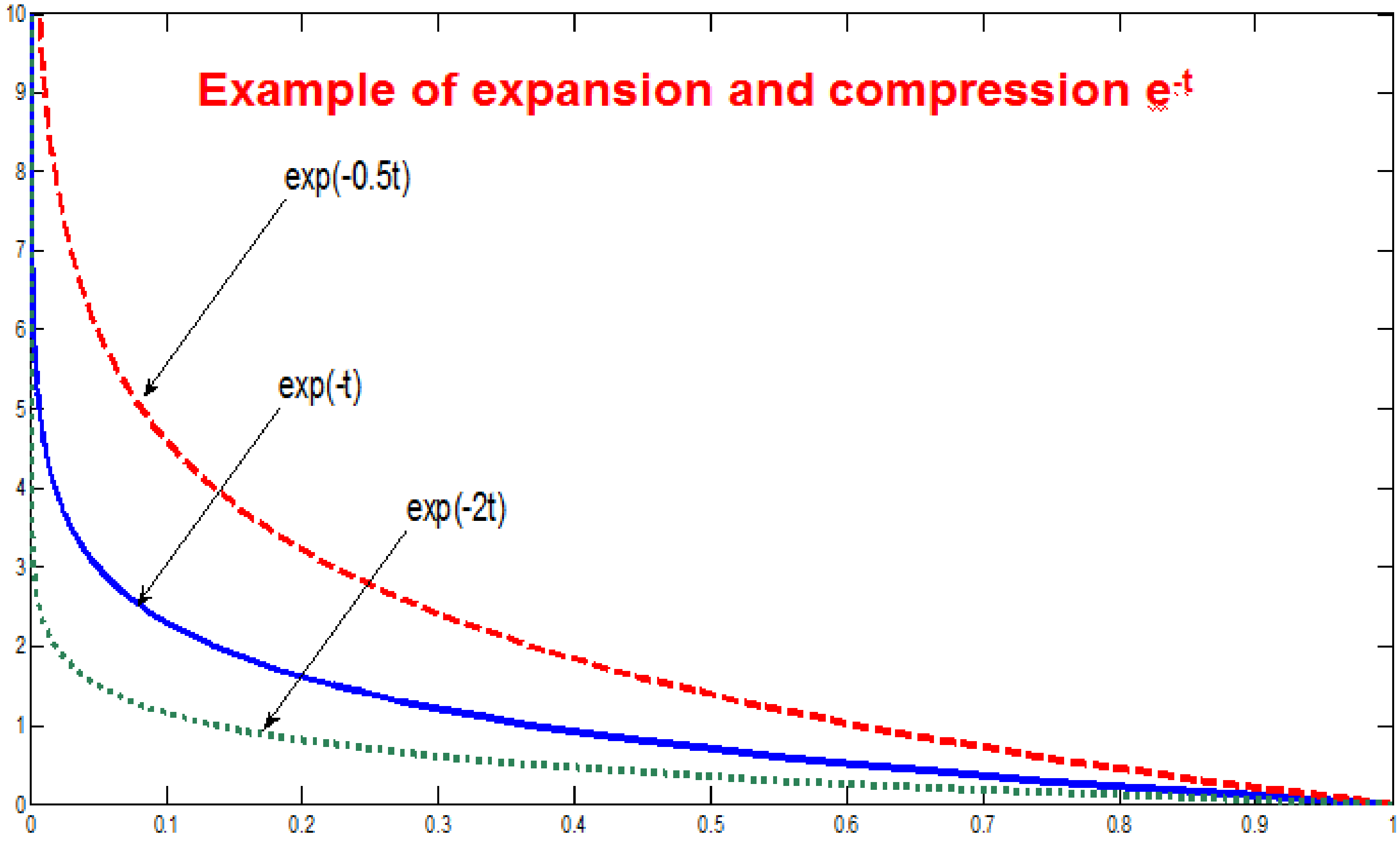
- Time scaling

Let y(t) is a compressed version of x(t). The signal y(t) obtained by scaling the independent variable, time t, by a factor k is defined by

$$y(t) = x(kt)$$

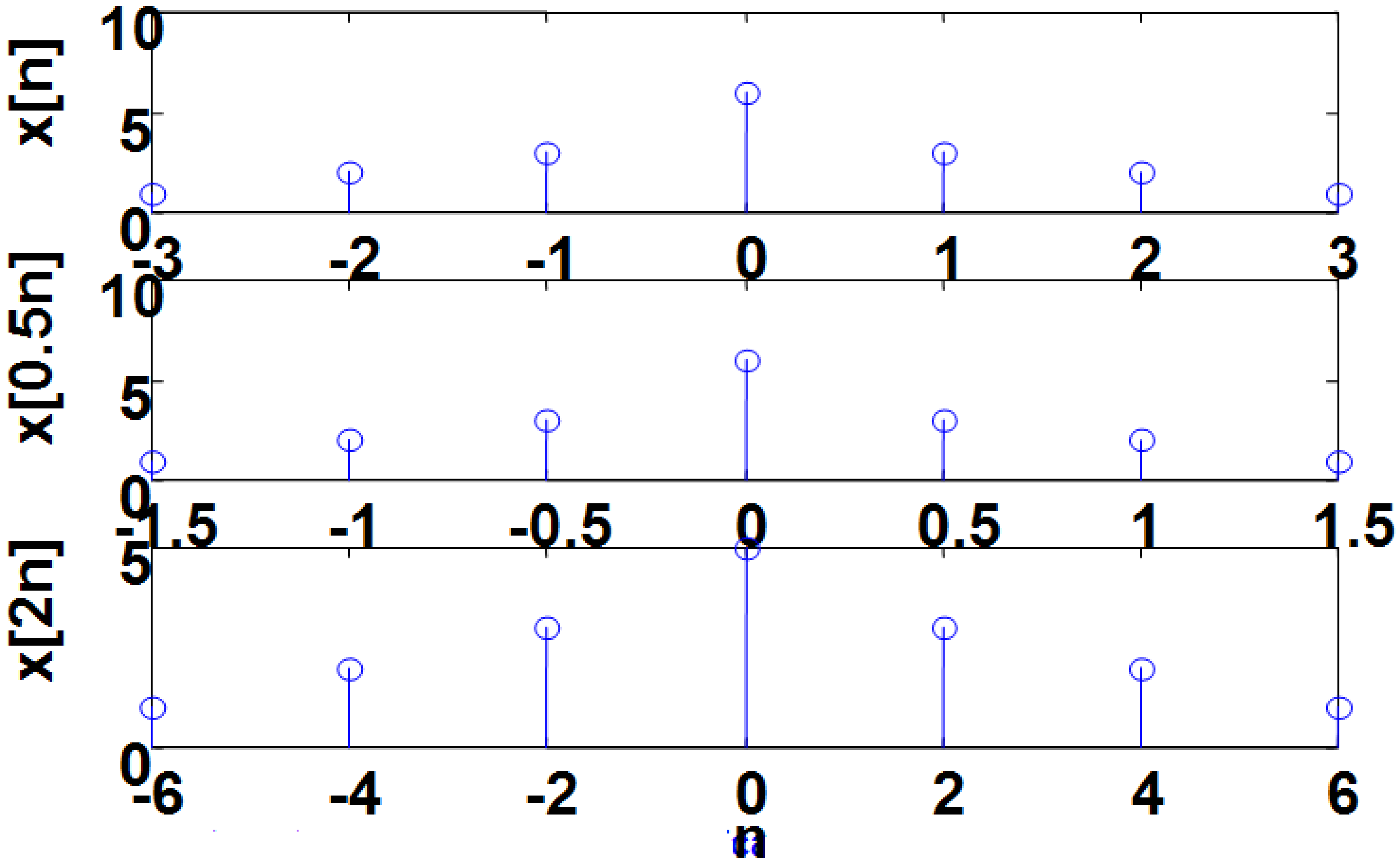
- if $k > 1$, the signal y(t) is a **compressed** version of x(t).
- If, on the other hand, $0 < k < 1$, the signal y(t) is an **expanded** (stretched) version of x(t).

Example:


Example of expansion and compression e-t
exp(-0.5t)
exp(-t)
exp(-2t)
0
1
2
3
4
5
6
7
8
9
10
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

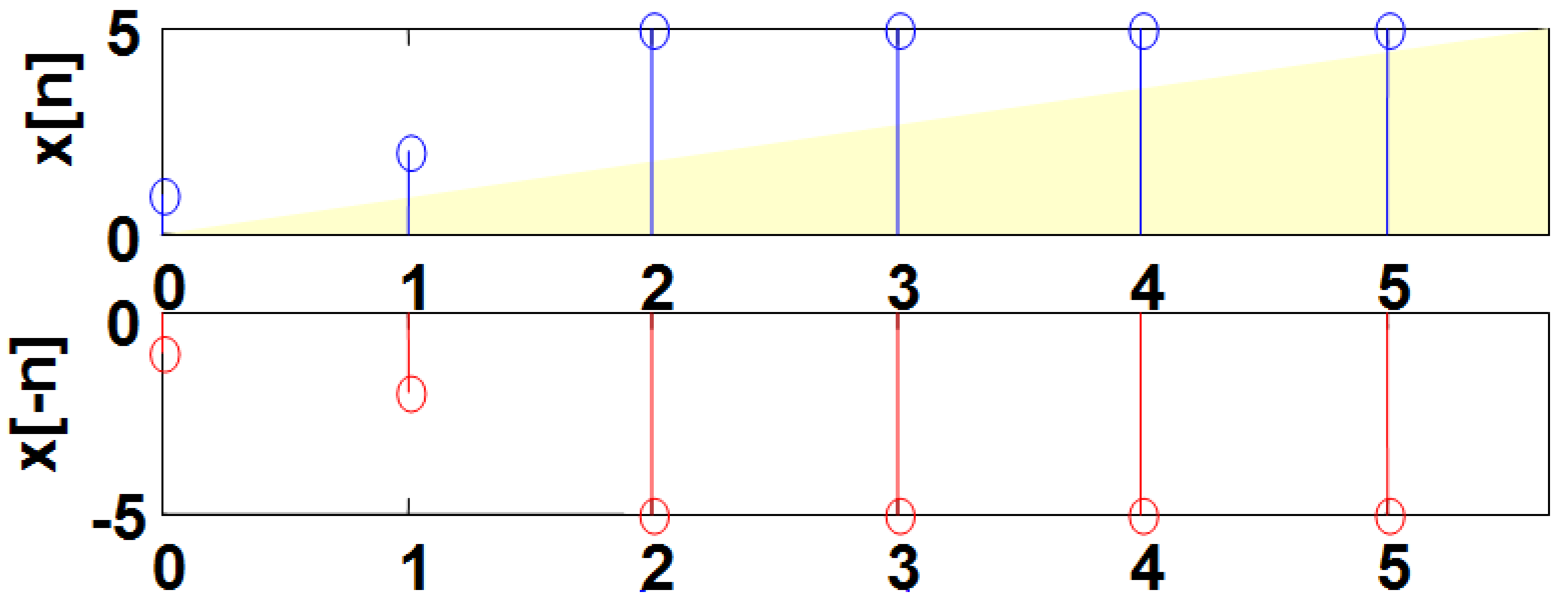
Basic Operations on Signals

Time scaling of discrete-time signals



Basic Operations on Signals

- Time Reversal
  - This operation reflects the signal about t = 0 and thus reverses the signal on the time scale.



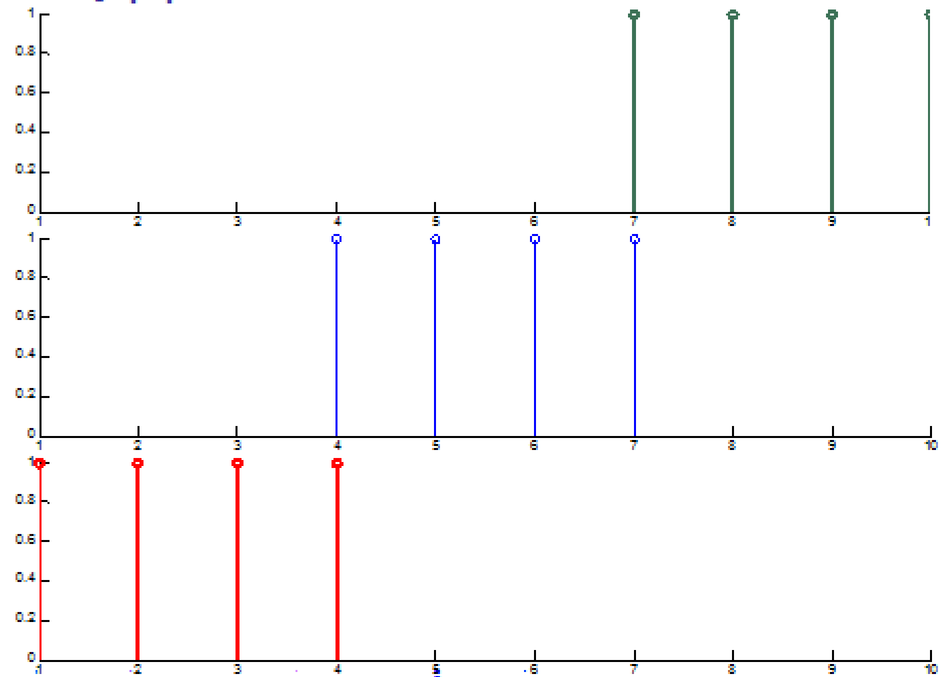
Time Shift

A signal may be shifted in time by replacing the independent variable n by n ± k, where k is an integer. If k is a positive integer, the time shift results in a delay of the signal by k units of time. If k is a negative integer, the time shift results in an advance of the signal by |k| units in time.

x[n-3] — Delay

x[n]

x[n+3] — Advance

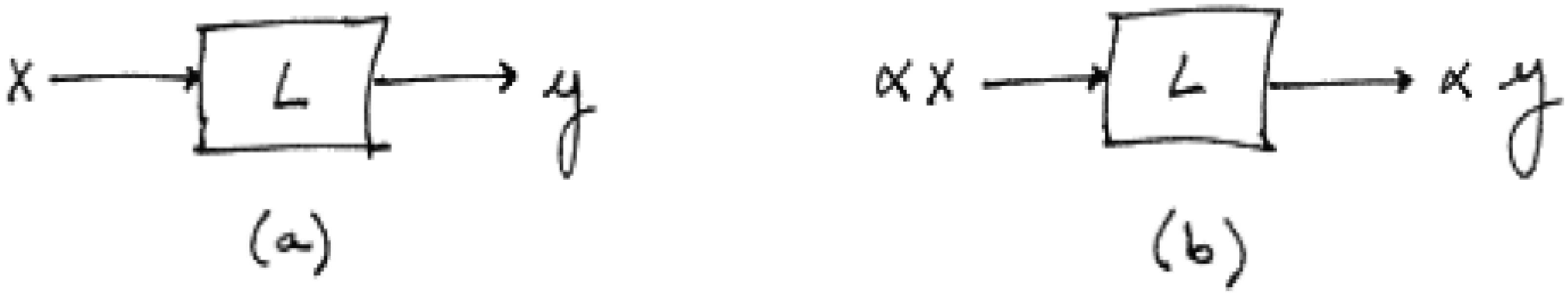


Linear Time-Invariant (LTI) Systems

- **Linear Systems**

If a system is linear, this means that when an input to a given system is scaled by a value, the output of the system is scaled by the same amount.

Linear Scaling



Figure

Linear Time-Invariant (LTI) Systems

- **Linear Systems**

If a system is linear, this means that when an input to a given system is scaled by a value, the output of the system is scaled by the same amount.

In part (a) of the figure above, an input x to the linear system L gives the output y If x is scaled by a value α and passed through this same system, as in part (b), the output will also be scaled by α.

A linear system also obeys the principle of superposition. This means that if two inputs are added together and passed through a linear system, the output will be the sum of the individual inputs' outputs.

That is, if (a) is true, then (b) is also true for a linear system. The scaling property mentioned above still holds in conjunction with the superposition principle. Therefore, if the inputs x and y are scaled by factors α and β, respectively, then the sum of these scaled inputs will give the sum of the individual scaled outputs:

Linear Time-Invariant (LTI) Systems

- **Linear Systems**

Superposition Principle

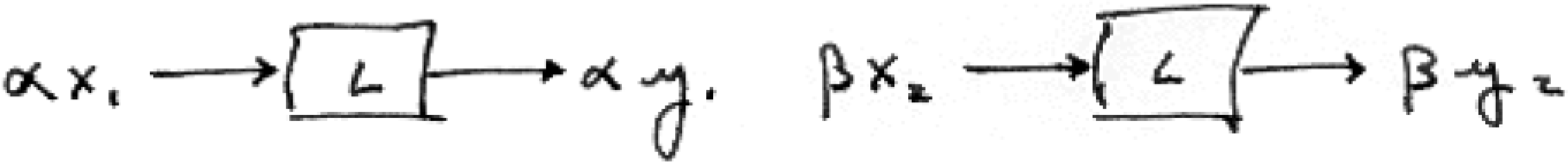
(a)

(b)

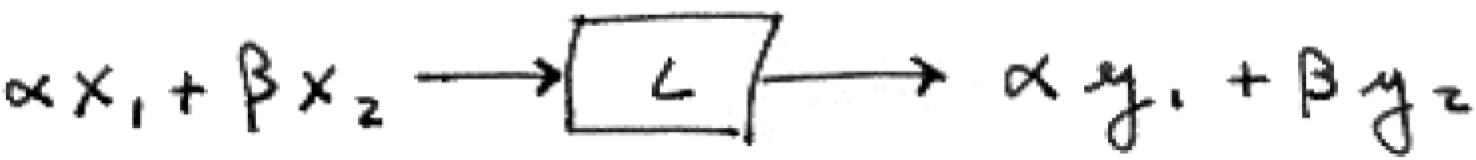
Figure : If (a) is true, then the principle of superposition says that (b) is true as well. This holds for linear systems.

- **Linear Systems**

Superposition Principle with Linear Scaling



(a)



(b)

Figure : Given (a) for a linear system, (b) holds as well.

Linear Time-Invariant (LTI) Systems

- **Time-Invariant Systems**

A time-invariant system has the property that a certain input will always give the same output, without regard to when the input was applied to the system.

In this figure, $x(t)$ and $x(t-t_0)$ are passed through the system TI. Because the system TI is time-invariant, the inputs $x(t)$ and $x(t-t_0)$ produce the same output. The only difference is that the output due to $x(t-t_0)$ is shifted by a time t_0.

Whether a system is time-invariant or time-varying can be seen in the differential equation (or difference equation) describing it. *Time-invariant systems are modeled with constant coefficient equations.* A constant coefficient differential (or difference) equation means that the parameters of the system are *not* changing over time and an input now will give the same result as the same input later.

Linear Time-Invariant (LTI) Systems

- **Time-Invariant Systems**

Time-Invariant Systems

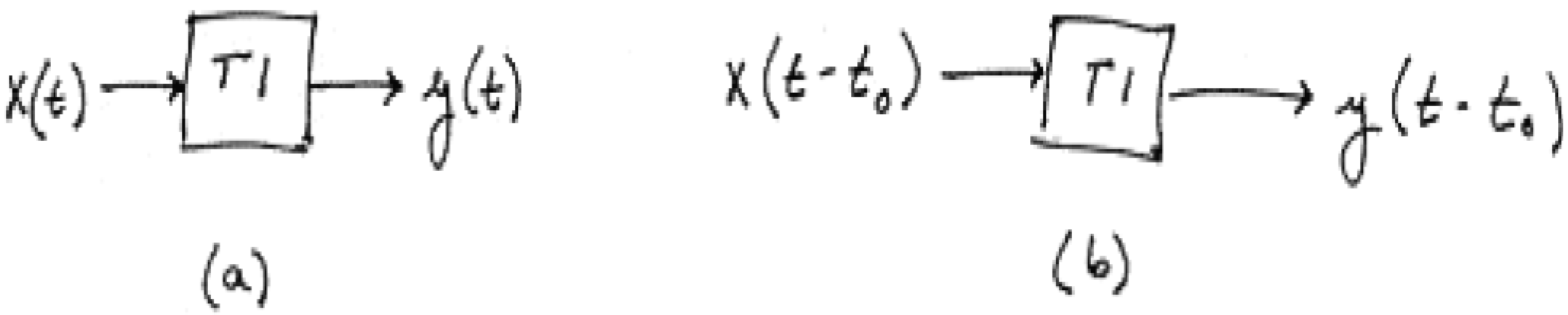


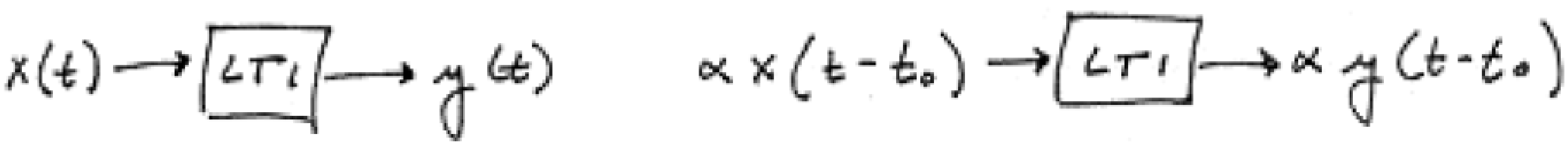
Figure : (a) shows an input at time t while (b) shows the same input t_0 seconds later. In a time-invariant system both outputs would be identical except that the one in (b) would be delayed by t_0.

Linear Time-Invariant (LTI) Systems

Certain systems are both linear and time-invariant, and are thus referred to as LTI systems.

As LTI systems are a subset of linear systems, they obey the principle of superposition. In the figure below, we see the effect of applying time-invariance to the superposition definition in the linear systems section above.

Linear Time-Invariant Systems

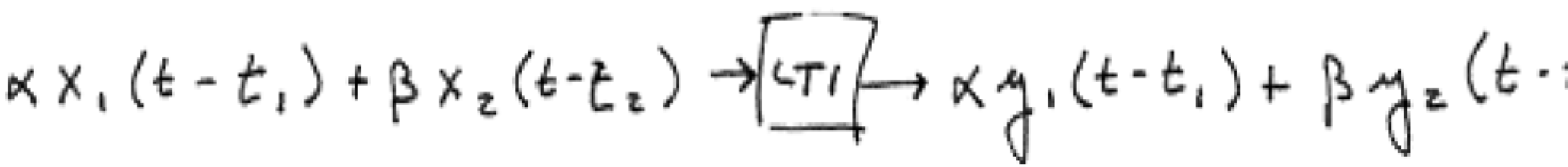


(a) (b)

Figure : This is a combination of the two cases above. Since the input to (b) is a scaled, time-shifted version of the input in (a), so is the output.

Superposition in Linear Time-Invariant Systems

(a)



(b)

Figure : The principle of superposition applied to LTI systems

Linear Time-Invariant (LTI) Systems

"LTI Systems in Series"

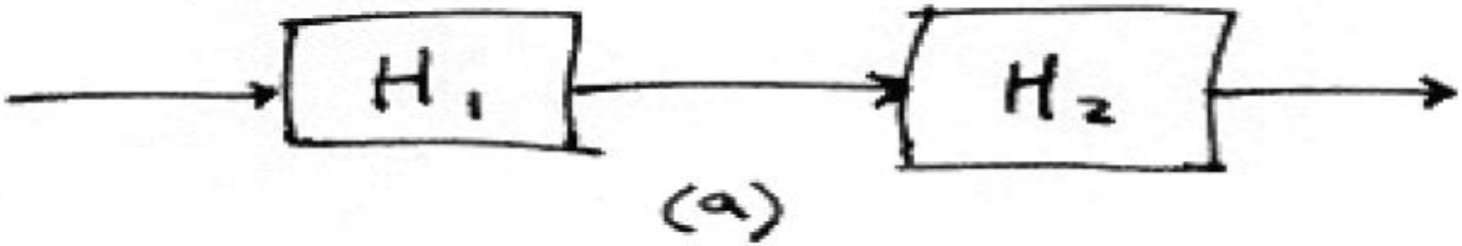
If two or more LTI systems are in series with each other, their order can be interchanged without affecting the overall output of the system. Systems in series are also called cascaded systems.

"LTI Systems in Parallel"

If two or more LTI systems are in parallel with one another, an equivalent system is one that is defined as the sum of these individual systems.

Linear Time-Invariant (LTI) Systems

Cascaded LTI Systems



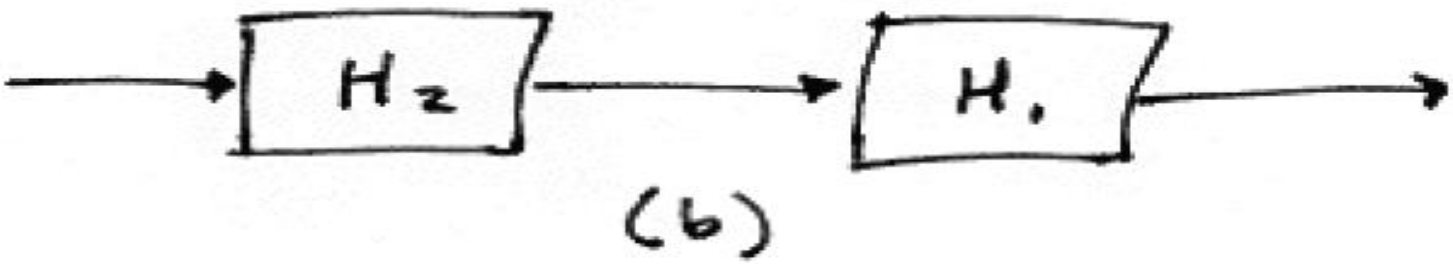


Figure : The order of cascaded LTI systems can be interchanged without changing the overall effect.

Parallel LTI Systems

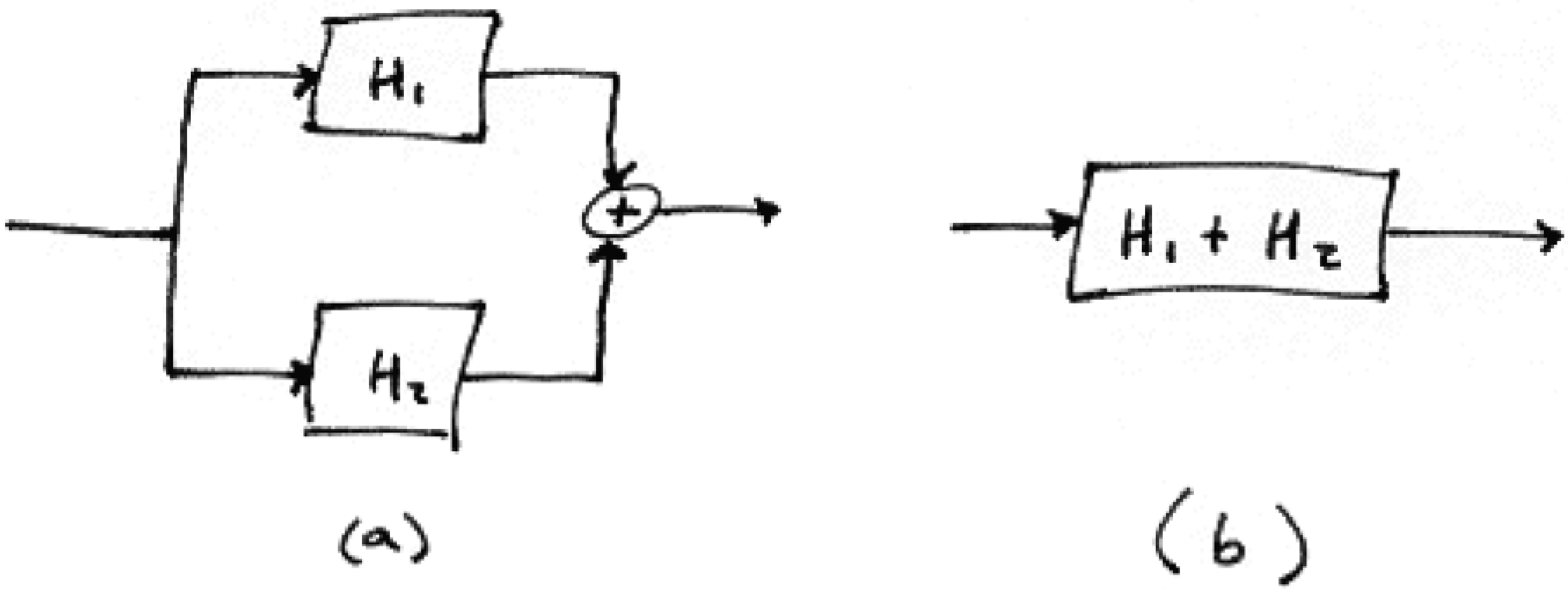


Figure : Parallel systems can be condensed into the sum of systems.

Digital Signal Processing

Course Instructor
Lecturer: WARQAA SHAHER

Lecture No. 4: Linear Time Invariant Systems

Third Class
Department of Computer and Software Engineering

http://www.engineering.uodiyala.edu.iq/
https://www.facebook.com/Engineer.College1

Lecture Outline

- Analysis of DT-LTI System
- Difference Equations
- Recursive Systems
- Non-recursive Systems
- Describing Digital Signals with Impluse Function
- Describing Digital LTI Responses
  1) Impulse Response
  2) Step Response
- Discrete-Time Convolution
- Discrete-Time Circular Convolution
- Discrete-Time Correlation
- Deconvolution

Analysis of DT-LTI system

There are two basic methods for analyzing the behavior or response of a Linear system to a given input signal.

1. Method based on direct solution of input-output equation for the system.
2. Decomposition of the input signal into a sum of elementary signals (usually samples)
   – Elementary signals are selected so that the response of the system to each signal component is easily determined.

Difference Equations

- Difference eqs can be used to describe how a linear, time-invariant, causal digital system works.
- If present i/p & o/p $\longrightarrow$ x[n] & y[n]

  then the preceding i/ps & o/ps

  x[n-1], x[n-2]….& y[n-1], y[n-2]…so on.
- Using these notations, the most general expression of the diff: eq: is

Weightings or Coefficients

$$a_0y[n]+a_1y[n-1]+a_2y[n-2]+\ldots+a_N\,y[n-N]$$
$$= b_0x[n]+b_1x[n-1]+b_2x[n-2]+\ldots+b_M\,x[n-M]$$

Difference Equations

- The equation can be presented more compactly as

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k] \longrightarrow Eq\ :(1)$$

- If we make $a_0 = 1$, the equation can be written as

Past inputs

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k] \longrightarrow Eq:(2)$$

Past outputs

- The eq:(2) form shows how each new o/p from the system can be calculated using past o/ps, present i/ps & past i/ps.

Recursive Systems

- When a digital system relies on both i/ps and past o/ps, it is referred to as a Recursive system.
- Eq: (2) is the equation for Recursive systems.

Non-Recursive Systems

- When the digital system relies only on i/ps (present & past), and not on past o/ps, it is referred to as a non recursive system.
- The following eq: gives the general form for this kind of filter.

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

$$Y[n]= b_0x[n]+b_1x[n-1]+b_2x[n-2]+\ldots+b_M x[n-M]$$

Example: Recursive Systems

- Example 1: A system has the difference eq: $y[n]= 0.5y[n-1]+x[n]$
  1. Identify all coefficients a_k & b_k.
  2. Is this a Recursive or Non-recursive diff: eq:.
  3. If the i/p $x[n]$ is as given in figure below, find the first 12 samples of the o/p, starting with $n=0$.

Solution:

1) Writing the o/ps on the left & i/ps on the right, we get

$$y[n]-0.5y[n-1]=x[n].$$

So, $a_0=1$, $a_1=-0.5$, $b_0=1$.

2) Since the o/p $y[n]$ depends on a past o/p $y[n-1]$, the digital system is recursive.

Example: Recursive Systems

3) Y[0]=0.5y[-1]+x[0] = 0.5(0.0)+1.0 = 1.0

(since the sys: is considered causal, the o/p cant begin until the i/p first becomes nonzero, in this case at n=0. Hence y[-1] = 0.

Y[1]=0.5y[0]+x[1] = 0.5(1.0)+1.0 = 1.5
Y[2]=0.5y[1]+x[2] = 0.5(1.5)+1.0 = 1.75
Y[3]=0.5y[2]+x[3] = 0.5(1.75)+1.0 = 1.875
Y[4]=0.5y[3]+x[4] = 0.5(1.875)+1.0 = 1.9375
Y[5]=0.5y[4]+x[5] = 0.5(1.9375)+1.0 = 1.9688
Y[6]=0.5y[5]+x[6] = 0.5(1.9688)+1.0 = 1.9844
Y[7]=0.5y[6]+x[7] = 0.5(1.9844)+1.0 = 1.9922
Y[8]=0.5y[7]+x[8] = 0.5(1.9922)+1.0 = 1.9961
Y[9]=0.5y[8]+x[9] = 0.5(1.9961)+1.0 = 1.9980
Y[10]=0.5y[9]+x[10] = 0.5(1.9980)+1.0 = 1.9990
Y[11]=0.5y[10]+x[11] = 0.5(1.9990)+1.0 = 1.9995

Example: Non-Recursive Systems

- Example 2: $y[n]=0.5x[n]-0.3x[n-1]$

1. Identify all coefficients a_k & b_k.
2. Is this a Recursive or Non-recursive diff: eq:.
3. For the i/p $x[n]=\sin(2\pi n/9)u[n]$. Find first 20 samples of the o/p.

Solution:

1) $a_0=1$, $b_0=0.5$, $b_1=-0.3$
2) The o/p does not depend on past o/ps, so the system is nonrecursive.

Example: Non-Recursive Systems

3) Because of the u[n] factor in the o/p, the values of the i/p before n=0 are zero.

n	**-1**	**0**	**1**	**2**	**3**	**4**	**5**
X[n]	0.0	0.0	0.643	0.985	0.866	0.342	-0.342
Y[n]	0.0	0.0	0.321	0.300	0.138	-0.089	-0.274
n	**6**	**7**	**8**	**9**	**10**	**11**	**12**
X[n]	-0.866	-0.985	-0.643	0.0	0.643	0.985	0.866
Y[n]	-0.330	-0.233	-0.026	0.193	0.321	0.300	0.138
n	**13**	**14**	**15**	**16**	**17**	**18**	**19**
X[n]	0.342	-0.342	-0.866	-0.985	-0.643	0.0	0.643
Y[n]	-0.089	-0.274	-0.330	-0.233	-0.026	0.193	0.321

Superposition in Diff: eq:

- In some instances, several i/ps may be applied to a system at the same time.
- When this happens, the system response to the sum of these inputs through superposition.
- Fortunately, when the sys: is linear, multiple inputs can be handled easily.

Tutorial 1

- **Tutorial 1:** A system is described by the difference equation

$$Y[n]=x[n]+0.5x[n-1]$$

Two i/ps are $x_1[n]=2u[n]$

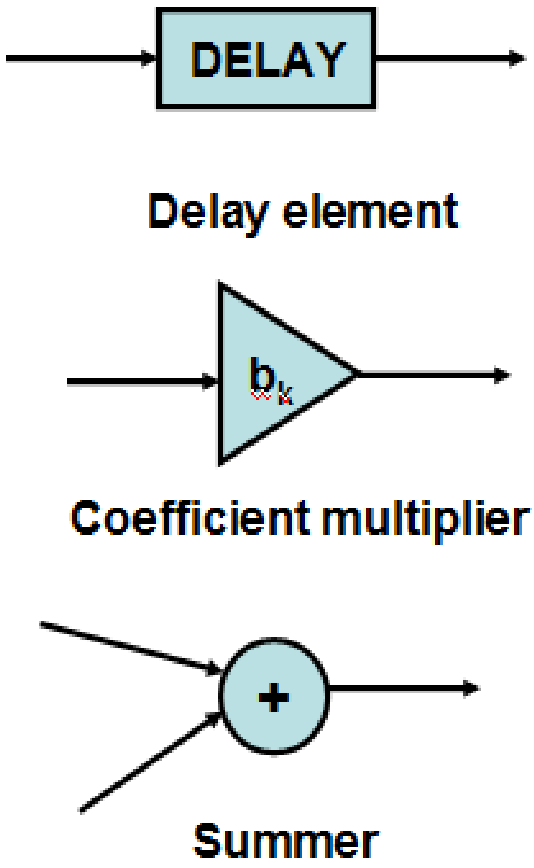
$$X_2[n]=\sin(n\pi/7)u[n]$$

Find and plot the first 20 samples of the o/p resulting from the combined effect of the 2i/ps.

Difference eq: Diagrams

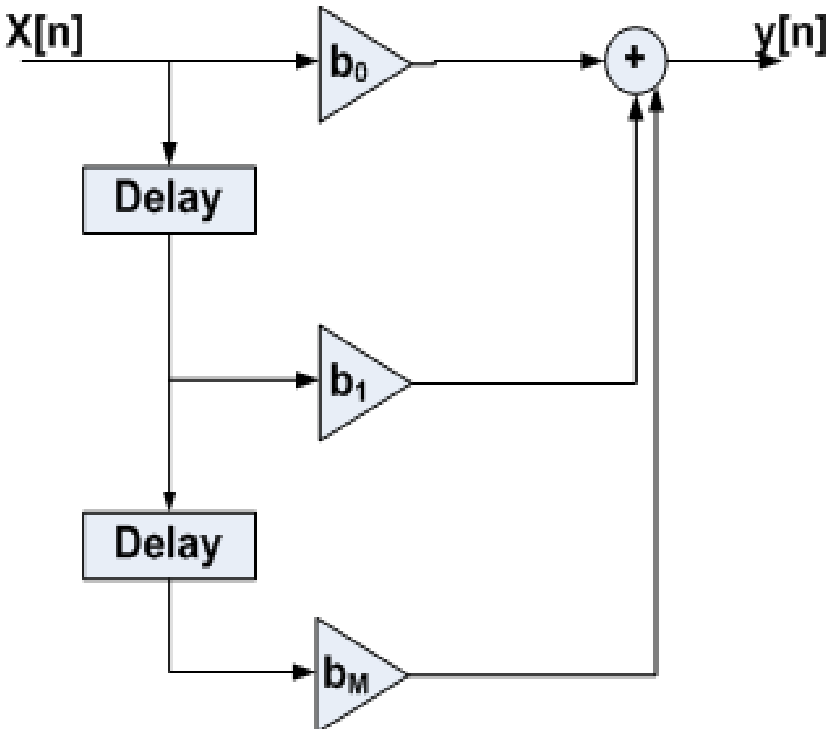
1. **Non-recursive Diff: eqs: diagram**

- The basic elements used in designing non-recursive diff: eq: diagrams are
  - Delay element
  - Coefficient multiplier
  - Summer



Difference eq: Diagrams

- A general non-recursive diff: eq: described previously can be presented schematically as below.

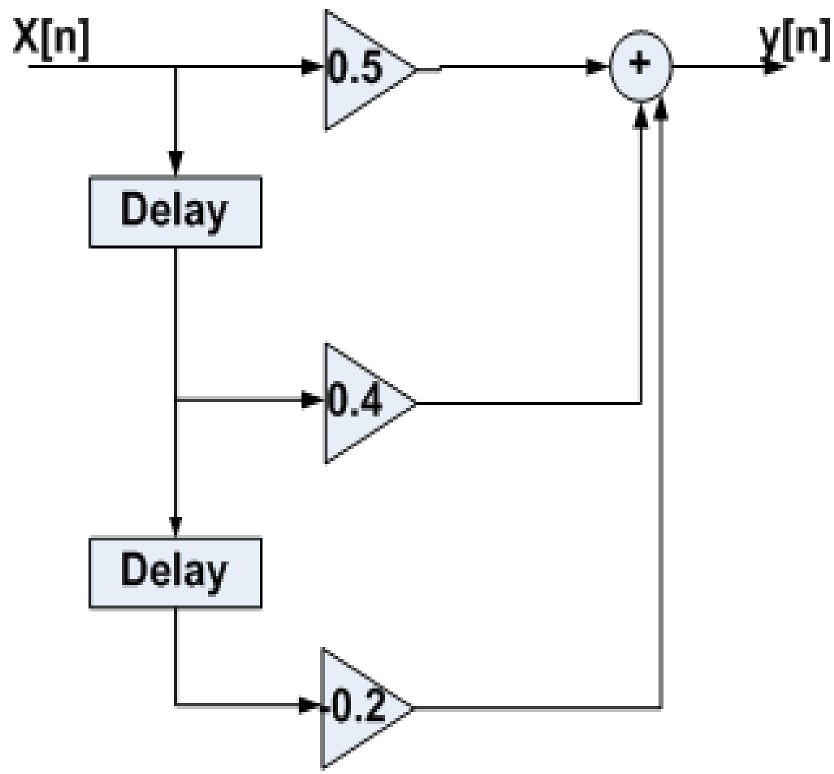


Non-recursive Difference equation Diagram

Difference eq: Diagrams

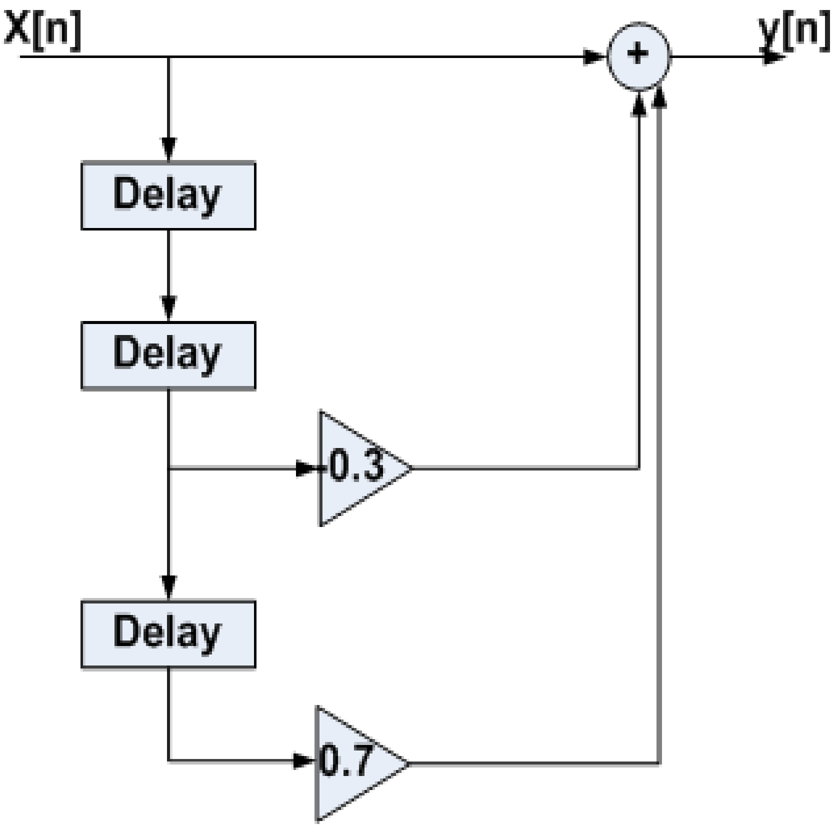
- **Example3:** Draw a diagram for the diff: eq: $y[n]=0.5x[n]+0.4x[n-1]-0.2x[n-2]$

  Solution:



Tutorial 2

- **Tutorial 2:** Write the diff: eq: that corresponds to the diagram given below

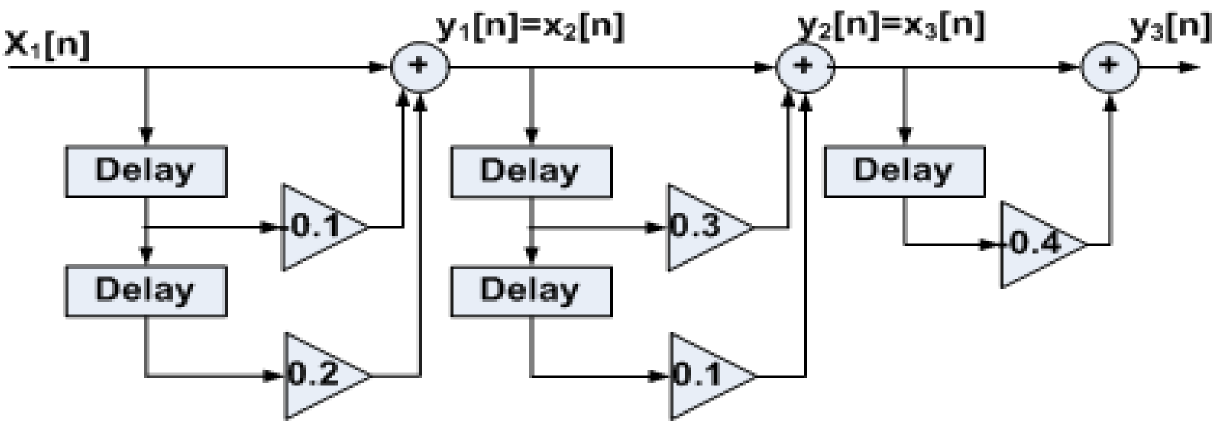


Difference eq: Diagrams

- Higher order system can be broken down into second order chunks, and cascaded together.
- When the order of the system is odd, a single first order section is added to the group of 2nd order section.
- The following example illustrates this point.

Difference eq: Diagrams

- **Example4:** Find the difference eq: of the following cascaded diagram



Solution:

The first stage produces the diff: eq:$Y_1[n]=x_1[n]-0.1x_1[n-1]+0.2x_1[n-2]$

The 2nd stage produces the diff: eq:$Y_2[n]=x_2[n]+0.3x_2[n-1]+0.1x_2[n-2]$

The 3rd stage produces the diff: eq:$Y_3[n]=x_3[n]-0.4x_3[n-1]$

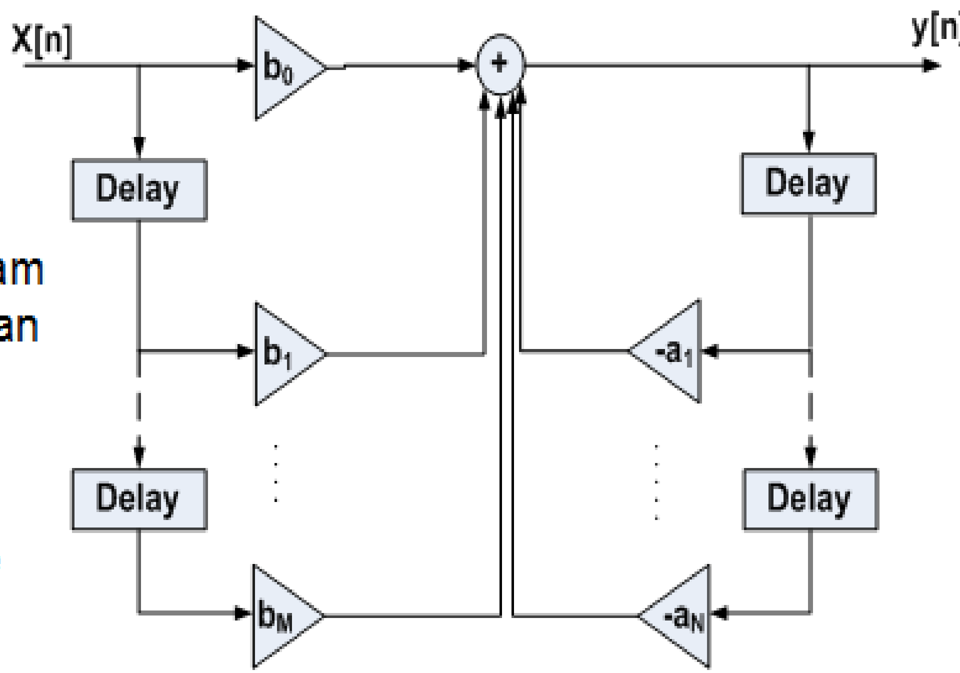
The final diff: eq: will become $y_3[n]=x_1[n]-0.2x_1[n-1]+0.19x_1[n-2]-0.058x_1[n-3]-0.008x_1[n-5]$

Difference eq: Diagrams

2. Recursive difference equations

1. Direct form 1 Realization

- In this form, the diagram of recursive diff: eq: can be made by using the previous diagram elements.
- The general recursive diff: eq: described previously cab be depicted as,



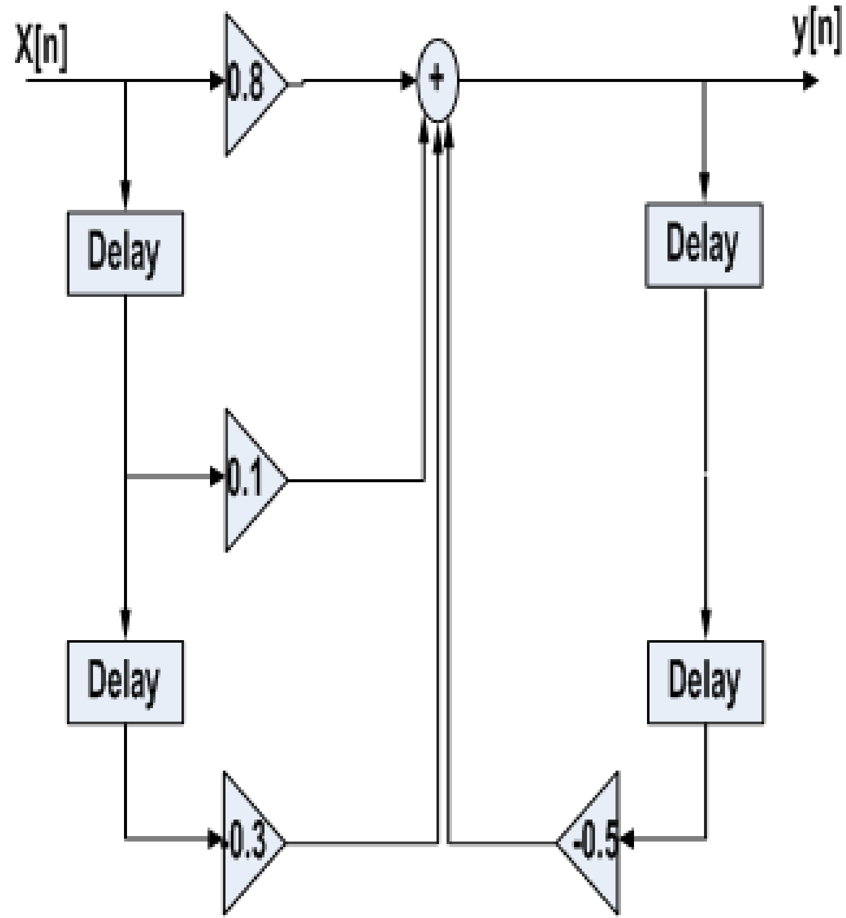
Recursive Direct Form 1 Realization
Difference equation Diagram

Difference eq: Diagrams

- **Example5:** Draw a direct form1 realization diff: eq: to describe the recursive system.
  $y[n]+0.5y[n-2]=0.8x[n]+0.1x[n-1]-0.3x[n-2]$

  Solution: Rearranging the eq: we get

  $$y[n]=-0.5y[n-2]+0.8x[n]+0.1x[n-1]-0.3x[n-2]$$



Difference eq: Diagrams

2. Direct for 2 Realization

- The Form 1 realization is not the most efficient to implement a recursive diff: eq:
- A much efficient way to implement recursive diff: eq: is direct form 2 realization.
- This realization requires the use of an intermediate signal w[n] that records salient information about the history of the system in place of past i/ps and past o/ps.
- The two eqs: that define DF2 realization are

$$w[n] = x[n] - \sum_{k=1}^{N} a_k w[n-k]$$

$$y[n] = \sum_{k=0}^{N} b_k w[n-k]$$

Difference Equation

A discrete-time signal $s(n)$ is **delayed** by n_0 samples when we write $s(n-n_0)$, with $n_0 > 0$. Choosing n_0 to be negative advances the signal along the integers. As opposed to analog delays (pg ??), discrete-time delays can *only* be integer valued. In the frequency domain, delaying a signal corresponds to a linear phase shift of the signal's discrete-time Fourier transform: $s(n-n_0) \leftrightarrow e^{-(j2\pi f n_0)} S\left(e^{j2\pi f}\right)$.

Linear discrete-time systems have the superposition property.

Superposition

$$S\left(a_1 x_1(n) + a_2 x_2(n)\right) = a_1 S\left(x_1(n)\right) + a_2 S\left(x_2(n)\right)$$

A discrete-time system is called **shift-invariant** (analogous to time-invariant analog systems (pg ??)) if delaying the input delays the corresponding output.

Shift-Invariant

$$If S\left(x(n)\right) = y(n), Then S\left(x(n-n_0)\right) = y(n-n_0)$$

Difference Equation

We use the term shift-invariant to emphasize that delays can only have integer values in discrete-time, while in analog signals, delays can be arbitrarily valued.

We want to concentrate on systems that are both linear and shift-invariant. It will be these that allow us the full power of frequency-domain analysis and implementations. Because we have no physical constraints in "constructing" such systems, we need only a mathematical specification. In analog systems, the differential equation specifies the input-output relationship in the time-domain. The corresponding discrete-time specification is the **difference equation** .

The Difference Equation

$$y(n) = a_1 y(n-1) + \ldots + a_p y(n-p) + b_0 x(n) + b_1 x(n-1) + \ldots + b_q x(n-q)$$

Here, the output signal $y(n)$ is related to its *past* values $y(n-l)$, $l = \{1, \ldots, p\}$, and to the current and past values of the input signal $x(n)$. The system's characteristics are determined by the choices for the number of coefficients p and q and the coefficients' values $\{a_1, \ldots, a_p\}$ and $\{b_0, b_1, \ldots, b_q\}$.

Analysis of DT-LTI system

- The second method for analyzing the behavior of LTI system to a given i/p signal is first to decompose or resolve the input signal into a sum of elementary signals.
- The elementary signals are selected so that the response of the system to each signal component is easily determined.
- Then using the linearity property of the sys, the responses of the system to the elementary signals are added to obtain the total response of the sys to the given i/p signal.
- The elementary signal we choose to analyze LTI system is impulse signal.

Resolution of DT signals into Impulses

Suppose we have an arbitrary signal x[n] that we wish to resolve into a sum of unit sample sequence.

Consider the product of a signal x[n] and the impulse sequence δ[n], written as x[n]δ[n]

Since δ[n]=1 only at n=0,so we can write

$$x[n]\delta[n] = x[0]\delta[n]$$

If we were to repeat the multiplication of x[n] with δ[n-k], where δ [n-k] is time shifted impulse sequence, the result will be a sequence that is zero everywhere except at n=k,

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

This property allows us to express x[n] as the following weighted sum of time shifted impulses:

$$x[n] = \ldots + x[-2]\delta[n+2] + x[-1]\delta[n+1] + x[0]\delta[n] + x[1]\delta[n-1] + x[2]\delta[n-2] + \ldots$$

Or in concise form as:

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

Right hand side gives us the resolution of any arbitrary signal x[n] into weighted sum of shifted unit sample sequences.

Impulse Response of LTI System

The **impulse response** is exactly what its name implies - the response of an LTI system, such as a filter, when the system's input is the unit impulse (or unit sample). A system can be completed describe by its impulse response due to the idea mentioned above that all signals can be represented by a superposition of signals. An impulse response gives an equivalent description of a system as a transfer fucntion, since they are Laplace Transforms of each other.

NOTATION: Most texts use $\delta(t)$ and $\delta[n]$ to denote the continuous-time and discrte-time impulse response, respectively.

Impulse Response of LTI System

The response of a system to the unit input $\delta[n]$ is called the *Impulse Response*, normally written as $h[n]$.

In the case of Linear-Time-Invariant (LTI) systems it completely characterizes their behavior. This is because every input sequence can be described as a linear combination of delayed copies of the unit sequence, and using linearity and time-invariance, the reponse can be built as a superposition of delayed impulse reponses. Such superposition can be written as the sum:

Impulse Response of LTI System

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

called *convolution* of the input and the impulse response.

The convolution has interesting properties, such as *commutativity* ($x*y = y*x$), *associativity* ($(x*y)*z = x*(y*z)$) and distributivity ($x*(y+z) = x*y+x*z$).

Furthermore, properties of LTI systems are simply described by $h[n]$:

- Stability: A system is stable if and only if the impulse response is absolutely summable ($\sum_{-\infty}^{\infty} |x[n]| < \infty$)
- Causality: A system is causal if and only if its impluse response is a causal signal.

Finally, simple interconnection schemes of systems result in simple composition of the impulse response:

- *Cascade connection*: The impulse response is the convolution of the responses ($h_{tot} = h_1 * h_2$). Important consequence is the fact that order is not important in cascade connections. Stability, passivity and losslessness are preserved.
- *Parallel connection*: The response is the sume of the responses ($h_{tot} = h_1 + h_2$). Stability is preserved.

Impulse Response of LTI System

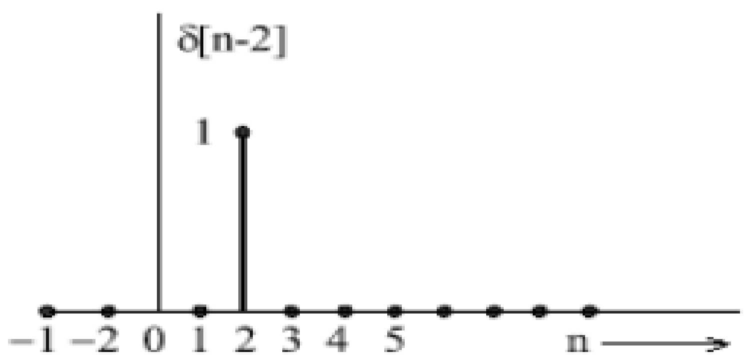
The unit impulse was described above as:

$$\delta[n] = 0, \quad n \neq 0$$
$$\delta[n] = 1, \quad n = 0$$

This is also sometimes known as the Kronecker ***delta function***

This can be tabulated

n		-2	-1	0	1	2	3	4	5	6	
δ[n]	0	0	0	1	0	0	0	0	0	0	0
δ[n-2]	0	0	0	0	0	1	0	0	0	0	0



Shifted impulse sequence, $\delta[n-2]$

Impulse Response of LTI System

The third row of table 1 gives the values of the shifted impulse δ[n – 2]
Now consider the following signal:

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

Table 2 shows the individual sequences and their sum.

n		-2	-1	0	1	2	3	4	5	6	
2δ[n]	0	0	0	2	0	0	0	0	0	0	0
4δ[n-1]	0	0	0	0	4	0	0	0	0	0	0
6δ[n-2]	0	0	0	0		6	0	0	0	0	0
4δ[n-3]	0	0	0	0	0	0	4	0	0	0	0
2δ[n-4]	0	0	0	0	0	0	0	2	0	0	0
x[n]	0	0	0	2	4	6	4	2	0	0	0

Table 2

Hence any sequence can be represented by the equation:

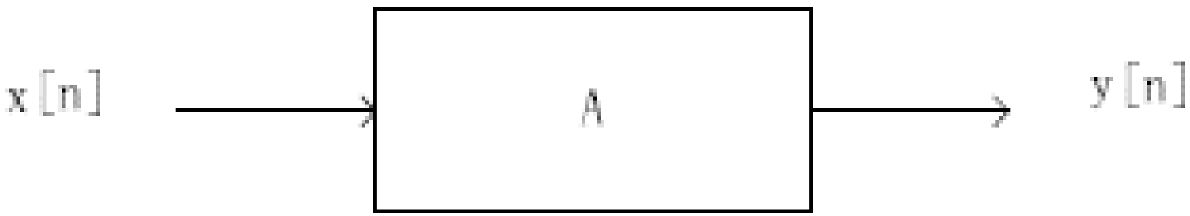
$$x[n] = \sum_{k} x[k]\delta[n-k]$$

$$= + \, x[-1]\delta[n+1] + x[0]\delta[n] + x[1]\delta[n-1] + x[2]\delta[n-2] + \ldots\ldots$$

When the input to an FIR filter is a unit impulse sequence, ***x[n] = δ[n]***, the output is known as the **unit impulse response**, which is normally donated as ***h[n]***.

Impulse Response of LTI System

Suppose that a signal x[n] is given as input to a linear system



First, let us look at $x[n]\delta[n-k]$ as a function of $n \in \mathbb{Z}$, where k is fixed.

$$x[n]\delta[n-k] = \begin{cases} x[k] & \text{if } n = k \\ 0 & \text{if } n \neq k \end{cases}$$

Impulse Response of LTI System

This holds for any fixed k,

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

This is the *sifting property*.

The system is linear. If the response of the system to $\delta[n-k]$ (where k is fixed and $n \in \mathbb{Z}$) is $h_k[n]$, then

Impulse Response of LTI System

the output of the system for $x[n]$ is given by

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

If in addition, the system is time-invariant(LTI), then if we let $h_0[n] = h[n]$ to be the response to $\delta[n]$, then $h_k[n] = h[n-k]$, so we have

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

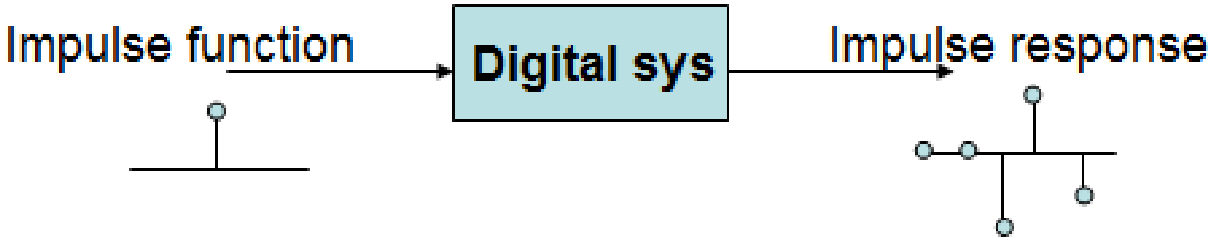
Impulse Response of LTI System

Summary

$h[n]$ is the response of an LTI system to $\delta[n]$ and is called the impulse response of an LTI system. Then $y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$ is the response of the system to $x[n]$.

Impulse Response & Difference Eq:

- When i/p to a system is a unit impulse function the o/p from the sys: is the unit impulse response as shown in figure.



- The diff: eq: for a sys can be used to calculate the impulse response for the system.
- Just replace x[n] by $\delta[n]$ and y[n] by h[n] an further steps are usual.

Impulse Response & Difference Eq:

- **Example7:** Find the first 6 samples of the impulse for the different equation.

  Y[n]-0.4y[n-1]=x[n]-x[n-1]

Solution:

First, replace x[n] with $\delta[n]$ and y[n] with h[n] to give

$h[n]-0.4h[n-1]= \delta[n] - \delta[n-1]$

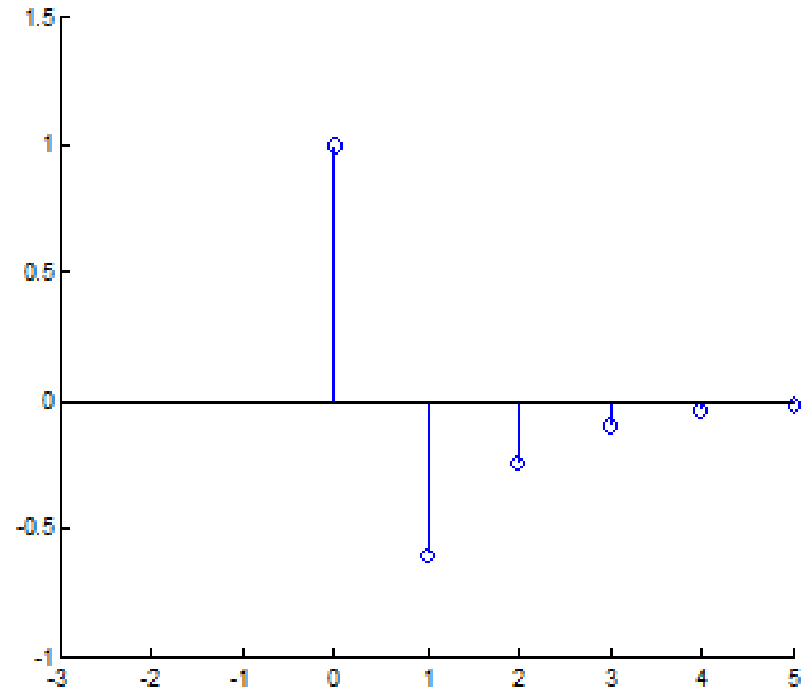
Starting with n=0:

$h[0]=0.4h[-1]+\delta[0]-\delta[-1]$

$h[0]=0.4(0.0)+1.0-0.0=1.0$

And further… h[1]=-0.6

h[2]=-0.24; h[3]=-0.096; h[4]=-0.0384; h[5]=-0.01536



Infinite Impulse Response (IIR)

- In previous example, the impulse response never dies away. Reason is the new o/ps depends on old o/ps.
- The impulse response that never dies away or tends to infinite is called IIR and is typical for recursive diff: eq:.

Infinite Impulse Response (IIR)

- **Example8:** Find & plot first 6 samples in the impulse response for the system

$y[n]=0.25(x[n]+x[n-1]+x[n-2]+x[n-3])$

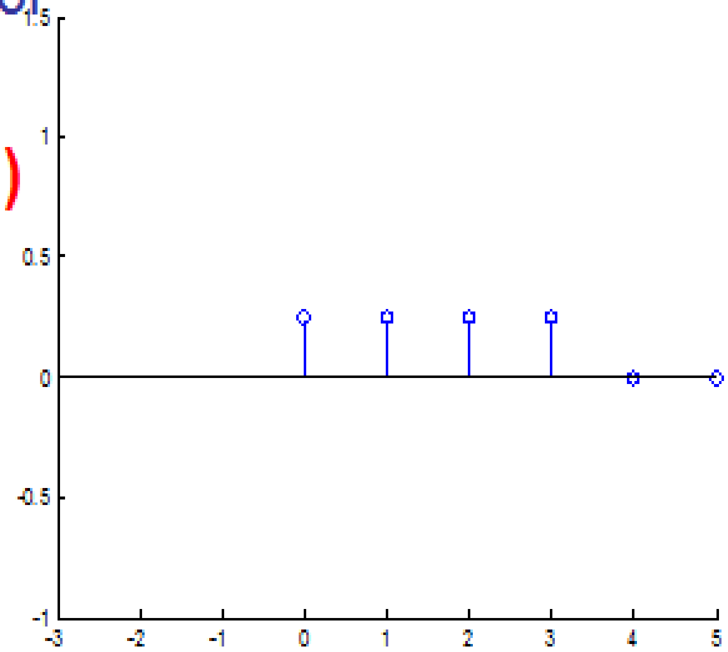
Solution:

Substituting symbols for impulse response we get

$h[n]=0.25(\delta[n]+\delta[n-1]+\delta[n-2]+\delta[n-3])$

So we get,

$h[0]=h[1]=h[2]=h[3]=0.25;\ h[4]=h[5]=0.0$



Finite Impulse Response (FIR)

- In example#8, note that the impulse response drops to zero after a finite no of nonzero samples.
- If the impulse response drops to zero after a finite no: of nonzero samples the response is said to as FIR and is typical for non-recursive diff: eqs:.

Step Response & Difference Equations

- It is a response for a system to a unit step function.
- Step function (i/p) is u [n]
- Step response (o/p) is s [n]
- There are two simple ways to find the step response for a sys:
    1. Use of diff: eq: with u[n] as i/p
    2. Determine impulse response and sum it

Tutorial 3

Tutorial#3:

Find & plot the step response for the system

$$y[n]-0.2y[n-2]=0.5x[n]+0.3x[n-1]$$

by the following methods:

1. Use of diff: eq: with u[n] as i/p
2. Determine impulse response and sum it

Step Response of an LTI System

The step response of an LTI system is simply the response of the system to a unit step. It conveys a lot of information about the system. For a discrete-time system with impulse response $h[n]$, the step response is $s[n] = u[n] * h[n]$. However, based on the commutative property of convolution, $s[n] = h[n] * u[n]$, and therefore, $s[n]$ can be viewed as the response to input $h[n]$ of a discrete-time LTI system with unit impulse response. We know that $u[n]$ is the unit impulse response of the accumulator. Therefore,

$$s[n] = \sum_{k=-\infty}^{n} h[k].$$

Step Response of an LTI System

From this equation, $h[n]$ can be recovered from $s[n]$ using the relation

$$h[n] = s[n] - s[n-1].$$

It can be seen the step response of a discrete-time LTI system is the ***running sum of its impulse*** response. Conversely, the impulse response of a discrete-time LTI system is the ***first difference*** of its step response.

Step Response of an LTI System

Similarly, in continuous time, the step response of an LTI system is the running integral of its impulse response,

$$s(t) = \int_{-\infty}^{t} h(\tau)d\tau ,$$

and the unit impulse response is the first derivative of the unit step response,

$$h(t) = \frac{ds(t)}{dt} = s'(t).$$

Therefore, in both continuous and discrete time, the unit step response can also be used to characterize an LTI system.

Step Response of an LTI System

The unit impulse response can be derived from the unit step response as

$$h(t) = \frac{ds(t)}{dt} = s'(t)$$

In discrete time

$$s[n] = u[n] * h[n] = \sum_{k=-\infty}^{n} h[k]$$

$$h[n] = s[n] - s[n-1]$$

Discrete-Time Convolution

- If $h(n)$ is the system impulse response, then the input-output relationship is a convolution.
- It is used for designing filter or a system.
- Definition of convolution:

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

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

Discrete-Time Convolution

The simplest example of convolution is the multiplication of two polynomials. e.g.

$$y = (4x^2 - 3x + 9)\left(3x^2 + 4x + 4\right)$$

This is calculated by:

$$y = \left(\left(4x^2 \times 3x^2\right) + \left(4x^2 \times 4x\right) + \left(4x^2 \times 4\right)\right) + \left(\left(-3x \times 3x^2\right) + \left(-3x \times 4x\right) + \left(-3x \times 4\right)\right) + \left(\left(9 \times 3x^2\right) + \left(9 \times 4x\right) + \left(9 \times 4\right)\right)$$

$$y = 12x^4 + 16x^3 + 16x^2 - 9x^3 - 12x^2 - 12x + 27x^2 + 36x + 36$$

$$y = 12x^4 + 7x^3 + 31x^2 + 24x + 36$$

Convolution is a weighted moving average with one signal ***flipped back to front***:

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

Discrete-Time Convolution

A tabulated version of convolution

n	n<0	0	1	2	3	4	5	6	7	n<7
x[n]	0	2	4	6	4	2	0	0	0	0
h[n]	0	3	-1	2	1					
h[0]x[n]	0	6	12	18	12	6	0	0	0	0
h[1]x[n-1]	0	0	-2	-4	-6	-4	-2	0	0	0
h[2]x[n-2]	0	0	0	4	8	12	8	4	0	0
h[3]x[n-3]	0	0	0	0	2	4	6	4	2	0
y[n]	0	6	10	18	16	18	12	8	2	0

$$h[0]x[n] = x[0] * h[0] + x[1] * h[0] + x[2] * h[0] + x[3] * h[0] + x[4] * h[0]$$

$$h[0]x[n] = 2 * 3 + 4 * 3 + 6 * 3 + 4 * 3 + 2 * 3$$

$$h[0]x[n] = 6 + 12 + 18 + 12 + 6$$

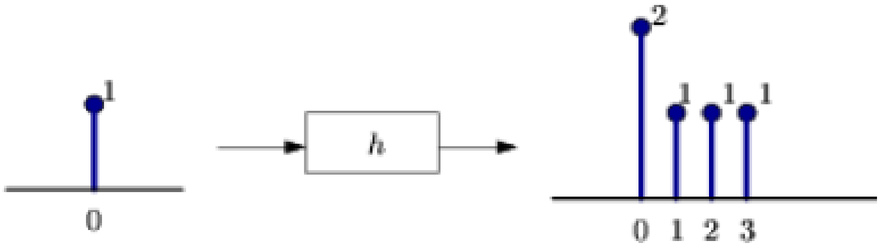
$$h[1]x[n-1] = x[0] * h[1] + x[1] * h[1] + x[2] * h[1] + x[3] * h[1] + x[4] * h[1]$$

$$h[1]x[n-1] = 2 * -1 + 4 * -1 + 6 * -1 + 4 * -1 + 2 * -1$$

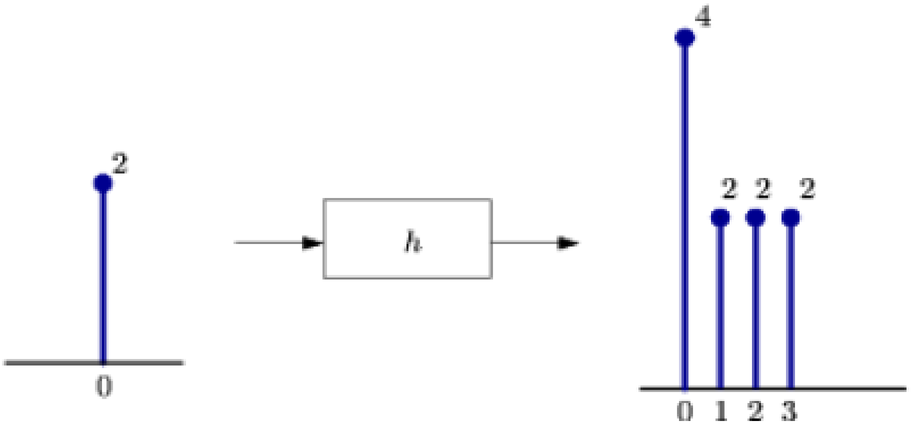
$$h[1]x[n-1] = -2 + -4 + -6 + -4 + -2$$

Discrete-Time Convolution

The diagrams below show how convolution works.

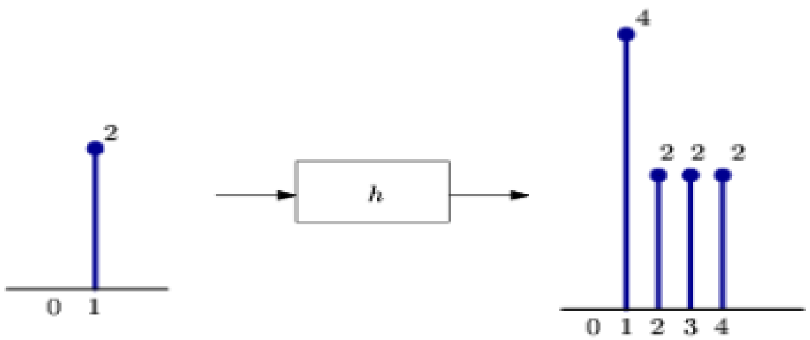


A single impulse input yields the system's impulse response

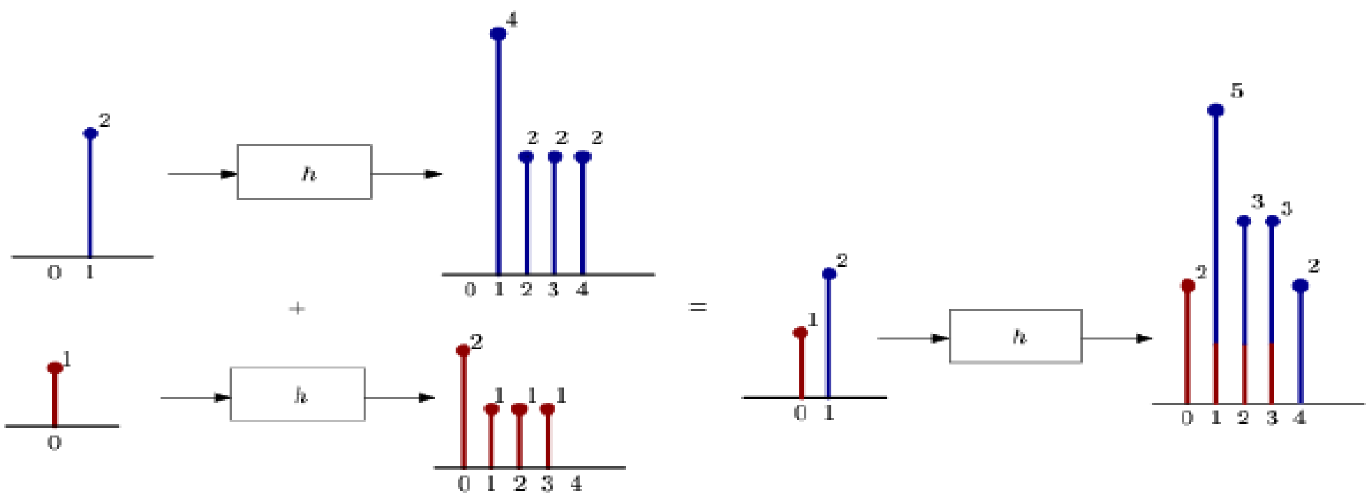


A scaled impulse input yields a scaled response, due to the *scaling property* of the system's linearity.

Discrete-Time Convolution



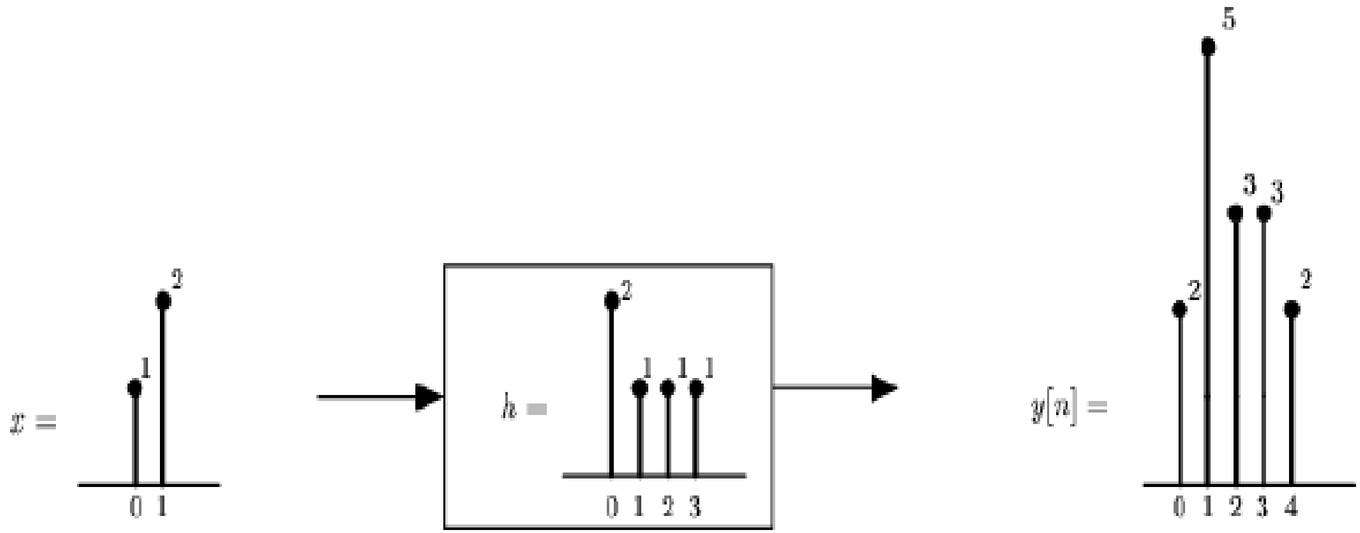
This demonstrates the use the *time-invariance property* of the system to show that a delayed input results in an output of the same shape, only delayed by the same amount as the input



Discrete-Time Convolution

This now demonstrates the additivity portion of the linearity property of the system to complete the picture. Since any discrete-time signal is just a sum of scaled and shifted discrete-time impulses, we can find the output from knowing the input and the impulse response

No if we convolve x(n) with h(n) as shown in Figure 9 we will get the output y(n)

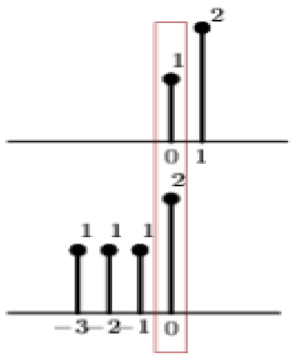


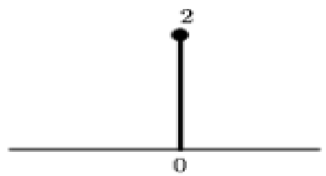
This is the end result that we are looking to find

Discrete-Time Convolution

The following diagrams are a breakdown of how the y(n) output is achieved.

$n = 0$

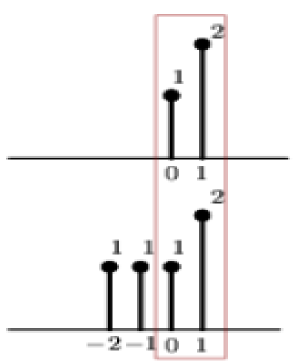


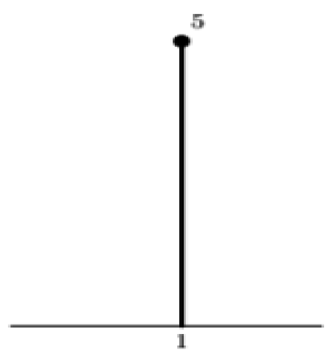


$x[0] \times h[-0] = 1 \times 2 = 2$

The impulse response, h , is reversed and begin its traverse at time 0.

$n = 1$



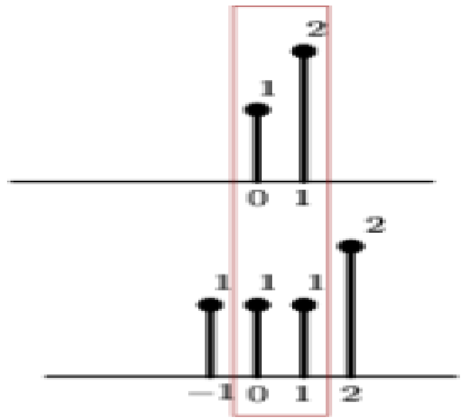


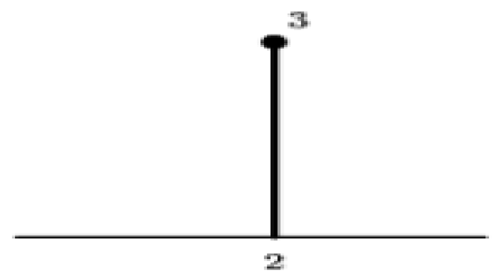
$(x[0] \times h[1]) + (x[1] \times h[0]) = 1 + 4 = 5$

Continuing the traverse. At time 1 , the two elements of the input signal are multiplied by two elements of the impulse response.

Discrete-Time Convolution

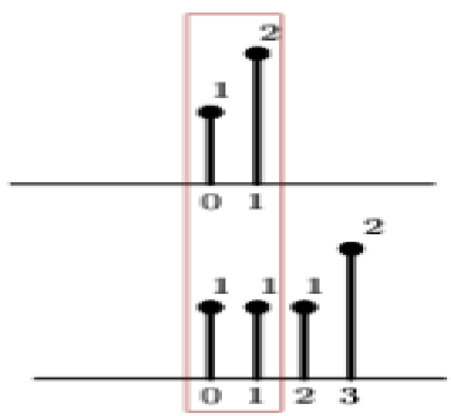
$n = 2$

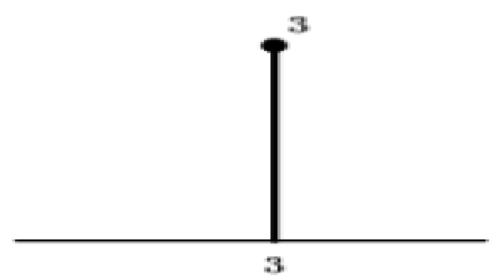




$(x[0] \times h[2]) + (x[1] \times h[1]) = 1 + 2 = 3$

$n = 3$





$(x[0] \times h[3]) + (x[1] \times h[2]) = 1 + 2 = 3$

Discrete Time Convolution

Overview

Convolution is a concept that extends to all systems that are both **linear and time-invariant** (LTI). The idea of **discrete-time convolution** is exactly the same as that of continuous-time convolution . For this reason, it may be useful to look at both versions to help your understanding of this extremely important concept. Recall that convolution is a very powerful tool in determining a system's output from knowledge of an arbitrary input and the system's impulse response. It will also be helpful to see convolution graphically with your own eyes and to play around with it some, so experiment with the applets available on the internet. These resources will offer different approaches to this crucial concept.

Discrete Time Convolution

As mentioned above, the convolution sum provides a concise, mathematical way to express the output of an LTI system based on an arbitrary discrete-time input signal and the system's response. The **convolution sum** is expressed as

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

As with continuous-time, convolution is represented by the symbol *, and can be written as

$$y[n] = x[n] * h[n]$$

By making a simple change of variables into the convolution sum, $k = n - k$, we can easily show that convolution is **commutative**:

$$x[n] * h[n] = h[n] * x[n]$$

Discrete-Time Convolution

Let us call $x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$ the <u>convolution</u> of $x[n]$ and $h[n]$.

- The response of an LTI system to $x[n]$ is given by $y[n] = x[n] * h[n]$, where $h[n]$ is the impulse response of the system.

Discrete-Time Convolution

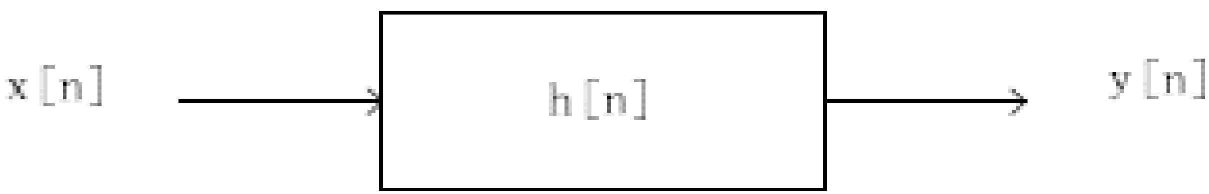
Let us consider $x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$

Let $n - k = m$, then $k = n - m$

$$\sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{m=-\infty}^{\infty} x[n-m]h[m] = h[n] * x[n]$$

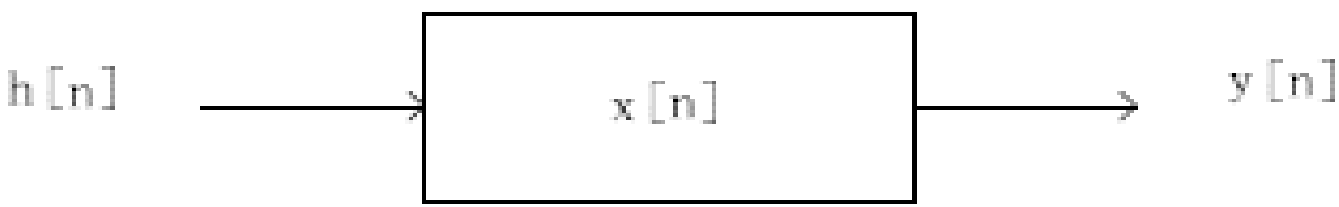
so convolution is <u>commutative</u>.

Notation:



Discrete-Time Convolution

This means that the box is an LTI system with impulse response $h[n]$. In the other case:

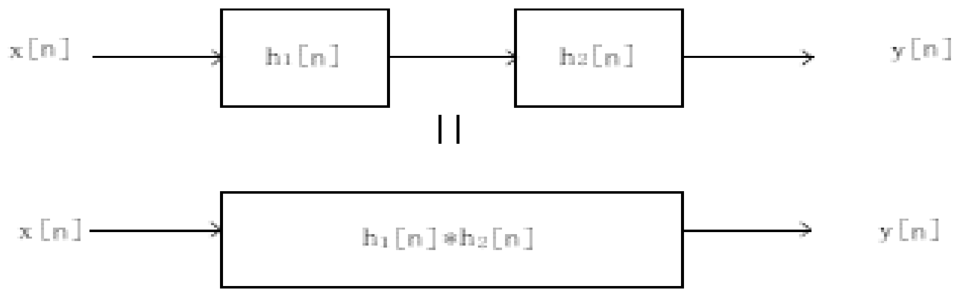


It can be shown that the convolution operation is associative and distributive.

Associativity:

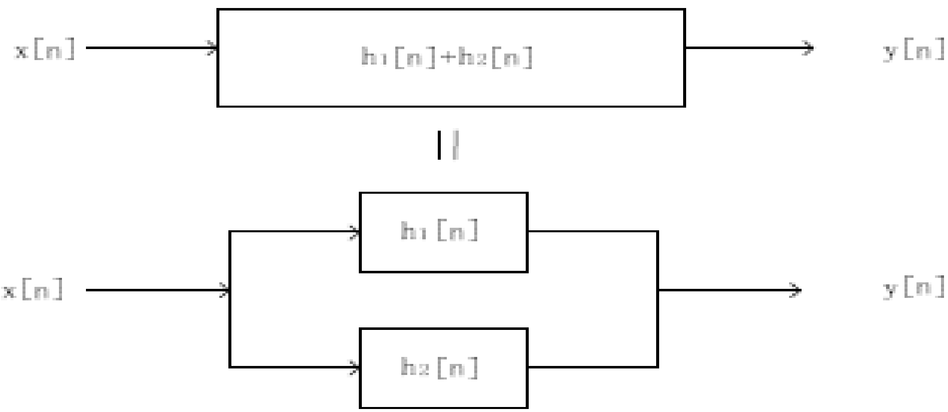
$$(x[n] * h_1[n]) * h_2[n] = x[n] * (h_1[n] * h_2[n])$$

Discrete-Time Convolution

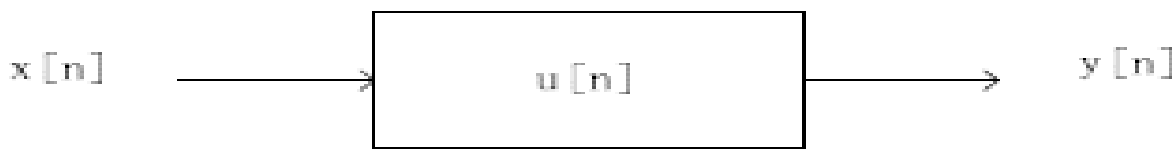


Distributivity:

$$x[n] * (h_1[n] + h_2[n]) = x[n] * h_1[n] + x[n] * h_2[n]$$

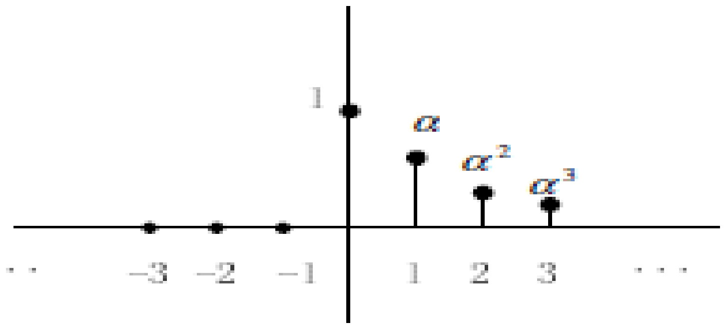


Example:

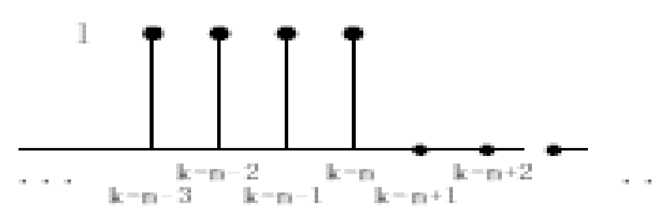


$$h[n] = u[n], \text{ and } x[n] = \alpha^n u[n], \ |\alpha| < 1, \text{ what is } y[n]?$$

$x[n] = \alpha^n u[n]$



$h[n-k] = u[n-k]$



Discrete-Time Convolution: Example 1

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

If $n < 0$, then $x[k]h[n-k] = 0$, so $x[n] * h[n] = 0$

If $n \geq 0$, then $x[k]h[n-k] = \begin{cases} \alpha^k, & 0 \leq k \leq n \\ 0, & otherwise \end{cases}$

so $x[n] * h[n] = \sum_{k=0}^{n} \alpha^k = \frac{1-\alpha^{n+1}}{1-\alpha}$

Discrete-Time Convolution: Example 1

Mathematically,

$$\begin{aligned} x[n] * h[n] &= \sum_{k=-\infty}^{\infty} x[k]h[n-k] \\ &= \sum_{k=-\infty}^{\infty} \alpha^k u[k]h[n-k] \\ &= \sum_{k=0}^{n} \alpha^k u[k]h[n-k] \\ &= \begin{cases} \sum_{k=0}^{n} \alpha^k & n \geq 0 \\ 0 & n < 0 \end{cases} \end{aligned}$$

Discrete-Time Convolution: Example 2

Consider a system with an impulse response of

$$h(n) = [\ 1\ 1\ 1\ 1\ ]$$

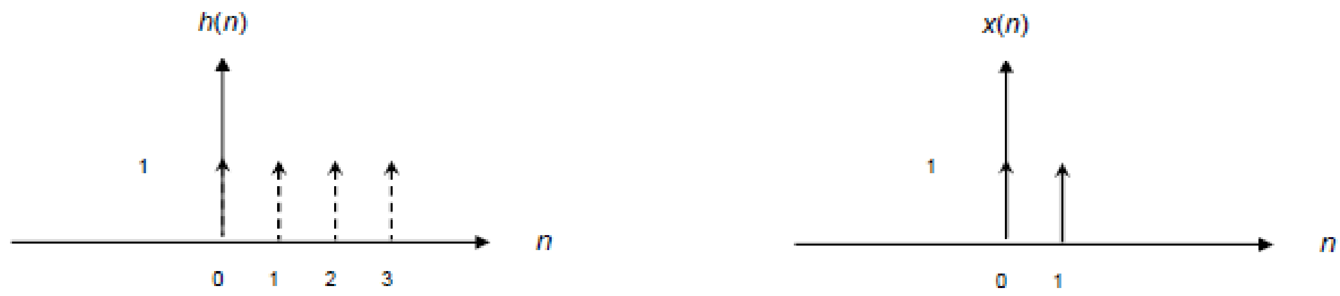
If the input to the signal is

$$x(n) = [\ 1\ 1\ ]$$

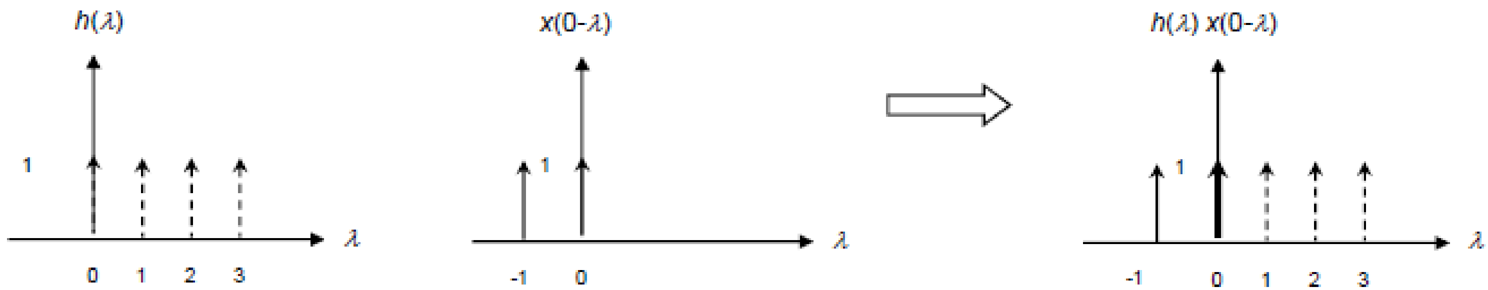
- Thus, the output of the system is $y(n) = \sum_{\lambda=-\infty}^{\infty} h(\lambda)x(n-\lambda)$

- The result of the convolution procedure in its graphical form is :

Discrete-Time Convolution: Example 2

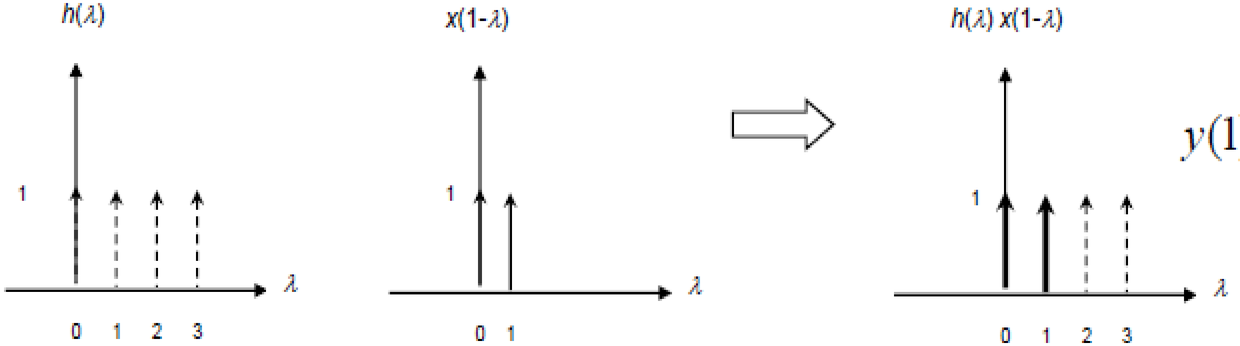


i) The definition of the system impulse response $h(n)$ and the input signal $x(n)$



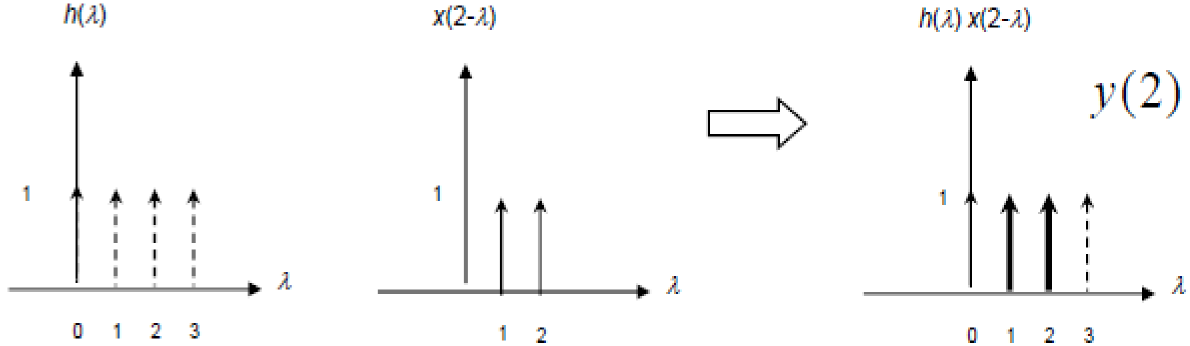
$$y(0) = \sum_{\lambda=-\infty}^{\infty} h(\lambda)x(0-\lambda) = 1$$

ii) The result at n=1.



$$y(1) = \sum_{\lambda=-\infty}^{\infty} h(\lambda)x(1-\lambda) = 2$$

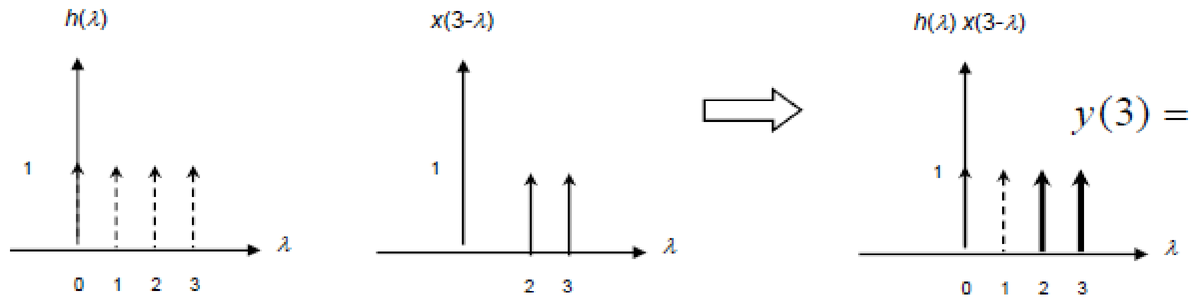
iii) The result at n=2.



$$y(2) = \sum_{\lambda=-\infty}^{\infty} h(\lambda)x(2-\lambda) = 2$$

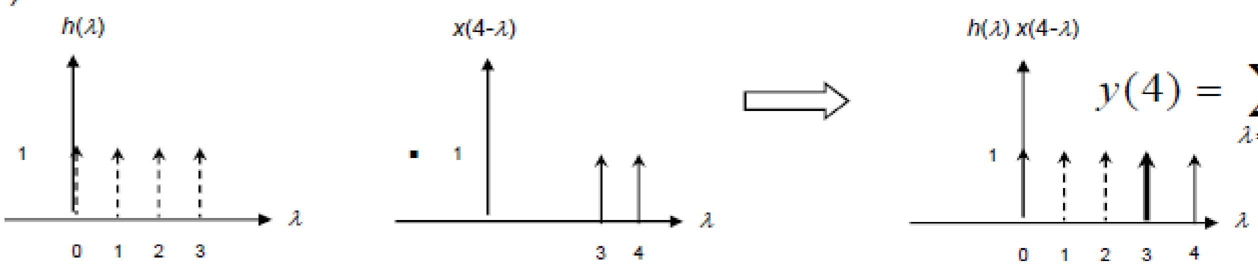
Discrete-Time Convolution: Example 2

iV) At n=3



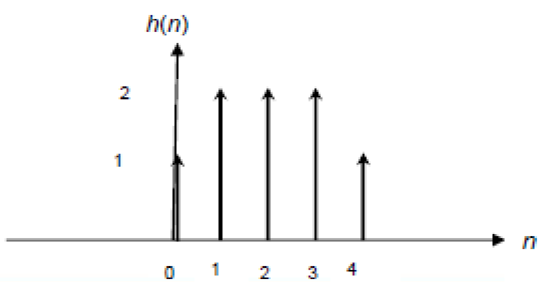
$$y(3) = \sum_{\lambda=-\infty}^{\infty} h(\lambda)x(3-\lambda) = 2$$

v) At n=4



$$y(4) = \sum_{\lambda=-\infty}^{\infty} h(\lambda)x(4-\lambda) = 1$$

Finally



Performance of Convolution

- Convolution can be performed in numerous ways. Some of those are:
  - Direct-evaluation
  - Graphical method
  - Slide-rule method
  - Fourier transform, and
  - Z-transform

Deconvolution

- Unwanted convolution is an inherent problem in transferring analog information. For instance, all of the following can be modelled as a convolution: image blurring in a shaky camera, echoes in long distance telephone calls, the finite bandwidth of analog sensors and electronics, etc. Deconvolution is the process of filtering a signal to compensate for an undesired convolution.

- The goal of deconvolution is to recreate the signal as it existed *before* the convolution took place. This usually requires the characteristics of the convolution (i.e., the impulse or frequency response) to be known. This can be distinguished from **blind deconvolution**, where the characteristics of the parasitic convolution are *not* known. Blind deconvolution is a much more difficult problem that has no general solution, and the approach must be tailored to the particular application.

Deconvolution

Deconvolution is nearly impossible to understand in the *time domain*, but quite straightforward in the *frequency domain*. Each sinusoid that composes the original signal can be changed in amplitude and/or phase as it passes through the undesired convolution. To extract the original signal, the deconvolution filter must *undo* these amplitude and phase changes.

For example, if the convolution changes a sinusoid's amplitude by 0.5 with a 30 degree phase shift, the deconvolution filter must amplify the sinusoid by 2.0 with a -30 degree phase change.

Deconvolution

Reverse of Convolution

$$\boxed{x_t = w_t * e_t} \rightarrow \boxed{e_t = x_t * w^{-1}_t}$$

=> Inverse Filtering

- Aim of Deconvolution
  1. Theoretical: Reconstruction of the Reflectivity function
  2. Practical:
  - Shorting of the Signal
  - Suppression of Noise
  - Suppression of Multiples

Deconvolution

In mathematics, **deconvolution** is an algorithm-based process used to reverse the effects of convolution on recorded data. The concept of deconvolution is widely used in the techniques of signal processing and image processing. Because these techniques are in turn widely used in many scientific and engineering disciplines, deconvolution finds many applications.

In general, the object of deconvolution is to find the solution of a convolution equation of the form:

$$f * g = h$$

Usually, h is some recorded signal, and f is some signal that we wish to recover, but has been convolved with some other signal g before we recorded it. The function g might represent the transfer function of an instrument or a driving force that was applied to a physical system. If we know g, or at least know the form of g, then we can perform deterministic deconvolution. However, if we do not know g in advance, then we need to estimate it. This is most often done using methods of statistical estimation.

In physical measurements, the situation is usually closer to

$$(f * g) + \varepsilon = h$$

In this case ε is noise that has entered our recorded signal. If we assume that a noisy signal or image is noiseless when we try to make a statistical estimate of g, our estimate will be incorrect. In turn, our estimate of f will also be incorrect. The lower the signal-to-noise ratio, the worse our estimate of the deconvolved signal will be. That is the reason why inverse filtering the signal is usually not a good solution. However, if we have at least some knowledge of the type of noise in the data (for example, white noise), we may be able to improve the estimate of f through techniques such as Wiener deconvolution.

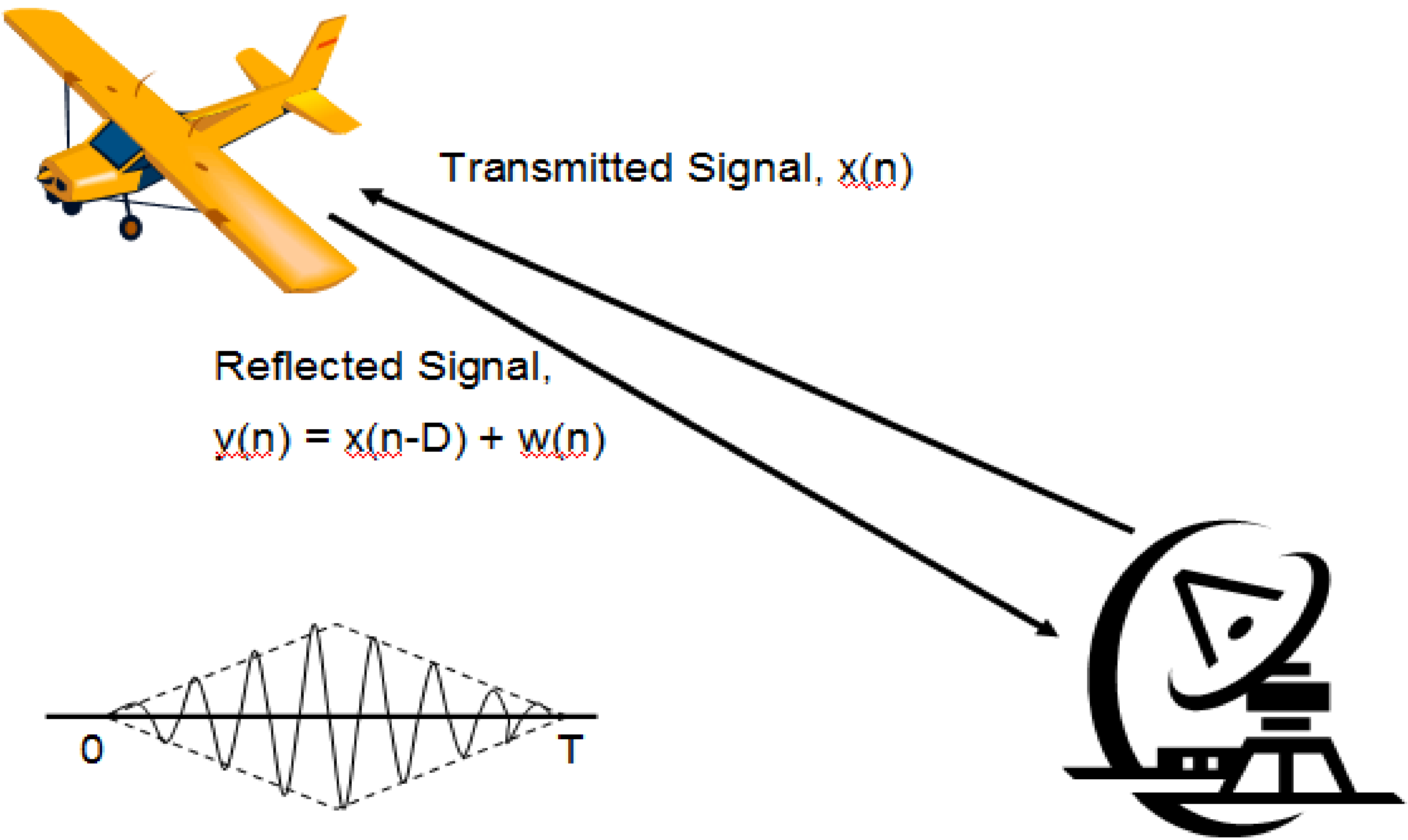
The foundations for deconvolution and time-series analysis were largely laid by Norbert Wiener of the Massachusetts Institute of Technology in his book *Extrapolation, Interpolation, and Smoothing of Stationary Time Series* (1949).[2] The book was based on work Wiener had done during World War II but that had been classified at the time. Some of the early attempts to apply these theories were in the fields of weather forecasting and economics.

Deconvolution

Deconvolution is a key area in signal and image processing. It is used for objectives in signal and image processing that include the following:

1. deblurring,
2. removal of atmospheric seeing degradation,
3. correction of mirror spherical aberration,
4. image sharpening,
5. mapping detector response characteristics to those of another,
6. image or signal zooming, and
7. optimizing display.

Correlation



Correlation

- A mathematical operation that closely resembles convolution is correlation.
- Just like in convolution, two signal sequences are involved in correlation.
- Correlation is a measure of the similarity between two signals as a function of time shift between them.
- Correlation is maximum when two signals are similar in shape, and are in phase (or 'unshifted' with respect to each other).
- Correlation is often encountered in Radar, Sonar, Digital communications etc
- Correlating two different signals is called Cross-correlation.
- Correlating a signal with itself is called autocorrelation.

Cross-Correlation

- Cross correlation can be used to identify a signal by comparison with a library of known reference signals.

<u>Definition: Crosscorrelation</u>

The crosscorrelation between two signals x[n] and y[n] is given by:

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x[n]y[n-l] \qquad (1)$$

where the time shift l is called the <u>lag</u>.

OR

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x[n+l]y[n] \qquad (2)$$

Cross-Correlation

- If we reverse the roles of x[n] and y[n] in (1) and (2) and hence reverse the order of indices xy, we obtain the cross correlation sequence

$$r_{yx}(l) = \sum_{n=-\infty}^{\infty} y[n]x[n-l] \qquad (3)$$

Or, equivalently

$$r_{yx}(l) = \sum_{n=-\infty}^{\infty} y[n+1]x[n] \qquad (4)$$

By comparing (1) and (4) or (2) and (3), we conclude that

$$r_{xy}(l) = r_{yx}(-l) \qquad (5)$$

Therefore, $r_{yx}[l]$ is simply the folded version of $r_{xy}[l]$.

Cross-Correlation

- **Example:** Determine the crosscorrelation sequence of the sequences

  $x[n] = \{\ldots, 0, 0, 2, -1, 3, 7, \underline{1}, 2, -3, 0, 0, \ldots\}$

  $y[n] = \{\ldots, 0, 0, 1, -1, 2, -2, \underline{4}, 1, -2, 5, 0, 0, \ldots\}$

Solution: The only difference in convolution and crosscorrelation is that in crosscorrelation we don't need to fold the sequence. Otherwise all steps are same.

Cross-Correlation

X[n]				**2**	**-1**	**3**	**7**	**1**	**2**	**-3**					
y[n+3]	1	-1	2	-2	4	1	-2	5							$R_{xy}(-3)=-14$
y[n+2]		1	-1	2	-2	4	1	-2	5						$R_{xy}(-2)=33$
Y[n+1]			1	-1	2	-2	4	1	-2	5					$R_{xy}(-1)=0$
Y[n]				**1**	**-1**	**2**	**-2**	**4**	**1**	**-2**	**5**				$R_{xy}(0)=7$
Y[n-1]					1	-1	2	-2	4	1	-2	5			$R_{xy}(1)=13$
Y[n-2]						1	-1	2	-2	4	1	-2	5		$R_{xy}(2)=-18$
Y[n-3]							1	-1	2	-2	4	1	-2	5	$R_{xy}(3)=16$

$$R_{xy}(l)=\{10, -9, 19, 36, -14, 33, 0, \underline{7}, 13, -18, 16, -7, 5, -3, 0\}$$

Auto-Correlation

- Autocorrelation is the special case of crosscorrelation, in which one signal is compared with its time shifted version.

Definition: Autocorrelation

The autocorrelation of a real signal x[n] is given by:

$$r_{xx}(l) = \sum_{n=-\infty}^{\infty} x[n]x[n-l]$$

where the time shift m is called the lag.

Or equivalently as, $r_{xx}(l) = \sum_{n=-\infty}^{\infty} x[n+l]x[n]$

Properties of Auto-correlation and Cross-correlation Sequences

Let us assume that we have two sequences x[n] and y[n] with finite energy from which we form the linear combination

$$a\,x[n] + b\,y[n-l]$$

where a and b are arbitrary constants and k is some time shift. The energy in this signal is

$$\sum_{n=-\infty}^{\infty}\left[ax[n]+by[n-l]\right]^2 = a^2\sum_{n=-\infty}^{\infty}x^2[n]+b^2\sum_{n=-\infty}^{\infty}y^2[n-l]$$

$$+2ab\sum_{n=-\infty}^{\infty}x[n]y[n-l] = a^2 r_{xx}(0)+b^2 r_{yy}(0)+2ab\,r_{xy}(l)$$

Note that $r_{xx}(0) = E_x$ and $r_{yy}(0) = E_y$, the energies of x[n] and y[n] respectively.

It is obvious that

$$a^2 r_{xx}(0)+b^2 r_{yy}(0)+2ab\,r_{xy}(k)\geq 0$$

Now, assuming that $b \neq 0$, we divide the above equation by b^2 to obtain

$$r_{xx}(0)\left(\frac{a}{b}\right)^2 + 2r_{xy}[l]\left(\frac{a}{b}\right) + r_{yy}(0) \geq 0 \qquad (3)$$

This is a quadratic equation. Since the quadratic is non-negative, its discriminant must be non-positive. That is,

$$4\left[r_{xy}^2[l] - r_{xx}(0)\,r_{yy}(0)\right] \leq 0 \qquad (4)$$

Therefore, the crosscorrelation sequence satisfies the condition that

$$\left|r_{xy}(l)\right| \leq \sqrt{r_{xx}(0)\,r_{yy}(0)} = \sqrt{E_x E_y} \qquad (5)$$

In the special case i-e in Autocorrelation where y[n] = x[n], (5) reduces to

$$\left|r_{xx}(l)\right| \leq r_{xx}(0) = E_x \qquad (6)$$

- This means that the autocorrelation sequence of a signal attains its maximum value at zero lag.
- If any one or both of the signals involved are scaled, the shape of the cross correlation sequence does not change; only the amplitudes of the crosscorrelation sequence are scaled accordingly.

It is often desirable in practice to normalize the autocorrelation and crosscorrelation sequences to the range from -1 to 1. The normalized autocorrelation sequence is defined as,

$$\rho_{xx}(l) = \frac{r_{xx}(l)}{r_{xx}(0)} \qquad (7)$$

Similarly, we define the normalized crosscorrelation sequence

$$\rho_{xy}(l) = \frac{r_{xy}(l)}{\sqrt{r_{xx}(0) r_{yy}(0)}} \qquad (8)$$

Properties of Auto-correlation and Cross-correlation Sequences

- One other important property of autocorrelation is that its an even function.
  - We know that $r_{xy}(l)=r_{yx}(-l)$, so
  - If we make x[n]=y[n] (autocorrelation) the condition becomes $r_{xx}(l)=r_{xx}(-l)$, Hence the autocorr: function is an even function.

Difference between Convolution and Correlation

- Convolution is usually between a signal and a filter; we think of it as a system with a single input and stored coefficients.
- Crosscorrelation is usually between two signals; we think of a system with two inputs and no stored coefficients.

جامعة ديالى
كلية الهندسة

www.engineering.uodiyala.edu.iq

University Of Diyala – College Of Engineering

Digital Signal Processing

Course Instructor
Dr. Ali J. Abboud

Lecture No. 5: Fourier Transforms

Third Class
Department of Computer and Software Engineering

http://www.engineering.uodiyala.edu.iq/
https://www.facebook.com/Engineer.College1

Lecture Outline

- Continuous Time Fourier Series
- Discrete time Fourier series
- Discrete Fourier Transform (DFT)
- Fast Fourier Transform (FFT)
- Decimation in time Fast Fourier Transform
- Decimation in frequency Fast Fourier Transform

Fourier Series Representation

Fourier Series:

- Fourier series allows any periodic waveform in time to be decomposed into a sum of sine and cosine waveforms. The first requirement in realising the FS is to calculate the fundamental period, T, which is the shortest time over which the signal repeats.
- For a periodic signal with fundamental period T sec, the FS represents this signal as a sum of sine and cosine components that are harmonics of the fundamental frequency $f_0 = 1/T$ Hz.

Fourier Series Representation

- The Fourier series can be written in a number of different ways:

$$x(t) = \sum_{n=0}^{\infty} A_n \cos\left(\frac{2\pi nt}{T}\right) + \sum_{n=1}^{\infty} B_n \sin\left(\frac{2\pi nt}{T}\right)$$

$$= A_0 + \sum_{n=1}^{\infty}\left[A_n \cos\left(\frac{2\pi nt}{T}\right) + B_n \sin\left(\frac{2\pi nt}{T}\right)\right]$$

$$= A_0 + \sum_{n=1}^{\infty}\left[A_n \cos(2\pi n f_0 t) + B_n \sin(2\pi n f_0 t)\right]$$

$$= A_0 + \sum_{n=1}^{\infty}\left[A_n \cos(n\omega_0 t) + B_n \sin(n\omega_0 t)\right]$$

$$= \sum_{n=0}^{\infty}\left[A_n \cos(n\omega_0 t) + B_n \sin(n\omega_0 t)\right] \qquad (1)$$

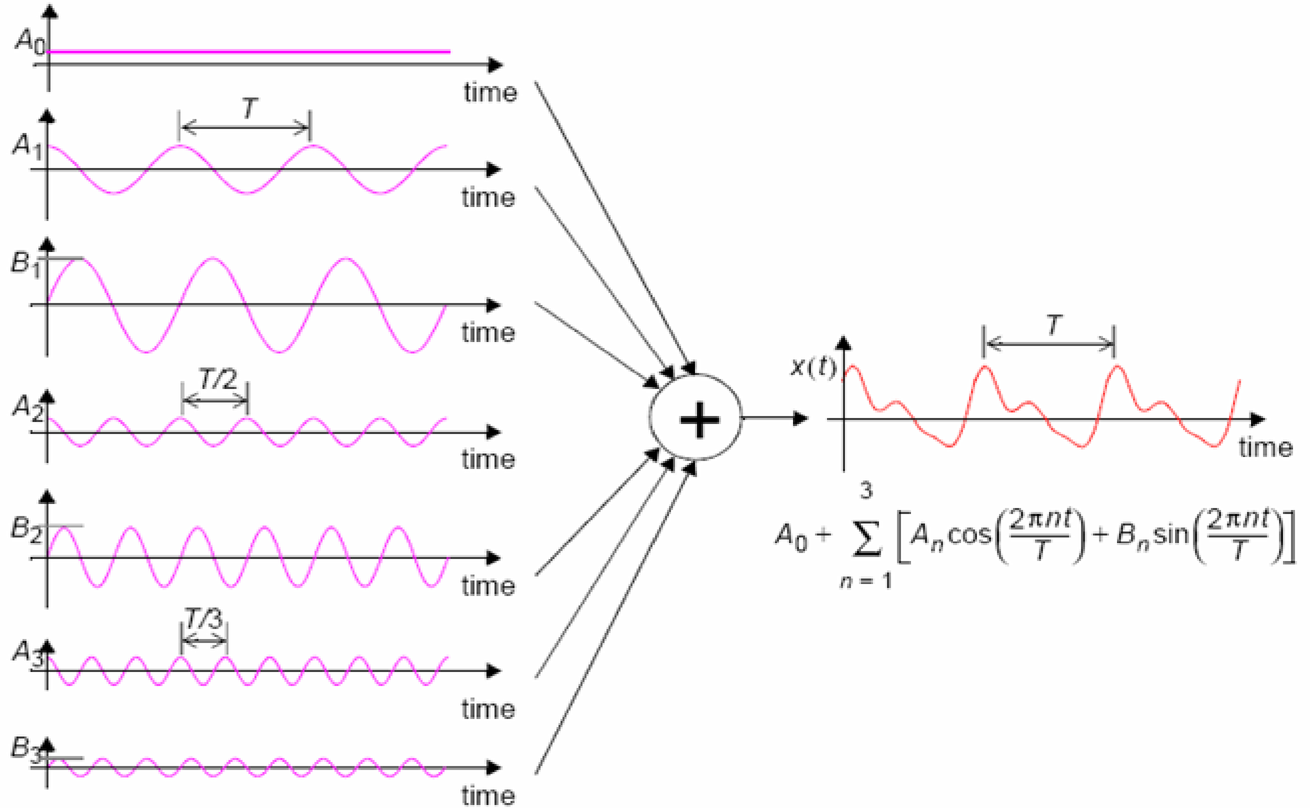
$$= A_0 + A_1 \cos(\omega_0 t) + A_2 \cos(2\omega_0 t) + A_3 \cos(3\omega_0 t) + \ldots$$
$$B_1 \sin(\omega_0 t) + B_2 \sin(2\omega_0 t) + B_3 \sin(3\omega_0 t) + \ldots$$

Where A_n and B_n are the amplitudes of the cos and sin waveforms, $\omega_0 = 2\pi f_0$ rad /sec is angular frequency.

Fourier Series Representation

- In more descriptive language, the above Fourier Series says that any periodic signal can be reproduced by adding a (possibly infinite!) series of harmonically related sinusoidal waveforms of amplitudes A_n or B_n.
- Therefore, *if a periodic signal with a fundamental period of say 0.01 sec is identified, then the Fourier Series will allow this waveform to be represented as a sum of various cosine and sine waves at frequencies of 100 Hz (fundamental frequency), 200 Hz, 300 Hz (Harmonics) and so on. The amplitudes of these components are given by A0, A_1, B_1, A_2, B_2 … and so on.*
- So, how are the values of A_n and B_n calculated??

Fourier Series Representation



Fourier series for a periodic signal $x(t)$. If we analyse a periodic signal and realizethe cosine and sine wave Fourier coefficients of appropriate amplitudes A_n and B_n, then summing these components will lead to exactly the original signal.

Fourier Series Representation

For that, we multiply both sides of (1) by cos ($p\omega_0 t$) where p is any arbitrary positive integer, then we get:

$$\cos(p\omega_0 t)x(t) = \cos(p\omega_0 t)\sum_{n=0}^{\infty}\left[A_n \cos(n\omega_0 t) + B_n \sin(n\omega_0 t)\right] \qquad (2)$$

Integrating Eq: (2) over one period, T, we get:

$$\int_0^T \cos(p\omega_0 t)x(t)dt = \int_0^T \left\{\cos(p\omega_0 t)\sum_{n=0}^{\infty}\left[A_n \cos(n\omega_0 t) + B_n \sin(n\omega_0 t)\right]\right\}dt$$

$$= \boxed{\sum_{n=0}^{\infty}\int_0^T \{A_n \cos(p\omega_0 t)\cos(n\omega_0 t)\}dt} + \boxed{\sum_{n=0}^{\infty}\int_0^T \{B_n \cos(p\omega_0 t)\sin(n\omega_0 t)\}dt} \qquad (3)$$

Using the trigonometric identity 2cosAsinB=sin (A+B)-sin (A-B), and sin($2\pi t/T$) = 0, note that the second term in the equation (3) is equal to zero, i.e.,

$$\sum_{n=0}^{\infty}\int_0^T \{B_n \cos(p\omega_0 t)\sin(n\omega_0 t)\}dt = \frac{B_n}{2}\int_0^T \{\sin(p+n)\omega_0 t - \sin(p-n)\omega_0 t\}dt$$

$$= \frac{B_n}{2}\int_0^T \sin\left[\frac{(p+n)2\pi t}{T}\right]dt - \frac{B_n}{2}\int_0^T \sin\left[\frac{(p-n)2\pi t}{T}\right]dt = 0 \qquad (4)$$

Fourier Series Representation

Eq: (4) is true for all positive integers of p and n.

Using trigonometric identity 2cosAcosB=cos (A+B)+ cos (A-B), we find that the first term of Eq: (3) is only equal to zero when p ≠n, i.e.,

$$\int_0^T \{A_n \cos(p\omega_0 t)\cos(n\omega_0 t)\}dt = \frac{A_n}{2}\int_0^T \{\cos(p+n)\omega_0 t + \cos(p-n)\omega_0 t\}dt = 0 \qquad \textbf{(5)}$$

If p=n, then

$$\int_0^T \{A_n \cos(p\omega_0 t)\cos(n\omega_0 t)\}dt = A_n\int_0^T \cos^2(n\omega_0 t)dt$$

$$= \frac{A_n}{2}\int_0^T (1+\cos 2n\omega_0 t)dt = \frac{A_n}{2}\int_0^T 1.dt = \frac{A_n}{2}t\Big|_0^T = \frac{A_n T}{2} \qquad \textbf{(6)}$$

Therefore, using Eq: (6), (5), (4), and (3), we note that:

$$\int_0^T \{\cos(p\omega_0 t)x(t)\}dt = \frac{A_n T}{2},$$

and therefore, since p=n,

$$A_n = \frac{2}{T}\int_0^T \{\cos(n\omega_0 t)x(t)\}dt \qquad \textbf{(7)}$$

Fourier Series Representation

By multiplying Eq: (3) by sin $(p\omega_0 t)$ and using a similar set of simplifications we can show that:

$$B_n = \frac{2}{T}\int_0^T \{x(t)\sin(n\omega_0 t)\}dt \qquad (8)$$

Hence, the three key equations for calculating the Fourier Series of a periodic signal with fundamental period T are:

$$x(t) = \sum_{n=0}^{\infty} A_n \cos\left(\frac{2\pi nt}{T}\right) + \sum_{n=1}^{\infty} B_n \sin\left(\frac{2\pi nt}{T}\right)$$

$$A_n = \frac{2}{T}\int_0^T \{x(t)\cos(n\omega_0 t)\}dt$$

$$B_n = \frac{2}{T}\int_0^T \{x(t)\sin(n\omega_0 t)\}dt$$

Complex Fourier Series

From Euler's theorem, note that:

$$e^{j\omega} = \cos(\omega) + j\sin(\omega) \qquad \cos(\omega) = \left(e^{j\omega} + e^{-j\omega}\right)/2 \qquad \sin(\omega) = \left(e^{j\omega} - e^{-j\omega}\right)/2j$$

Substituting these values in Eq: (1), and rearranging gives:

$$\begin{aligned} x(t) &= A_0 + \sum_{n=1}^{\infty}\left[A_n \cos\left(\frac{2\pi nt}{T}\right) + B_n \sin\left(\frac{2\pi nt}{T}\right)\right] \\ &= A_0 + \sum_{n=1}^{\infty}\left[A_n\left(\frac{e^{jn\omega_0 t} + e^{-jn\omega_0 t}}{2}\right) + B_n\left(\frac{e^{jn\omega_0 t} - e^{-jn\omega_0 t}}{2j}\right)\right] \\ &= A_0 + \sum_{n=1}^{\infty}\left[\left(\frac{A_n}{2} + \frac{B_n}{2j}\right)e^{jn\omega_0 t} + \left(\frac{A_n}{2} - \frac{B_n}{2j}\right)e^{-jn\omega_0 t}\right] \\ &= A_0 + \sum_{n=1}^{\infty}\left(\frac{A_n - jB_n}{2}\right)e^{jn\omega_0 t} + \sum_{n=1}^{\infty}\left(\frac{A_n + jB_n}{2}\right)e^{-jn\omega_0 t} \qquad (9) \end{aligned}$$

For the second summation term, if the sign of the complex sinusoid is negated and the summation limits are reversed, then we can rewrite as:

$$x(t) = A_0 + \sum_{n=1}^{\infty}\left(\frac{A_n - jB_n}{2}\right)e^{jn\omega_0 t} + \sum_{n=-\infty}^{-1}\left(\frac{A_n + jB_n}{2}\right)e^{jn\omega_0 t} = \sum_{n=-\infty}^{\infty} C_n e^{jn\omega_0 t} \qquad (10)$$

Complex Fourier Series

where Cn in terms of the Fourier series coefficients of Eq. 7 and 8 gives:

$$
\begin{aligned}
&C_0 = A_0 \\
&C_n = (A_n - jB_n)/2 \qquad \text{for n} > 0 \\
&C_n = (A_n + jB_n)/2 \qquad \text{for n} < 0
\end{aligned}
\qquad \textbf{(11)}
$$

From Eq. 11 note that for n > 0,

$$C_n = \frac{A_n - jB_n}{2} = \frac{1}{T}\int_0^T x(t)\cos(n\omega_0 t)dt - j\frac{1}{T}\int_0^T x(t)\sin(n\omega_0 t)dt$$

$$= \frac{1}{T}\int_0^T x(t)[\cos(n\omega_0 t) - j\sin(n\omega_0 t)]dt = \frac{1}{T}\int_0^T x(t)e^{-jn\omega_0 t}dt \qquad \textbf{(12)}$$

For n < 0, it is clear from Eq. 11 that, $C_n = C^*_{-n}$, where '*' denotes complex conjugate. Therefore, the two important equation for complex exponential Fourier series are

$$\boxed{\begin{aligned} x(t) &= \sum_{n=-\infty}^{\infty} C_n e^{jn\omega_0 t} \\ C_n &= \frac{1}{T}\int_0^T x(t)e^{-jn\omega_0 t}dt \end{aligned}}$$

Example

The ease of working with complex exponentials can be illustrated by this simple example.

Example 1: Simplify the following equations in to a sum of sine waves:

$$\sin(\omega_1 t)\sin(\omega_2 t)$$

This requires the recollection (or rederivation!) of trigonometric identities to yield:

$$\sin(\omega_1 t)\sin(\omega_2 t) = \frac{1}{2}\cos(\omega_1 - \omega_2)t + \frac{1}{2}\cos(\omega_1 + \omega_2)t$$

However, it is relatively easier to simplify the following expression to a sum of complex exponentials:

$$e^{j\omega_1 t} e^{j\omega_2 t} = e^{j(\omega_1 + \omega_2)t}$$

Although, seemingly a simple comment this is the basis of using complex exponentials rather than sines and cosines; they make the maths easier. Of course, in situations where the signal being analysed is complex, then the complex Fourier series *must* be used!

Fourier Transform

- The Fourier Series allows a *periodic* signal to be broken down into a sum of sin and cos components.
- However, most practical signals are aperiodic!
- Therefore, the Fourier Transform was derived in order to analyse the frequency content of *aperiodic* signals.

Discrete Fourier Transform

The DT Fourier Transform (DTFT) of a finite energy discrete time signal x[n] is defined as:

$$X(\omega) = X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}, \qquad \omega \in [-\pi, \pi]$$

$X(\omega)$ may be regarded as a decomposition of x[n] into its *frequency components*.

- It is not difficult to verify that $X(\omega)$ is periodic with frequency 2π.

The Inverse Fourier Transform of $X(\omega)$ may be defined as:

$$x[n] = \frac{1}{2\pi}\int_{2\pi} X(\omega)e^{j\omega n}\,d\omega$$

Discrete Fourier Transform

- Notation: $x[n] \leftrightarrow X(\omega)$, $x[n]=F^{-1}(X(\omega))$, $X(\omega)=F^{-1}(x[n])$
- Signal has a transform if it satisfies Dirichlet conditions.
- $X(\omega)$ is called the spectrum of x[n]:

$$X(\omega)=|X(\omega)|e^{j\angle X(\omega)} \Rightarrow \begin{cases} |X(\omega)|= & \text{magnitude spectrum,} \\ \angle X(\omega)= & \text{phase spectrum,} \end{cases}$$

The magnitude spectrum is often expressed in deceibels (dB)

- DTFT describes the frequency content of x[n]
- For real signals
  - $|X(\omega)|=|X(-\omega)| \rightarrow$ Even function, and
  - phase $\angle X(-\omega)=-\angle X(\omega) \rightarrow$ Odd function.

$$\boxed{\begin{aligned} X(\omega) &= \sum_{n=-\infty}^{\infty} x[n]e^{-jwn} \\ x[n] &= \frac{1}{2\pi}\int_{2\pi} X(\omega)e^{j\omega n}d\omega \end{aligned}}$$

Energy Density Spectrum of Aperiodic Signals

Energy of a discrete time signal x[n] is defined as:

$$E_x = \sum_{n=-\infty}^{\infty} |x[n]|^2$$

Let us now express the energy Ex in terms of the spectral characteristic X(w). First we have

$$E_x = \sum_{n=-\infty}^{\infty} x[n]x^*[n] = \sum_{n=-\infty}^{\infty} x[n]\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} X^*(\omega)e^{-j\omega n}d\omega\right]$$

If we interchange the order of integration and summation in the above equation, we obtain

$$E_x = \frac{1}{2\pi}\int_{-\pi}^{\pi} X^*(\omega)\left[\sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}\right]d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(\omega)|^2 d\omega$$

Therefore, the energy relation between x[n] and X(w) is

$$\boxed{E_x = \sum_{n=-\infty}^{\infty} |x[n]|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(\omega)|^2 d\omega}$$

Parseval's relation for DT Aperiodic signals

Energy Density Spectrum of Aperiodic Signals

- The spectrum is, in general, a complex valued function of frequency.
- The quantity $S_{xx}(w)=|X(w)|^2$ represents the distribution of energy as a function of frequency and it is called Energy Density Spectrum of x(n).
- $S_{xx}(w)$ does not contain any phase information.

Energy Density Spectrum of Aperiodic Signals

Example - 1: Determine DTFT and sketch the energy density spectrum $S_{xx}(w)$ of the sequence:

$$x[n]=\alpha^n u[n] \qquad |\alpha|<1$$

Solution- 1:

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

$$X(\omega)=\sum_{n=0}^{\infty} \alpha^n e^{-j\omega n}=\sum_{n=0}^{\infty}\left(\alpha e^{-j\omega}\right)^n$$

Using the geometric sequence, provided $|\alpha|<1$, this sum is:

$$\boxed{X(\omega)=\frac{1}{1-\alpha e^{-j\omega}}}$$

Energy Density Spectrum of Aperiodic Signals

Energy Density Spectrum is given by

$$S_{xx}(\omega) = |X(\omega)|^2 = X(\omega)X^*(\omega)$$

$$S_{xx}(\omega) = \frac{1}{(1 - ae^{-j\omega})}\frac{1}{(1 - ae^{j\omega})}$$

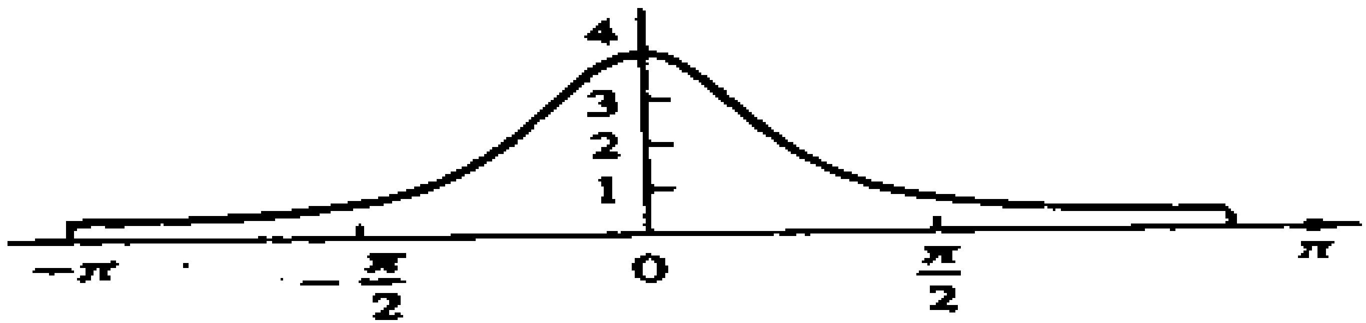
$$S_{xx}(\omega) = \frac{1}{1 - a(e^{jw} + e^{-jw}) + a^2}$$

$$S_{xx}(\omega) = \frac{1}{1 - 2a\cos\omega + a^2}$$

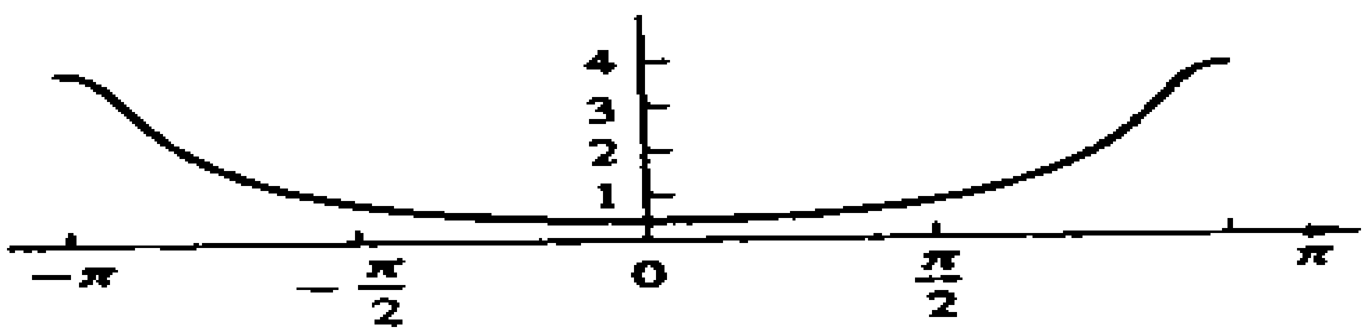
Figure on next slide shows x(n) and its corresponding spectrum for a=0.5 & a=-0.5

Energy Density Spectrum of Aperiodic Signals

$$S_{xx}(\omega) = \frac{1}{1 - 2a\cos\omega + a^2}, \quad a = 0.5$$



$$S_{xx}(\omega) = \frac{1}{1 - 2a\cos\omega + a^2}, \quad a = -0.5$$



Energy Density Spectrum of Aperiodic Signals

Example – 2: Determine the Fourier Transform and the energy density spectrum of the sequence

$$x[n] = \begin{cases} A, & 0 \le n \le L-1 \\ 0, & otherwise \end{cases}$$

Solution – 2:

$$X(w) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} = \sum_{0}^{L-1} Ae^{-j\omega n} = A\frac{1-e^{-j\omega L}}{1-e^{-j\omega}} = Ae^{-j(\omega/2)(L-1)}\frac{\sin(\omega L/2)}{\sin(\omega/2)}$$

The magnitude of x[n] is

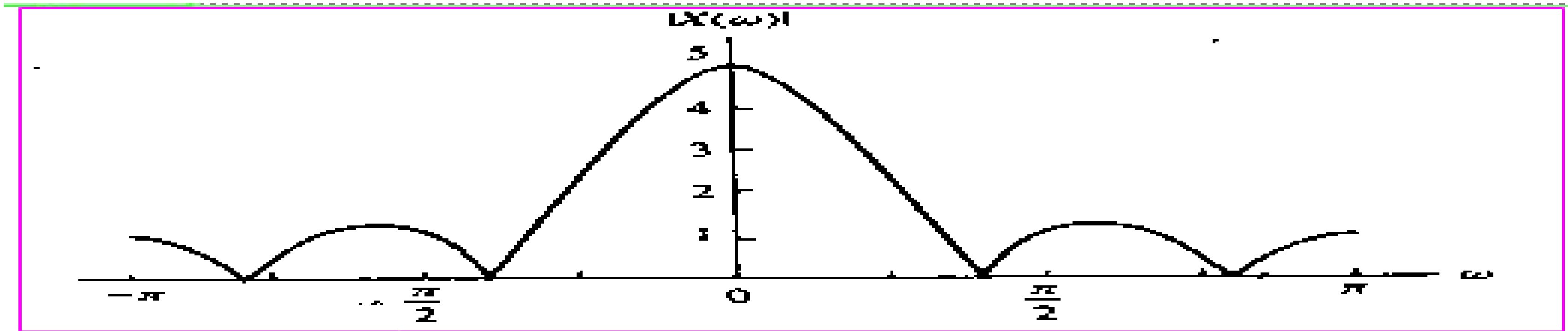
$$|X(\omega)| = \begin{cases} |A|L, & \omega = 0 \\ |A|\left|\dfrac{\sin(\omega L/2)}{\sin(\omega/2)}\right|, & \text{otherwise} \end{cases}$$

and the phase spectrum is

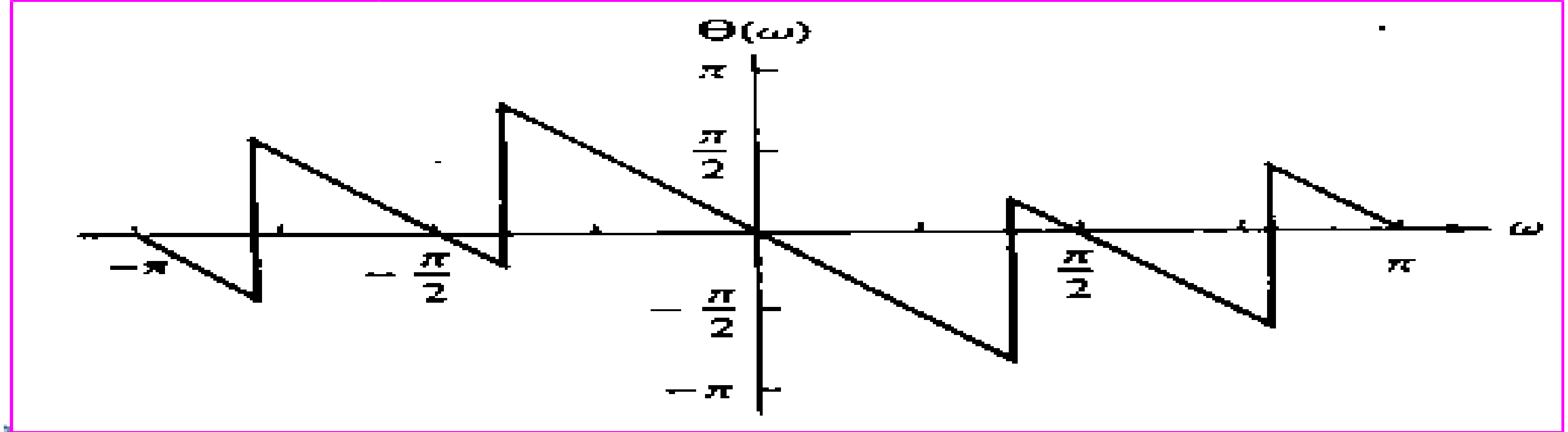
$$\angle X(\omega) = \angle A - \angle\frac{\omega}{2}(L-1) + \angle\frac{\sin(\omega L/2)}{\sin(\omega/2)}$$

The signal x[n] magnitude and phase is plotted on the next slide.

Energy Density Spectrum of Aperiodic Signals



Magnitude response



Phase response

Some Common DTFT

Sequence	Discrete-Time Fourier Transform
$\delta(n)$	1
$\delta(n-n_0)$	$e^{-jn_0\omega}$
1	$2\pi\delta(\omega)$
$e^{jn\omega_0}$	$2\pi\delta(\omega-\omega_0)$
$a^n u(n), \quad \lvert a\rvert < 1$	$\dfrac{1}{1-ae^{-j\omega}}$
$-a^n u(-n-1), \quad \lvert a\rvert > 1$	$\dfrac{1}{1-ae^{-j\omega}}$
$(n+1)a^n u(n), \quad \lvert a\rvert < 1$	$\dfrac{1}{(1-ae^{-j\omega})^2}$
$\cos n\omega_0$	$\pi\delta(\omega+\omega_0)+\pi\delta(\omega-\omega_0)$

Properties of DTFT

- A FT for Aperiodic finite energy DT signals described possesses a number of properties that are very useful in reducing the complexity of frequency analysis problems in many practical applications.
- For convenience, we adopt the notations

$$x[n] \stackrel{F}{\longleftrightarrow} X(\omega)$$

$$x[n]=F^{-1}\{X(\omega)\}$$

$$X(\omega)=F^{-1}(x[n])$$

Properties of DTFT

- **Symmetry:**
  - Real and even $x(n)$ $\rightarrow$ Real and Even $X(\omega)$
  - Real and odd $x(n)$ $\rightarrow$ Imaginary and odd $X(\omega)$
  - Imaginary and odd $x(n)$ $\rightarrow$ Real and odd $X(\omega)$
  - Imaginary and even $x(n)$ $\rightarrow$ Imaginary and even $X(\omega)$
- **Linearity:**
  - If $x_1[n] \xleftrightarrow{F} X_1(\omega)$

    $x_2[n] \xleftrightarrow{F} X_2(\omega)$

$$\boxed{a_1 x_1[n] + a_2 x_2[n] \xleftrightarrow{DTFT} a_1 X_1(\omega) + a_2 X_2(\omega)}$$

Properties of DTFT

Example – 3: Determine the DTFT of the signal $x[n] = a^{|n|}$

Solution – 3: First, we observe that $x[n]$ can be expressed as: $x[n]=x_1[n]+ x_2[n]$ (linearity prop:)

Where $x_1[n] = \begin{cases} a^n, & n \geq 0 \\ 0, & n < 0 \end{cases}$ and $x_2[n] = \begin{cases} a^{-n}, & n < 0 \\ 0, & n \geq 0 \end{cases}$

Now, $$X_1(\omega) = \sum_{n=-\infty}^{\infty} x_1[n]e^{-j\omega n} = \sum_{n=0}^{\infty} a^n e^{-j\omega n} = \sum_{n=0}^{\infty} \left(ae^{-j\omega}\right)^n$$

$$= 1 + ae^{-j\omega} + \left(ae^{-j\omega}\right)^2 + \left(ae^{-j\omega}\right)^3 + ... = \frac{1}{1 - ae^{-j\omega}}$$

and, $$X_2(\omega) = \sum_{n=-\infty}^{\infty} x_2[n]e^{-j\omega n} = \sum_{n=-\infty}^{-1} a^{-n} e^{-j\omega n} = \sum_{n=-\infty}^{-1} \left(ae^{j\omega}\right)^{-n}$$

$$= \sum_{k=1}^{\infty} \left(ae^{j\omega}\right)^k = ae^{j\omega} + (ae^{j\omega})^2 + ... = \frac{ae^{j\omega}}{1 - ae^{j\omega}}$$

$$X(\omega) = X_1(\omega) + X_2(\omega) = \frac{1}{1 - ae^{-j\omega}} + \frac{ae^{j\omega}}{1 - ae^{j\omega}} = \frac{1 - a^2}{1 - 2a\cos\omega + a^2}$$

15

Properties of DTFT

Time shifting:

- If $x_1[n] \xleftrightarrow{F} X_1(\omega)$
- Then $\boxed{x[n-k] \xleftrightarrow{DTFT} e^{-j\omega k} X(\omega)}$

Proof: Taking FT of x[n-k]

$$F\left[x[n-k]\right] = \sum_{n=-\infty}^{\infty} x[n-k] e^{-j\omega n}$$

Let n – k = m or n = m+k

$$\therefore F\left[x[n-k]\right] = \sum_{m=-\infty}^{\infty} x[m] e^{-j\omega(m+k)} = e^{-j\omega k} \sum_{m=-\infty}^{\infty} x[m] e^{-j\omega m} = e^{-j\omega k} X(\omega)$$

Similarly for x[n+k], $F\{x[n+k]\}=e^{jwk}X(w)$

Properties of DTFT

- Time reversal:
  - If $x[n] \xleftrightarrow{F} X(\omega)$
  - Then $\boxed{x[-n] \xleftrightarrow{DTFT} X(-\omega)}$

Proof: Let m = -n

$$F[x[-n]] = \sum_{n=-\infty}^{\infty} x[-n] e^{-j\omega n}$$

$$F[x[-n]] = \sum_{m=\infty}^{-\infty} x[m] e^{j\omega(-m)} = \sum_{m=-\infty}^{\infty} x[m] e^{-j(-\omega m)} = X(-\omega)$$

Properties of DTFT

- Convolution:
  - If

$$x_1[n] \xleftrightarrow{F} X_1(\omega)$$

$$x_2[n] \xleftrightarrow{F} X_2(\omega)$$

$$\boxed{x[n] = x_1[n] * x_2[n] \xleftrightarrow{DTFT} X(\omega) = X_1(\omega) X_2(\omega)}$$

This theorem is one of the most powerful tool. If we convolve 2 signals in time Domain, then this is equal to multiplying Their spectra in the freq: domain.

Proof: Recall convo: formula

$$x[n] = x_1[n] * x_2[n] = \sum_{k=-\infty}^{\infty} x_1[k] x_2[n-k]$$

Multiply both sides of this eq: by e-jwn and sum over all n, we get

$$\sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n} = \sum_{n=-\infty}^{\infty} \left[\sum_{k=-\infty}^{\infty} x[k] y[n-k] \right] e^{-j\omega n}$$

Interchanging the order of summation and making a substitution n – k = m,

$$X(\omega) = \sum_{k=-\infty}^{\infty} x_1[k] \left[\sum_{m=-\infty}^{\infty} x_2[m] \right] e^{-j\omega(m+k)} = \left[\sum_{k=-\infty}^{\infty} x_1[k] e^{-j\omega k} \right] \left[\sum_{m=-\infty}^{\infty} x_2[m] e^{-j\omega m} \right]$$

$$X(\omega) = X_1(\omega) X_2(\omega)$$

Example – 4: Determine the convolution of the sequences $x_1[n] = x_2[n] = [1\ \ \underline{1}\ \ 1]$

Solution – 4:

$$X_1(\omega) = X_2(\omega) = \sum_{n=-\infty}^{\infty} x_1[n]e^{-j\omega n} = \sum_{n=-1}^{1} x_1[n]e^{-j\omega n}$$

$$= \left[x_1[-1]e^{j\omega} + x_1[0]e^{0} + x_2[1]e^{-j\omega}\right] = \left[e^{j\omega} + 1 + e^{-j\omega}\right]$$

$$= 1 + 2\cos\omega$$

Therefore,

$$X(\omega) = X_1(\omega)X_2(\omega) = (1+2\cos\omega)^2 = 1 + 4\cos\omega + 4\cos^2\omega$$

$$= 1 + 4\cos\omega + \frac{4}{2}(1+\cos 2\omega) = 3 + 4\cos\omega + 2\cos 2\omega$$

$$= 3 + 2\left(e^{j\omega} + e^{-j\omega}\right) + \left(e^{j2\omega} + e^{-j2\omega}\right)$$

$$X(\omega) = X_1(\omega)X_2(\omega) = e^{j2\omega} + 2e^{j\omega} + 3 + 2e^{-j\omega} + e^{-j2\omega}$$

Hence the convolution of $x_1[n]$ and $x_2[n]$ is $x[n] = [1\ \ 2\ \ \underline{3}\ \ 2\ \ 1]$

Properties of DTFT

- Correlation:
  - If

$$x_1[n] \xleftrightarrow{F} X_1(\omega)$$

$$x_2[n] \xleftrightarrow{F} X_2(\omega)$$

  - Then

$$\boxed{r_{x_1x_2} \xleftrightarrow{DTFT} S_{x_1x_2}(\omega) = X_1(\omega)X_2(-\omega)}$$

- **The Wiener- Khintchine Theorem**:
  - Let x(n) be a real signal, then

$$\boxed{r_{xx}(l) \xleftrightarrow{DTFT} S_{xx}(\omega)}$$

  - That is, *the DTFT of autocorrelation function is equal to its energy density function*. This is a special case.
  - Autocorrelation sequence of a signal & its energy spectral density contain the same info: about the signal.

Proof: The autocorrelation of x[n] is defined as

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

Now

$$F[r_{xx}[n]] = \sum_{n=-\infty}^{\infty}\left[\sum_{k=-\infty}^{\infty} x[k]x[k-n]\right]e^{-j\omega n}$$

Re-arranging the order of summations and making Substitution m= k-n,

$$F[r_{xx}[n]] = \sum_{k=-\infty}^{\infty} x[k]\left[\sum_{m=\infty}^{-\infty} x[m]\right]e^{-j\omega(k-m)} = \left[\sum_{k=-\infty}^{\infty} x[k]e^{-j\omega k}\right]\left[\sum_{m=-\infty}^{\infty} x[m]e^{j(-\omega m)}\right]$$

$$= X(\omega)X(-\omega) = |X(\omega)|^2 = S_{xx}(\omega)$$

Properties of DTFT

- Frequency shifting:
  - If $x[n] \xleftrightarrow{F} X(\omega)$
  - Then $\boxed{e^{j\omega_0 n}x[n] \xleftrightarrow{DTFT} X(\omega-\omega_0)}$

According to this property, multiplication of a sequence x(n) by e^{jw0n}, is equivalent to a frequency translation of the spectrum X(w) by w_0

Proof:

$$F\left[x[n]e^{j\omega_0 n}\right] = \sum_{n=-\infty}^{\infty} x[n]e^{j\omega_0 n}e^{-j\omega n} = \sum_{n=-\infty}^{\infty} x[n]e^{-j(\omega-\omega_0)n} = X(\omega-\omega_0)$$

Properties of DTFT

- Modulation theorem:
  - If $x[n] \xleftrightarrow{F} X(\omega)$
  - Then $\boxed{x[n]\cos\omega_0 n \xleftrightarrow{DTFT} \frac{1}{2}[X(\omega+\omega_0)+X(\omega-\omega_0)]}$

- Parseval's Theorem:
  - If $x_1[n] \xleftrightarrow{F} X_1(\omega)$
    $x_2[n] \xleftrightarrow{F} X_2(\omega)$
  - Then

$$\boxed{\sum_{n=-\infty}^{\infty} x_1[n]x_2^*[n] \xleftrightarrow{DTFT} \frac{1}{2\pi}\int_{-\pi}^{\pi} X_1(\omega)X_2^*(\omega)d\omega}$$

Proof:

$$R.H.S. = \frac{1}{2\pi}\int_{-\pi}^{\pi} X_1(\omega)X_2^*(\omega)d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[\sum_{n=-\infty}^{\infty} x_1[n]e^{-j\omega n}\right]X_2^*(\omega)d\omega$$

$$= \sum_{n=-\infty}^{\infty} x_1[n]\frac{1}{2\pi}\int_{-\pi}^{\pi} X_2^*(\omega)e^{-j\omega n}d\omega = \sum_{n=-\infty}^{\infty} x_1[n]x_2^*[n] = L.H.S$$

In the special case where $x_1[n] = x_2[n] = x[n]$, the Parseval's Theorem reduces to:

$$\sum_{n=-\infty}^{\infty} |x(n)|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(\omega)|^2 d\omega$$

We observe that the LHS of the above equation is energy Ex of the Signal and the R.H.S is equal to the energy density spectrum. Thus we can re-write the above equation as:

$$E_x = \sum_{n=-\infty}^{\infty} |x[n]|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(\omega)|^2 d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} S_{xx}(\omega)d\omega$$

Properties of DTFT

- Multiplication of two sequences (Windowing theorem):
  - If $x_1[n] \xleftrightarrow{F} X_1(\omega)$

    $x_2[n] \xleftrightarrow{F} X_2(\omega)$
  - Then $\boxed{x_3 \equiv x_1[n]x_2[n] \xleftrightarrow{DTFT} X_3(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\lambda)Y(\omega-\lambda)d\lambda}$

This theorem states that: The multiplication of two time domain sequences is equivalent to the convolution of their Fourier transforms.

Proof:

$$F[x_1[n]x_2[n]] = \sum_{n=-\infty}^{\infty} x_1[n]x_2[n]e^{-j\omega n} = \sum_{n=-\infty}^{\infty}\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} X(\lambda)e^{j\lambda n}d\lambda\right]x_2[n]e^{-j\omega n}$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} x_1(\lambda)d\lambda\left[\sum_{n=-\infty}^{\infty} x_2[n]e^{-j(\omega-\lambda)n}\right] = \frac{1}{2\pi}\int_{-\pi}^{\pi} x_1(\lambda)x_2(\omega-\lambda)d\lambda$$

Properties of DTFT

- Differentiation in the Frequency domain:
  - If $x[n] \xleftrightarrow{F} X(\omega)$
  - Then $\boxed{nx[n] \xleftrightarrow{DTFT} j\dfrac{dX(\omega)}{d\omega}}$

Proof:

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

$$\frac{dX(\omega)}{d\omega} = \frac{d}{d\omega}\left[\sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}\right] = \sum_{n=-\infty}^{\infty} x[n]\frac{d}{d\omega}e^{-j\omega n}$$

$$\frac{dX(\omega)}{d\omega} = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}(-jn)$$

$$j\frac{dX(\omega)}{d\omega} = \sum_{n=-\infty}^{\infty} nx[n]e^{-j\omega n}$$

Properties of DTFT

- Linearity: $a_1x_1[n]+a_2x_2[n]\xleftrightarrow{DTFT}a_1X_1(\omega)+a_2X_2(\omega)$
- Time shifting: $x[n-k]\xleftrightarrow{DTFT}e^{-j\omega k}X(\omega)$
- Conjugate: $x^*[n]\xleftrightarrow{DTFT}X^*(e^{-j\omega})=X^*(\omega)$
- Time reversal: $x[-n]\xleftrightarrow{DTFT}X(-\omega)$
- Frequency shifting: $e^{j\omega_0 n}x[n]\xleftrightarrow{DTFT}X(\omega-\omega_0)$
- Differentiation: $nx[n]\xleftrightarrow{DTFT}j\dfrac{dX(\omega)}{d\omega}$
- Convolution: $x[n]=x_1[n]*x_2[n]\xleftrightarrow{DTFT}X(\omega)=X_1(\omega)X_2(\omega)$
- Correlation: $r_{x_1x_2}\xleftrightarrow{DTFT}S_{x_1x_2}(\omega)=X_1(\omega)X_2(-\omega)$
- Wiener khinchine: $r_{xx}(l)\xleftrightarrow{DTFT}S_{xx}(\omega)$
- Multiplication: $x_3\equiv x_1[n]x_2[n]\xleftrightarrow{DTFT}X_3(\omega)=\dfrac{1}{2\pi}\int_{-\pi}^{\pi}X(\lambda)Y(\omega-\lambda)d\lambda$
- Parseval's Theorem: $\sum_{n=-\infty}^{\infty}x_1[n]x_2^*[n]\xleftrightarrow{DTFT}\dfrac{1}{2\pi}\int_{-\pi}^{\pi}X_1(\omega)X_2^*(\omega)d\omega$
- Modulation Theorem:

$$x[n]\cos\omega_0 n\xleftrightarrow{DTFT}\frac{1}{2}[X(\omega+\omega_0)+X(\omega-\omega_0)]$$

Tutorial:

1. Find the DTFT of $y[n]=-\alpha^n u(-n-1)$, provided $|\alpha|>1$
2. Prove correlation property of DTFT.
3. Prove modulation theorem of DTFT.
4. Prove differentiation property of DTFT.
5. An LTI system is characterized by its impulse response $h[n] = (1/2)^n u[n]$. Determine the spectrum and the energy density spectrum of the output signal when the system is excited by the signal $x[n] = (1/4)^n u[n]$.

Discrete Fourier Transform

- Recall the definition of DTFT:

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \longrightarrow \quad (1)$$

- While the DTFT is useful from a theoretical point of view, its numerical evaluation poses difficulties:
  - The summation over n is infinite
  - The variable ω is continuous
- In many situations of interest, it is either not possible, or not necessary to implement the infinite summation in (1).
  - Only the signal samples of $x[n]$ from n=0 to N-1 are available;
  - The signal is known to zero outside this range; or
  - The signal is periodic with period N.
- In all these cases, we would like to analyze the frequency content of signal $x[n]$ based only on the finite set of samples $x[0], x[1], \ldots, x[N-1]$.
- We would also like a frequency domain representation of these samples in which the frequency variable only take a finite set of values, say ω_k for $k=0, 1, \ldots, N-1$.
- The Discrete Fourier Transform (DFT) fulfils these needs. It can be seen as an approximation to the DTFT.

Definition: Discrete Fourier Transform

The N-point DFT is a transformation that maps DT signal samples $\{x[0], \ldots, x[N-1]\}$ into a periodic sequence $x[k]$, defined by

$$\boxed{x[k] = DFT_N\{x[n]\} = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \qquad k \in Z}$$

Remarks:

- Only the samples $x[0], \ldots, x[N-1]$, are used in computation.
- The N-point DFT is periodic, with period N: $x[k+N]=x[k]$. Thus it is sufficient to specify $x[k]$ for $k=0,1, \ldots, N-1$.

Inverse DFT (IDFT)

Definition: Inverse DFT

The N-point IDFT of the samples $x[0], \ldots, x[N\text{-}1]$ is defined as the periodic sequence $x[k]$, defined by:

$$\tilde{x}[n] = IDFT_N\{X[k]\} = \frac{1}{N}\sum_{n=0}^{N-1} x[k]e^{j2\pi kn/N}, \qquad k \in Z$$

Remarks:

- In general, $\tilde{x}[n] \neq x[n]$ for all n є Z
- Only the samples, $x[0], \ldots, x[N\text{-}1]$, are used in the computation.
- The N-point DFT is periodic, with period N: $\tilde{x}[n+N] = x[k]$

IDFT Theorem

IDFT Theorem:

If $X[k]$ is the N-point DFT of $\{x[0], \ldots x[N-1]\}$, then

$$\tilde{x}[n] = x[n], \qquad n = 0,1,\ldots,N-1 \text{ only}.$$

Remarks:

- Theorem states that $\tilde{x}[n] = x[n]$ for $n = 0,1, \ldots, N$-1 only.
- In general, the values of $x[n]$ for $n < 0$ and for $n \geq N$ cannot be recovered from the DFT samples $X[k]$. This is understandable since these sample values are not used when computing $X[k]$.
- However, there are two important cases when the complete signal $x[n]$ can be recovered from the DFT samples $X[k]$ (k=0,1,…, N-1)
  - $x[n]$ is periodic with period N.
  - $x[n]$ known to be zero for $n < 0$ and for $n \geq N$.

$$\boxed{\begin{aligned} x[k] &= \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} \\ \tilde{x}[n] &= \frac{1}{N}\sum_{n=0}^{N-1} x[k] e^{j2\pi kn/N} \end{aligned}}$$

Example

Example: Prove that DFT is periodic with period N.

Proof: we know that, DFT is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-jk2\pi n/N}$$

Therefore,

$$X[k+N] = \sum_{n=0}^{N-1} x[n] e^{-j(k+N)2\pi n/N} = \sum_{n=0}^{N-1} x[n] e^{-jk2\pi n/N} e^{-j2\pi n}$$

Sine $e^{-j2\pi n} = 1$,

$$\therefore X[k+N] = \sum_{n=0}^{N-1} x[n] e^{-jk2\pi n/N} = x[k]$$ hence, proved.

Example

Example: Find the DFT of x[n] = [1 0 0 1]

Solution: $$X[k]=\sum_{n=0}^{N-1}x[n]e^{-jk2\pi n/N}=\sum_{n=0}^{3}x[n]e^{-jk2\pi n/4}=\sum_{n=0}^{3}x[n]e^{-jk\pi n/2}$$

Now, $$X[0]=\sum_{n=0}^{3}x[n]=x[0]+x[1]+x[2]+x[3]=1+0+0+1=2$$

$$X[1]=\sum_{n=0}^{3}x[n]e^{-jk\pi n/2}=x[0]+0+0+x[3]e^{-j3\pi/2}$$

$$=1+1.e^{-j3\pi/2}=1+\cos(\tfrac{3\pi}{2})-j\sin(\tfrac{3\pi}{2})=1+j$$

$$X[2]=\sum_{n=0}^{3}x[n]e^{-j\pi n}=x[0]+x[3]e^{-j3\pi n}$$

$$=1+1.[\cos(3\pi n)-j\sin(3\pi n)]=0$$

$$X[3]=\sum_{n=0}^{3}x[n]e^{-j3\pi n/2}=x[0]+x[3]e^{-j9\pi/2}=1-j$$

Example

Example: Find the IDFT of the sequence y[n]=[2 1+i 0 1-i]

Solution: $x[n]=\frac{1}{N}\sum_{k=0}^{N-1}X[k]e^{jk2\pi n/N}$

$$x[0]=\frac{1}{4}\sum_{k=0}^{N-1}X[k]=\frac{1}{4}[2+(1+i)+0+(1-i)]=1$$

$$x[1]=\frac{1}{4}\sum_{k=0}^{3}X[k]e^{jk2\pi/4}=\frac{1}{4}\sum_{k=0}^{3}X[k]e^{jk\pi/2}=\frac{1}{4}\left[2+(1+i)e^{j\pi/2}+0.e^{j\pi}+(1-i)e^{j3\pi/2}\right]=0$$

$$x[2]=\frac{1}{4}\sum_{k=0}^{3}X[k]e^{jk4\pi/4}=\frac{1}{4}\sum_{k=0}^{3}X[k]e^{jk\pi}=\frac{1}{4}\left[2+(1+i)e^{j\pi}+0.e^{j2\pi}+(1-i)e^{j3\pi}\right]=0$$

$$x[3]=\frac{1}{4}\sum_{k=0}^{3}X[k]e^{jk6\pi/4}=\frac{1}{4}\sum_{k=0}^{3}X[k]e^{jk3\pi/2}=\frac{1}{4}\left[2+(1+i)e^{j3\pi/2}+0.e^{j3\pi}+(1-i)e^{j9\pi/2}\right]=1$$

Properties of DFT

Periodicity:

The N-point DFT is periodic, with period N: $\tilde{x}[n+N]=x[k]$

Linearity:

If $x[n]$ and $y[n]$ have N-point DFTs $X(k)$ and $Y(k)$, respectively,

$$ax[n]+by[n] \xleftrightarrow{DFT} aX(k)+bY(k)$$

In using this property, it is important to ensure that the DFTs are the same length. If $x[n]$ and $y[n]$ have different lengths, then shorter sequence must be *padded* with zeros in order to make it the same length as the longer sequence.

Symmetry:

If $x[n]$ is real-valued, $X(k)$ is *conjugate symmetric,*

$$X(k)=X^*((-k))=X^*((N-k))_N$$

and if $x[n]$ is imaginary, $X(k)$ is *conjugate antisymmetric,*

$$X(k)=-X^*((-k))=-X^*((N-k))_N$$

Example

Example: A finite duration sequence of length L is given by

$$x[n] = \begin{cases} 1, & 0 \le n \le L-1 \\ 0, & \text{otherwise} \end{cases}$$

Determine the N point DFT of this sequence for N≥L.

Solution: The DTFT of the sequence was calculated as

$$X(w) = \frac{\sin(wL/2)}{\sin(w/2)} e^{-jw(L-1)/2}$$

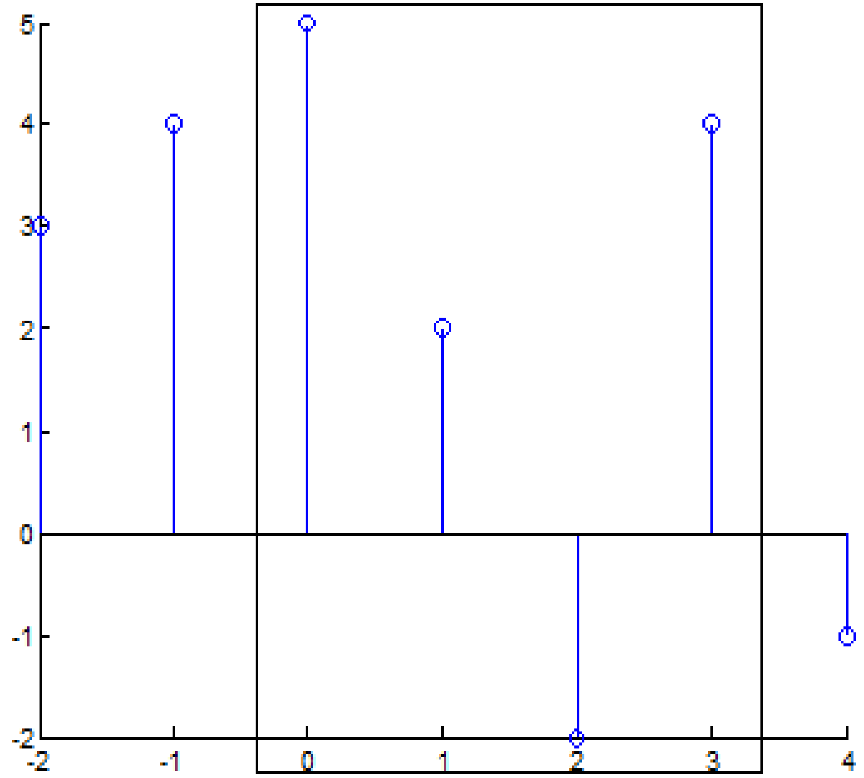
The N point DFT is simply X(w) evaluated at the set of N equally spaced frequencies $w_k = 2\pi k/N$, $k = 0,1,\ldots,N-1$. Hence

$$X(k) = \frac{\sin(\pi kL/N)}{\sin(\pi k/N)} e^{-jwk(L-1)/N}$$

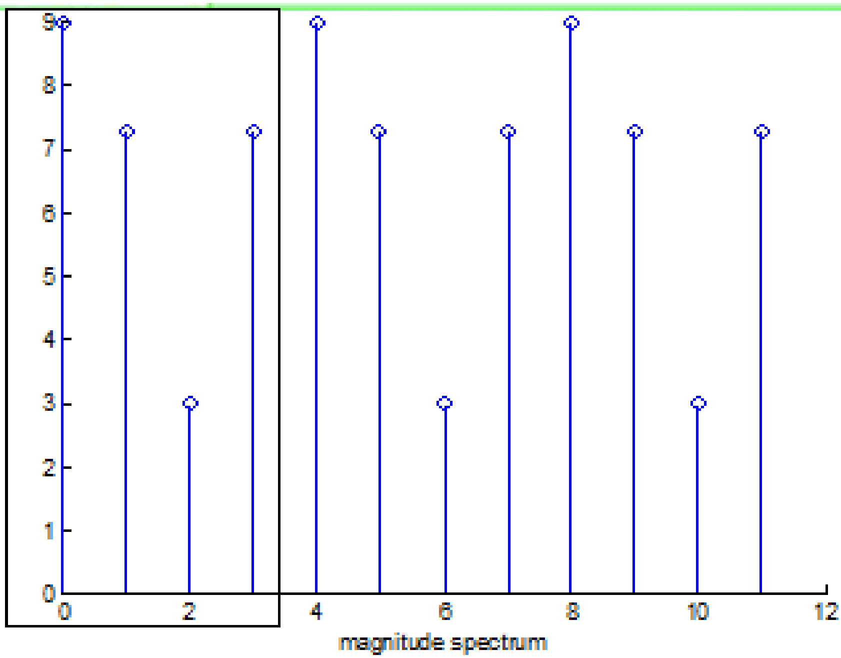
Example

- Example: Find DFT magnitude and phase spectra for the samples of the signal selected in figure. Also verify that IDFT reproduces these samples.
- Solution:

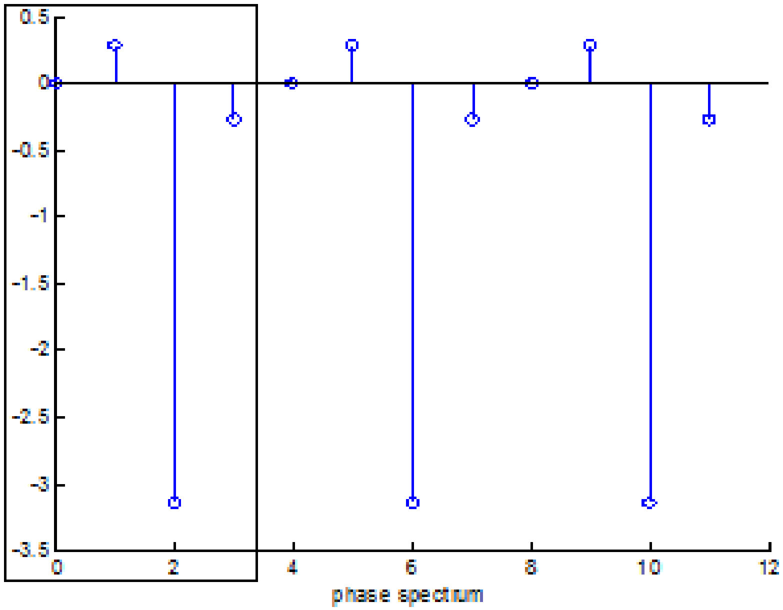
K	X[k]	|X[k]|	<θ radians
0	9	9	0
1	7+2j	7.2801	0.2782
2	-3	3	-3.1416
9	7-2j	7.2801	-0.2782



Example



0
1
2
3
4
5
6
7
8
9
0
2
4
6
8
10
12
magnitude spectrum

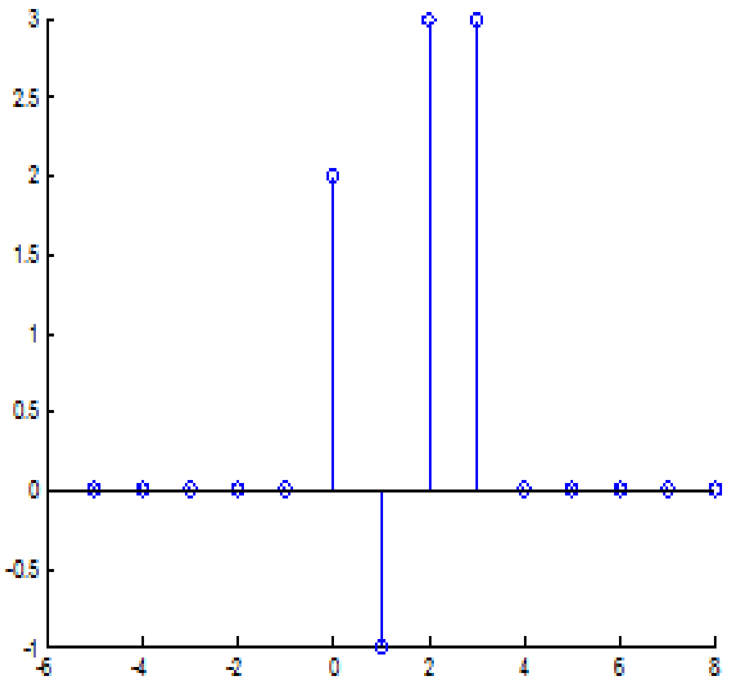


0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
0
2
4
6
8
10
12
phase spectrum

Example

Tutorial:

1. Find the DFT of x[n]=[2 1 0 2]
2. Find the IDFT of X[k]=[1+i 0 1 1-i]
3. Find the DFT of the 4-point sequence x[n] = [0 1 2 3]
4. Find the 4 point IDFT of the sequence [6,-2+2j,-2,-2-2j].
5. Find magnitude spectrum using both DTFT and DFT for the signal shown here.



Fast Fourier Transform (FFT)

Recap:

- Let $x[n]$ be a discrete-time signal defined for $0 \le n \le N\text{-}1$.
- The DFT:

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \qquad k = 0,1,\ldots, N-1 \qquad \textbf{(1)}$$

$$W_N = e^{-j2\pi/N} = \cos(2\pi/N) + j\sin(2\pi/N)$$

- Notes:
  - Note that the direct computation of DFT requires N^2 computations.
  - The same is true for IDFT
  - The FFT only requires $N\log_2 N$ calculations.
  - The computational saving achieved by FFT is therefore a factor of $N/\log_2 N$. When N is large this saving can be significant.
  - The following table compares the number of calculations required for different values of N for the DFT and FFT:

N	DFT	FFT
32	1024	160
1024	1048576	10240
32768	~ 1 x 10^9	~ 0.5 X 10^6

Fast Fourier Transform (FFT)

What is FFT

- FFT stands for Fast Fourier Transform
- FFT is a method of computing the Discrete Fourier Transform (DFT) that exploits the redundancy in the general DFT equation given in (1).
- The FFT is not a new transform; it refers to a family of efficient algorithms for computing the DFT.
- Typically, FFT requires $N\log_2 N$ while DFT requires N^2.

Basic Principle

- The FFT relies on the concept of divide and conquer
- It is obtained by breaking the DFT of size N into a cascade of smaller size DFTs.
- To achieve this:
  - N must be a composite number
  - The properties of W_N must be exploited, e.g.;

$$W_N^k = W_N^{k+N} \qquad (2)$$

$$W_N^{Lk} = W_{N/L}^k \qquad (3)$$

Example

Example: We can highlight the existence of redundant computations in the DFT by inspecting Eq. (1). Using the DFT algorithm to calculate the first four components of the DFT of a signal with only 8 samples requires the following computations:

$$X[0] = x[0] + x[1] + x[2] + x[3] + x[4] + x[5] + x[6] + x[7]$$
$$X[1] = x[0] + x[1]W_8^{-1} + \boxed{x[2]W_8^{-2}} + x[3]W_8^{-3} + x[4]W_8^{-4} + x[5]W_8^{-5} + x[6]W_8^{-6} + x[7]W_8^{-7}$$
$$X[2] = x[0] + x[1]W_8^{-2} + x[2]W_8^{-4} + x[3]W_8^{-6} + x[4]W_8^{-8} + x[5]W_8^{-10} + x[6]W_8^{-12} + x[7]W_8^{-14} \quad (4)$$
$$X[3] = x[0] + x[1]W_8^{-3} + \boxed{x[2]W_8^{-6}} + x[3]W_8^{-9} + x[4]W_8^{-12} + x[5]W_8^{-15} + x[6]W_8^{-18} + x[7]W_8^{-21}$$

However note that there is redundant (repeated) terms in Eq. (4). For e.g., consider 3rd term in 2nd line of Eq. (4).

$$x[2]W_8^{-2} = x[2]e^{j2\pi\left(\frac{-2}{8}\right)} = x[2]e^{\frac{-j\pi}{2}}$$

Now, consider the computation of third term in the fourth line of Eq. (4):

$$x[2]W_8^{-6} = x[2]e^{j2\pi\left(\frac{-6}{8}\right)} = x[2]e^{\frac{-j3\pi}{2}} = x[2]e^{-j\pi}e^{\frac{-j\pi}{2}} = -x[2]e^{\frac{-j\pi}{2}}$$

Therefore we can save one multiply operation by noting that $x[2]W_8^{-6} = -x[2]W_8^{-2}$

In fact because of the periodicity of $x[k]W_N^{nk}$ every term in the fourth line of Eq. (4) is available from the computed terms in the second line of the equation.

Example

More generally, we can show that the terms in the second line of Eq. (4) are:

$$x[k]W_8^{-k} = x[k]e^{\frac{-j2\pi k}{2}} = x[k]e^{\frac{-j\pi k}{2}}$$

and for the terms in fourth line of Eq. (4):

$$x[k]W_8^{-3k} = x[k]^{-j\frac{6\pi k}{2}} = x[k]e^{-j\frac{3\pi k}{2}} = x[k]e^{-j\left(\frac{\pi}{2}+\frac{\pi}{4}\right)k}$$

$$= x[k]e^{-j\frac{\pi k}{2}}e^{-j\frac{\pi k}{4}} = x[k](-j)^k e^{-j\frac{\pi k}{4}} = (-j)^k x[k]W_8^{-k}$$

This exploitation of the computational redundancy is the basis of FFT which allows the same results as the DFT to be computed, but with less computations.

Different Types of FFT

- There are several FFT algorithms sometimes grouped via the names Cooley- Tukey, prime factor, decimation in time, decimation in frequency, radix-2 and so on. The bottom line for all FFT algorithms is, however, that they remove redundancy from the direct DFT computational algorithm of Eq. (1).

Notable Examples of FFT Algorithms:

- $N = 2^{v} \rightarrow$ Radix – 2 FFTs. These are the most commonly used algorithms. Even then, there are many variations:
  - Decimation in Time (DIT)
  - Decimation in Frequency (DIF)
- $N = r^{v} \rightarrow$ Radix – r FFTs. The special case r = 3 and r = 4 are not uncommon.
- More generally, $N = p_1 p_2 p_3 \ldots p_l$ where the $p_i s$ are prime numbers lead to so called mixed-radix FFTs.

Radix-2 are the most important. Only in very specialized situations will it be more advantageous to use other radix-type FFTs.

We'll focus on this type only in this course

Radix-2 FFT

- We only consider radix – 2 FFTs (i.e., $N = 2^{\nu}$), where
  - DFT_N is decomposed into a cascade of ν stages
  - Each stage is made up of $N/2$ DFT_2

Radix – 2 FFT via Decimation in Time:

- Let $x[n]$ be a discrete-time signal defined for $0 \le n \le N\text{-}1$, where $N = 2^{\nu}$.
- The basic idea behind decimation in time (DIT) is to partition the input sequence $x[n]$, of length N, into two sub-sequences, i.e. $x[2r]$ and $x[2r+1]$, $r = 0, 1, \ldots, (N/2) - 1$, corresponding to even and odd values of time, respectively.
- The N-point DFT of $x[n]$ can be computed by properly combining the $(N/2)$-point DFTs of each subsequences.
- In turn, the same principle can be applied in the computation of the $(N/2)$-point DFT of each subsequence, which can be reduced to DFTs of size $N/4$.
- This basic principle is repeated until only 2-point DFTs are involved.
- The final result is an FFT algorithm of complexity $N/2\log_2 N$ complex multiplication and $N\log_2 N$ complex additions..

Radix-2 FFT

- Radix-2 rearranges the DFT equation into 2 parts having indices as

$$X(k)=\sum_{n=0}^{N-1}x(n)e^{-\frac{j2\pi kn}{N}}$$

$$n=\{0,2,4,\ldots N-2\}$$

$$n=\{1,3,5,\ldots N-1\}$$

$$X(k)=\sum_{n=0}^{\frac{N}{2}-1}x(2n)e^{-\frac{j2\pi k(2n)}{N}}+\sum_{n=0}^{\frac{N}{2}-1}x(2n+1)e^{-\frac{j2\pi k(2n+1)}{N}}$$

$$X(k)=\sum_{n=0}^{\frac{N}{2}-1}x(2n)e^{-\frac{j2\pi kn}{\frac{N}{2}}}+e^{-\frac{j2\pi k}{N}}\sum_{n=0}^{\frac{N}{2}-1}x(2n+1)e^{-\frac{j2\pi kn}{\frac{N}{2}}}$$

$$X(k)=G(k)+W_N^kH(k)$$

- This is called Decimation in time because the time samples are rearranged in alternating groups

Radix-2 FFT

- Radix-2 rearranges the DFT equation into 2 parts having indices as

$$X(k)=\sum_{n=0}^{N-1}x(n)e^{-\frac{j2\pi kn}{N}} \qquad n=\{0,2,4,\ldots N-2\} \qquad n=\{1,3,5,\ldots N-1\}$$

$$X(k)=\sum_{n=0}^{\frac{N}{2}-1}x(2n)e^{-\frac{j2\pi k(2n)}{N}}+\sum_{n=0}^{\frac{N}{2}-1}x(2n+1)e^{-\frac{j2\pi k(2n+1)}{N}}$$

$$X(k)=\sum_{n=0}^{\frac{N}{2}-1}x(2n)e^{-\frac{j2\pi kn}{\frac{N}{2}}}+e^{-\frac{j2\pi k}{N}}\sum^{\frac{N}{2}-1}\cdots e^{-\frac{j2\pi kn}{N}\cdots}$$

$$X(k)=G(k)+W_N^k H(k)$$

The mathematical simplification reveal that all DFT freq: o/ps X(k) can be computed as the sum of the o/ps of two length N/2 DFTs, of even & odd indexed discrete time samples respectively, where the odd-indexed short DFT is multiplied by a so called Twiddle factor term.

- This is called Decimation in …
samples are rearranged in a …

2-Points FFT

The 2-point FFT:

- In the case $N = 2$, (1) specializes to,

$$X[k] = G[k] + H[k]W_2^k, \qquad k = 0,1$$

- Since, $W_2 = e^{-j\pi} = -1$, this can be further simplified to

$$X[0] = G[0] + H[1]$$

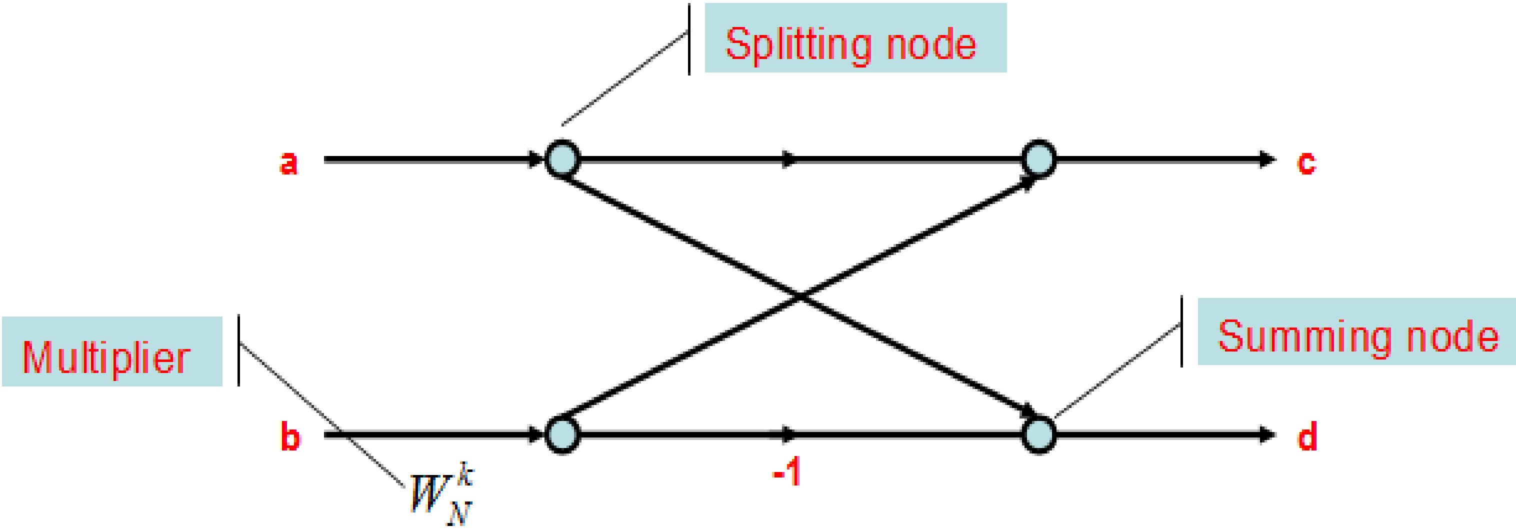
$$X[1] = G[0] - H[1]$$

Main steps of DIT:

- Split the summation $\sum_n$ in (1) into even $\sum_{n\ \text{even}}$ and odd $\sum_{n\ \text{odd}}$ parts as (N/2)-point DFTs.
- If N /2 = 2 stop; else, repeat the above steps for each of the individual (N/2)-point DFT.

"Butterfly" Signal Flow Graph

- In general, the equations for FFT are awkward to write mathematically, and therefore the algorithm is very often represented as a "butterfly" based signal flow graph (SFG), the butterfly being a simple SFG of the form:



- The multiplier is a complex number and the input data, a and b, may also be complex. One butterfly computation requires one complex multiply and two complex additions (assuming data is complex).

The 4-point FFT

Case $N = 4 = 2^2$:

- Step – 1: $$X[k] = X[0] + X[1]W_4^k + X[2]W_4^{2k} + X[3]W_4^{3k},$$ $$= \left(X[0] + X[2]W_4^{2k}\right) + W_4^k\left(X[1] + X[3]W_4^{2k}\right)$$
- Step – 2: Using the property $W_4^{2k} = W_4^k$, we can write $$X[k] = \left(X[0] + X[2]W_4^k\right) + W_4^k\left(X[1] + X[3]W_4^k\right)$$ $$= G[k] + W_4^k H[k]$$ $$G[k] = DFT_2\{\text{even samples}\}$$ $$H[k] = DFT_2\{\text{odd samples}\}$$

Note that $G[k]$ and $H[k]$, are 2-periodic, i.e.

$$G[k+2] = G[k], \qquad H[k+2] = H[k]$$

- Step – 3: Since $N/2 = 2$, we simply stop; that is, the 2-point DFTs $G[k]$ and $H[k]$ cannot be further simplified via DIT.

The 4-point FFT

Interpretation:

- The 4-point DFT can be computed by properly combining the 2-point DFTs of the even and odd samples, i.e. $G[k]$ and $H[k]$, respectively:

$$X[k] = G[k] + W_4^k H[k], \qquad k = 0, 1, 2, 3$$

- Since $G[k]$ and $H[k]$ are 2-periodic, they only need to be computed for $k = 0$, 1:

$$X_0[k] = G[0] + W_4^0 H[0]$$

$$X_1[k] = G[1] + W_4^1 H[1]$$

$$X_2[k] = G[2] + W_4^2 H[2] = G[0] + W_4^2 H[0]$$

$$X_3[k] = G[3] + W_4^3 H[0] = G[1] + W_4^3 H[1]$$

Radix-4 FFT

- The radix-4 decimation in time algorithm rearranges of every fourth discrete time index

$$n = \{0, 4, 8, \ldots N-4\}$$
$$n = \{1, 5, 9, \ldots N-3\}$$
$$n = \{2, 6, 10, \ldots N-4\}$$
$$n = \{3, 7, 11, \ldots N-4\}$$

- This works out only when the FFT length is multiple of four.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi kn}{N}}$$

$$X(k) = \sum_{n=0}^{\frac{N}{4}-1} x(4n) e^{-\frac{j2\pi k(4n)}{N}} + \sum_{n=0}^{\frac{N}{4}-1} x(4n+1) e^{-\frac{j2\pi k(4n+1)}{N}}$$
$$+ \sum_{n=0}^{\frac{N}{4}-1} x(4n+2) e^{-\frac{j2\pi k(4n+2)}{N}} + \sum_{n=0}^{\frac{N}{4}-1} x(4n+3) e^{-\frac{j2\pi k(4n+3)}{N}}$$

Radix-4 FFT

$$X(k) = DFT_{\frac{N}{4}}[x(4n)] + W_N^k DFT_{\frac{N}{4}}[x(4n+1)]$$

$$+ W_N^{2k} DFT_{\frac{N}{4}}[x(4n+2)] + W_N^{3k} DFT_{\frac{N}{4}}[x(4n+3)]$$

- This is called Decimation in time because time samples are rearranged in alternating groups and a radix-4 algorithm because there are four groups.

Split Radix FFT

- By mixing radix-2 & radix-4 computations appropriately, an algorithm of lower complexity than other can be derived.

$$X(k)=\sum_{n=0}^{N-1}x(n)e^{-\frac{j2\pi kn}{N}}$$

$$X(k)=\sum_{n=0}^{\frac{N}{2}-1}x(2n)e^{-\frac{j2\pi k(2n)}{N}}+\sum_{n=0}^{\frac{N}{4}-1}x(4n+1)e^{-\frac{j2\pi k(4n+1)}{N}}+\sum_{n=0}^{\frac{N}{4}-1}x(4n+3)e^{-\frac{j2\pi k(4n+3)}{N}}$$

$$X(k)=DFT_{\frac{N}{2}}[x(2n)]+W_N^kDFT_{\frac{N}{4}}[x(4n+1)]+W_N^{3k}DFT_{\frac{N}{4}}[x(4n+3)]$$

End of Chapter