Ninevah University
College of Electronics Engineering
Control and Systems Engineering
Third Stage / 1st semester 2025-2026
Digital Signal Processing (DSP)

Lecture1: Discrete-Time Signals in The Time Domain

Instructor:
Asst. Prof. Dr. Ahmed Jameel Abdulqader

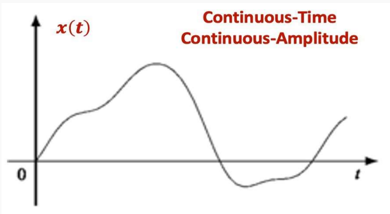
Place and Date: Mosul / College of Electronics Engineering, /9/2025

Reference Books

- Digital Signal Processing Principles, Algorithms and Applications. By John G.Proakis & Dimitris G.Manolakis.
- Digital Signal Processing. By Sen M. Kuo & Woon-Seng Gan.
- Digital Signal Processing A Practical Approach. By Emmanuel C. Ifeachor & Barrie W. Jervis.
- Digital Signal Processing Using MATLAB for Students and Researches. By John W. Leis.
- Digital Signal Processing Using MATLAB Third Edition. By Vinay K. Ingle John G. Proakis.
- Introduction to Digital Signal Processing and Filter Design. By B. A. Shenoi.
- Understanding Digital Signal Processing with MATLAB and Solutions. By Alexander D. Poularikas

Continuous-Time (CT) signal

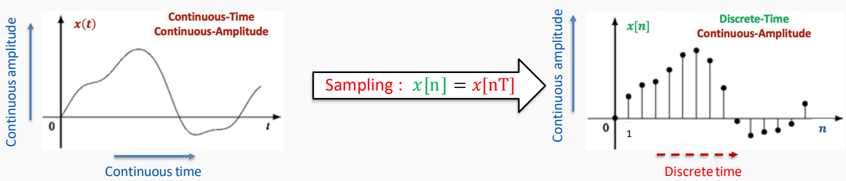
- Continuous-Time (CT) signal $x(t)$ is a signal that exists at every instant of time
  - A CT signal is often referred to as analog signal
  - The independent variable is a continuous variable
  - Continuous signal can assume any value over a continuous range of numbers



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

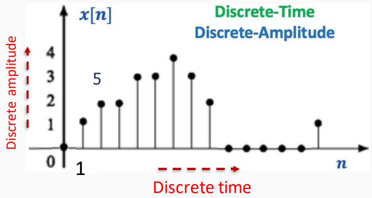
Discrete-Time (DT) Signals

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



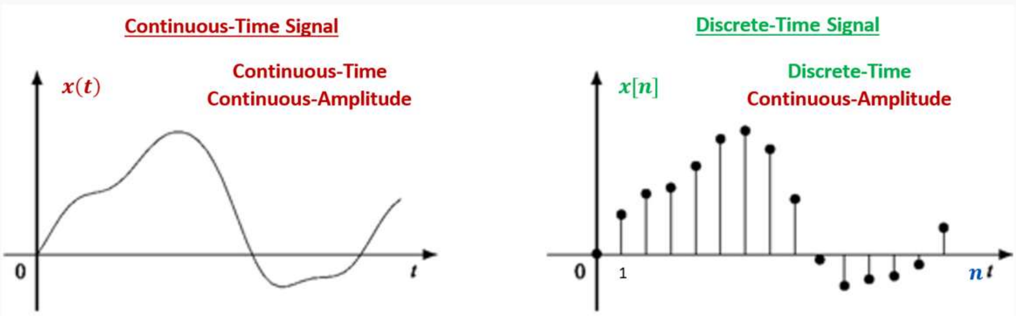
Digital Signals

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

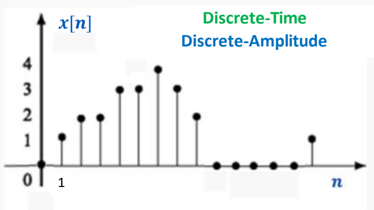


- The digital signal is the sampled and quantized (rounded) representation of the analog signal. A digital signal consists of a sequence of samples, which in this case are integers: 0, 1, 2, 2, 3, 3, 4, 3, …

Digital Signals

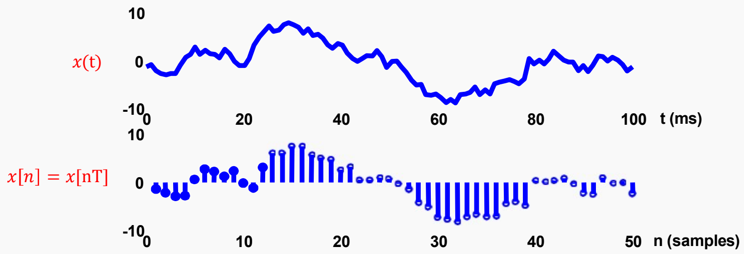


Digital Signal



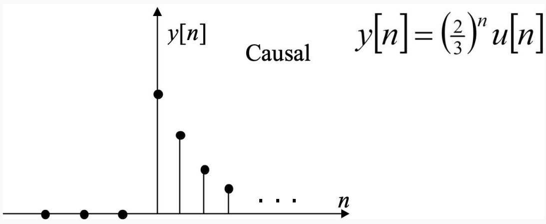
Digital Signals

- Discrete-Time (DT) 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 $x(t)$ at $x[nT]$ where T is the sampling period



Causal Sequences

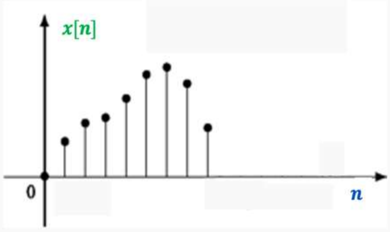
- Discrete-time signal or sequence is called **causal** if it has zero values for $n < 0$.
- Here is an example of a causal sequence



Causality and Duration of Sequences

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

This is an example of causal finite-duration sequence
- $x[n] = 0$ for $n < 0$ (causal)
- $x[n] \neq 0$ only for $0 \leq n \leq 8$



Basic Discrete-Time Signals (Sequences)

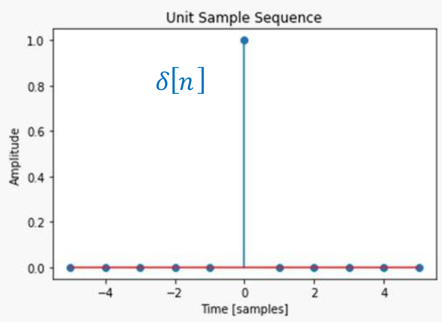
The most basic discrete-time signals (sequences) are
- Unit Impulse Sequence
- Unit Step Sequence
- Unit Ramp Sequence
- Power Sequence
- Exponential Sequence
- Sinusoidal Sequence

10

Unit Impulse Sequence

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

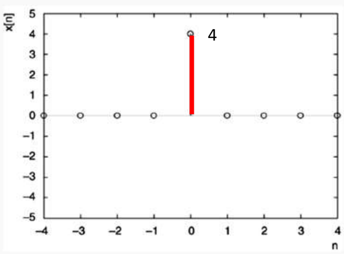
$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$



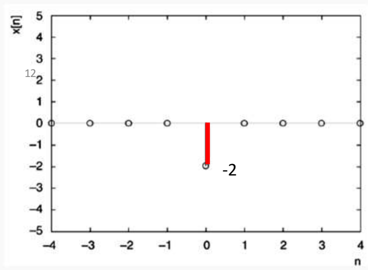
- The unit impulse sequence $\delta[n]$ has an amplitude of zero at all samples except $n = 0$, where it has the value 1.
- Every discrete-time signal can be written as a sum of unit impulse sequences, using the amplitude at each sample.

Scaled and Shifted Unit Impulse Function

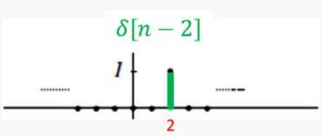
1. Draw the sequence of $x[n] = 4\delta[n]$



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

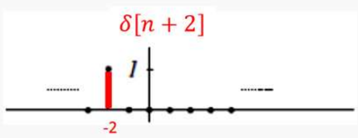


3. Draw the sequence of $x[n] = \delta[n-2]$



Shift to right

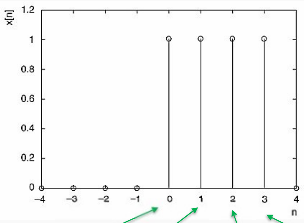
4. Draw the sequence of $x[n] = \delta[n+2]$



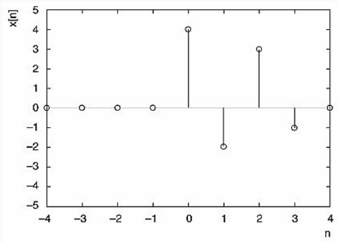
Shift to left

Finite-Duration Sequence in Terms of Unit Impulses

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



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

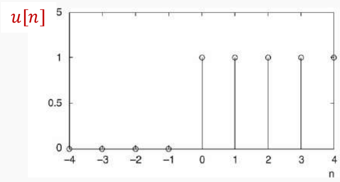


$x[n] = 4\delta[n] - 2\delta[n-1] + 3\delta[n-2] - \delta[n-3]$

Unit Step Sequence

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

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

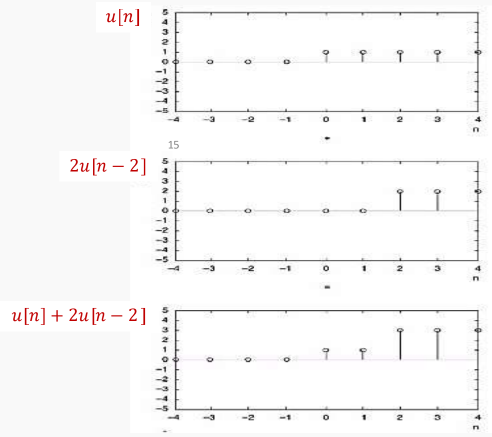


- The unit step function $u[n]$ has an amplitude of zero for $n < 0$ and an amplitude of one for all other samples.
- The signal $u[-n]$ has the value one up to and including $n = 0$, and the value zero thereafter.

Unit Step Sequence Examples

1-Draw the sequence as sum of two step sequences:

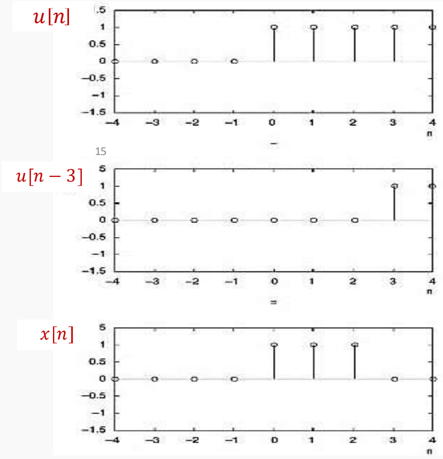
$x[n] = u[n] + 2u[n-2]$



2-Draw the sequence as sum of two step sequences:

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

This is a finite duration sequence



Connection between Impulse and Step Sequences

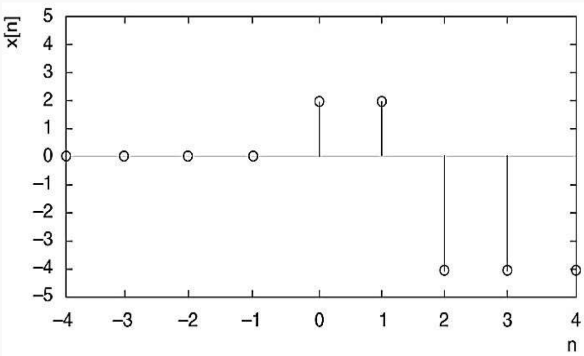
- **Write a sequence to describe the signal in terms of unit sample and unit step sequences of the figure.**

Unit Impulse Sequence

x[n] = 2δ[n] + δ[n-1] - 4δ[n-2] - 4δ[n-3] -4δ[n-4] - ....

Unit Step Sequence

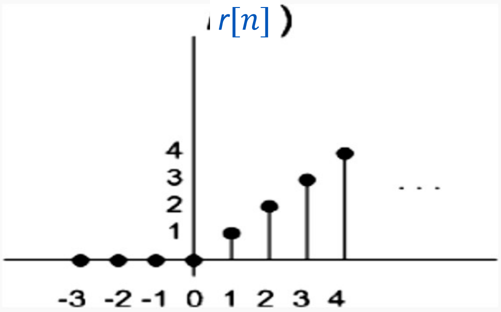
x[n] = 2u[n] – 6u[n-2]



Unit Ramp Sequence

- The unit-ramp sequence is defined as

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



Power Sequences

- Power sequences take the form:

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

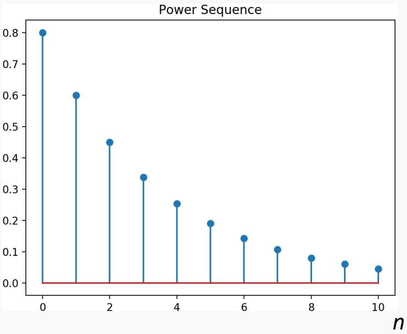
- In the special case where $\alpha = e$ (Euler number), such sequences are called exponential sequences.
- When β is positive, the values of the sequence decays.
- When β is negative the values of the sequence grows.
- When α is negative, the values of the sequence alternate positive and negative.
-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

Power Sequence Examples

- **Draw a signal**

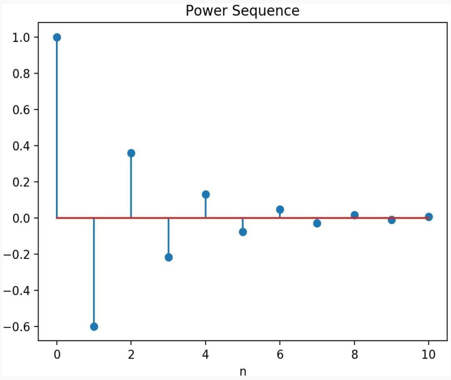
  $x[n] = 0.8(0.75)^n$

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



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

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



Exponential Sequences

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

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

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

- **Draw an Exponential Sequence of**

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

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

Complex Exponential Sequences

- A sequence of the form

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

is called a complex exponential sequence.

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

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

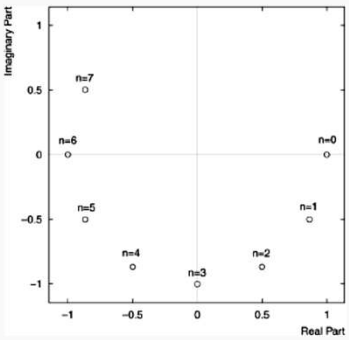
- The general form of a **complex exponential sequence** has the forms

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

Plot the first eight samples of a complex exponential sequence:

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

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



Sinusoidal Sequence

- The sinusoidal functions take the form

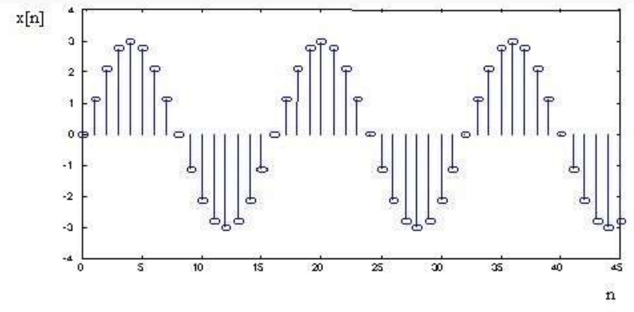
$$x[n] = A\cos(\omega n + \phi) \quad \text{or} \quad x[n] = A\sin(\omega n + \phi)$$

for all n with real A

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

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

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

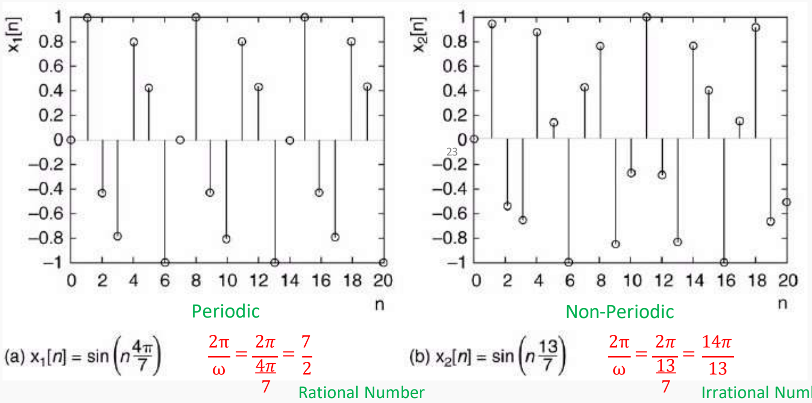


Periodicity of Sinusoidal Sequence

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

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

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



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

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

Periodicity of Sinusoidal Sequence Examples HW

A discrete-time signal is defined as $x[n] = \cos(2n)$
- Is this a periodic sequence?
- Find the first eight elements in the sequence.

A discrete-time signal is defined as $x[n] = \cos\left(\frac{4\pi}{5}n\right)$
- Is this a periodic sequence?
- Find the first eight elements in the sequence.

Draw a signal x[n] = u[n]u[3-n]
- The signal can also be expressed as a sum of Impulse functions.

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

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

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

Draw a signal x[n] = 0.5e-0.2nsin(nπ/9)u[n]

12

Energy and Power Signals

The total energy of a continuous time signal x(t) is defined as:

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

and its average power is

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

In the case of a discrete time signal x[nT], the total energy of the signal is

$$E_{dx} = T \sum_{n=-\infty}^{\infty} \left| x^2[n] \right|$$

and its average power is defined by

$$P_{dx} = \lim_{N\to\infty} \left(\frac{1}{2N+1} \right) \sum_{n=-N}^{N} \left| x[nT] \right|^2$$

Energy and Power Signals

- A signal is referred to as an energy signal, if and only if the total energy of the signal satisfies the condition $0 < E < \infty$
- On the other hand, it is referred to as a power signal, if and only if the average power of the signal satisfies the condition $0 < P < \infty$
- An energy signal has zero average power, whereas a power signal has infinite energy.
- Periodic signals and random signals are usually viewed as power signals, whereas signals that are both deterministic and non-periodic are energy signals.

Example:

Compute the signal energy and signal power for x[nT] = (-0.5)^nu(nT), T = 0.01 seconds

Solution:

$$E_{dx} = \lim_{N\to\infty} T \sum_{n=-N}^{N} |x(nT)|^2 = 0.01 \sum_{n=0}^{\infty} |(-0.5)^n|^2$$

$$= 0.01 \sum_{n=0}^{\infty} (-0.5)^{2n} = 0.01 \sum_{n=0}^{\infty} 0.25^n$$

$$= 0.01\left[1 + 0.25 + (0.25)^2 + (0.25)^3 + \ldots\ldots\right]$$

$$= \frac{0.01}{1 - 0.25} = 1/75$$

Since E_{dx} is finite, the signal power is zero.

Example:

Repeat Example3 for $y[nT] = 2e^{j3n}u[nT]$, T = 0.2 second.

Solution:

$$P_{dx} = \lim_{N\to\infty}\left(\frac{1}{2N+1}\right)\sum_{n=-N}^{N}|y(nT)|^2 = \lim_{N\to\infty}\left(\frac{1}{2N+1}\right)\sum_{n=0}^{N}\left|2e^{j3n}\right|^2$$

$$= \lim_{N\to\infty}\left(\frac{1}{2N+1}\right)\sum_{n=0}^{N} 2^2 = \lim_{N\to\infty}\frac{4}{2N+1}\sum_{n=0}^{N} 1 = \lim_{N\to\infty}\frac{4(N+1)}{2N+1}$$

$$= \lim_{N\to\infty} 4\left(\frac{N}{2N+1} + \frac{1}{2N+1}\right) = 4 \times \frac{1}{2} = 2$$

What is energy of this signal?

Tutorial 1:

Determine the signal energy and signal power for each of the given signals and indicate whether it is an energy signal or a power signal?

(a) $y[nT] = 3(-0.2)^n u[n-3],$ $T = 2\text{ ms}$

(b) $z[nT] = 4(1.1)^n u[n+1]$ $T = 0.02\text{ s}$

Ninevah University
College of Electronics Engineering
Control and Systems Engineering
Third Stage / 1st semester 2025-2026
Digital Signal Processing (DSP)

Lecture2: ADC (Sampling, Quantization, and Encoding)

Instructor:
Asst. Prof. Dr. Ahmed Jameel Abdulqader

Place and Date: Mosul / College of Electronics Engineering, /10/2025

Analog to Digital Conversion

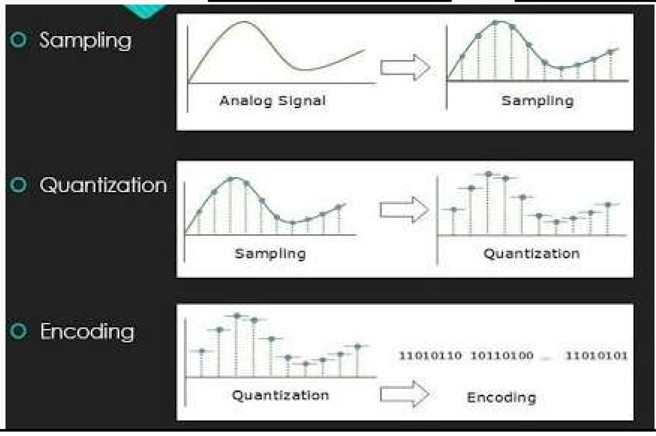
- A/D conversion can be viewed as a three-step process

Sampling: This is the conversion of a continuous time signal into a discrete time signal obtained by taking "**samples**" of the continuous time signal at discrete time instants. Thus, if x(t) is the input to the sampler, the output is x(nT), where T is called the **Sampling interval**.

Quantization: This is the conversion of discrete time continuous valued signal into a discrete-time discrete- value (digital) signal. The value of each signal sample is represented by a value selected from a finite set of possible values. The difference between unquantized sample and the quantized output is called the **Quantization error**.

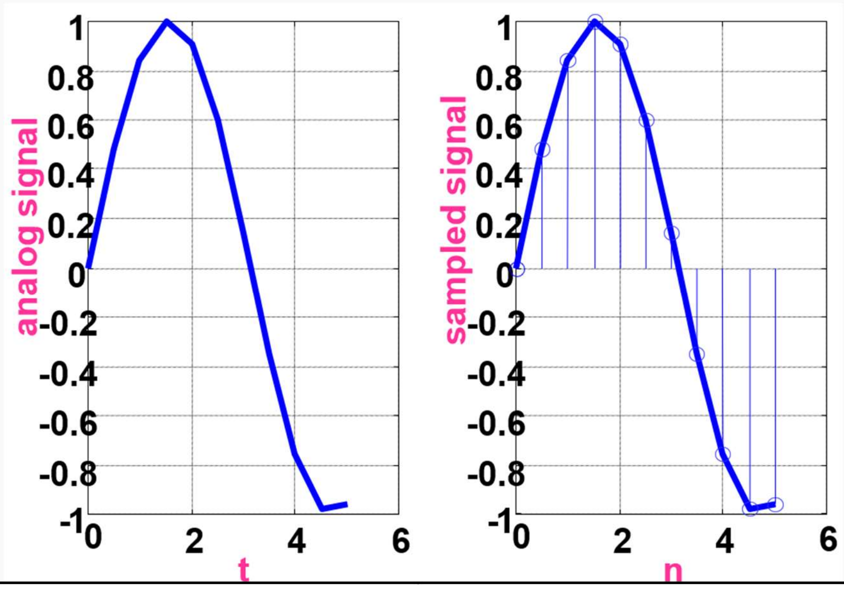
Analog to Digital Conversion (cont.)

Coding: In the coding process, each discrete value is represented by a b-bit binary sequence. A/D Converter



Sampling of Analog Signals

Uniform Sampling: $x[n] = x[nT]$



Uniform sampling

Uniform sampling is the most widely used sampling scheme.

This is described by the relation

$$x[n] = x[nT] \qquad -\infty < n < \infty$$

where x(n) is the discrete time signal obtained by taking samples of the analogue signal x(t) every T seconds

The time interval Ts between successive samples is called the Sampling Period or Sampling interval and its reciprocal **1/Ts = Fs** is called the Sampling Rate (samples per second) or the Sampling Frequency (Hertz).

A relationship between the time variables t and n of continuous time and discrete time signals respectively, can be obtained as **T = nTs = n/Fs**

Shannon's Sampling Theorem

- How frequently do we need to sample?
- The solution: Shannon's Sampling Theorem: A continuous-time signal x(t) with frequencies no higher than f_{max} can be reconstructed exactly from its samples x[n] = x(nTs), if the samples are taken a rate fs = 1 / Ts that is greater than **2** f_{max}.
- Note that the minimum sampling rate, **2** f_{max} , is called the **Nyquist rate**.
- Shannon's theorem tell us that if we have at **least 2 samples per period** of a sinusoid, we have enough information to reconstruct the sinusoid.
- What happens if we sample at a rate which is less than the Nyquist Rate?
  - **Aliasing will occur!!!!**

Sampling Theorem

- Signal sampling at a rate less than the Nyquist rate is referred to as **undersampling**.
- Signal sampling at a rate greater than the Nyquist rate is known as the **oversampling**.

Example 1:

For a continuous time signals **x(t)= 8cos(200 π t).** Find
1- Minimum sampling rate.
2- If **Fs=400Hz**, what is the discrete time signal?
3- If **Fs=150Hz**, what is the discrete time signal?
4- Comment on result obtained in 2 and 3 with details.

Sampling Theorem

Consider the continuous signal

x(t) = 3cos100πt

(a) Determine the minimum required sampling rate to avoid aliasing.

(b) Suppose that the signal is sampled at the rate Fs = 200 Hz. What is the discrete time signal obtained after sampling?

Solution:

(a) The frequency of the continuous signal is F = 50 Hz. Hence the minimum sampling rate to avoid aliasing is 100Hz.

(b) $$x[n] = 3\cos\frac{100\pi}{200}n = 3\cos\frac{\pi}{2}n$$

Sampling Theorem

Example 3

Consider the continuous signal

$x(t) = 3\cos 50\pi t + 10\sin 300\pi t - \cos 100\pi t$

What is the Nyquist rate for this signal.

Solution:

The frequencies present in the signal above are

$F1 = 25$ Hz, $F2 = 150$ Hz $F3 = 50$ Hz.

Thus Fmax = 150 Hz.

Nyquist rate = 2.Fmax = 300 Hz.

Tutorial

Q1: Find the minimum sampling rate that can be used to obtain samples that completely specify the signals:

(a) $x(t) = 10\cos(20\pi t) – 5\cos(100 \pi t) + 20\cos(400 \pi t)$

(b) $y(t) = 2\cos(20 \pi t) + 4\sin(20 \pi t - \pi /4) + 5\cos(8 \pi t)$

Q2: Consider the continuous signal

$x(t) = 3\cos 2000 \pi t + 5\sin 6000 \pi t + 10\cos 12000 \pi t$

(a) What is the Nyquist rate for this signal?

(b)Assume now that we sample this signal using a sampling rate Fs = 5000 samples/s. What is the discrete time signal obtained after sampling?

Tutorial

Example

Suppose continuous-time signal $x(t)$ = cos($w_0 t$) is sampled at a sampling frequency of 1000Hz to produce $x[n]$

$$x[n] = \cos\left(\frac{\pi n}{4}\right)$$

- Determine 2 possible positive values of w_0, say w_1 and w_2 .
- Discuss if cos(w_1 t) or cos(w_2 t) will be obtained when passing $x[n]$ through the DC converter .

Sol: With T= 1/1000 s : $\cos\left(\frac{\pi n}{4}\right) = x[n] = x(nT) = \cos\left(\frac{w_0 n}{1000}\right)$

w_1 is easily computed as:

$$\frac{\pi n}{4} = \frac{w_1 n}{1000} = w_1 = \frac{1000\pi}{4} = 250\pi$$

w_2 can be obtained by noting the periodicity of a sinusoid :

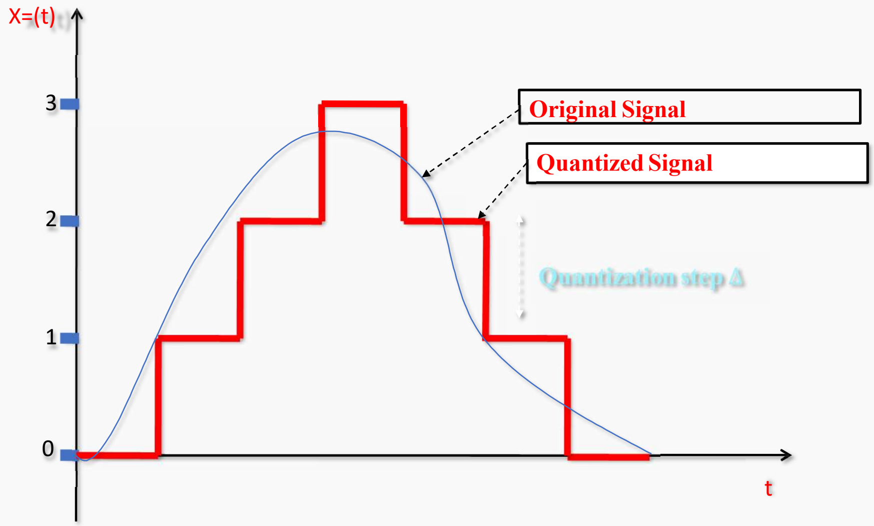
$$\cos\left(\frac{\pi n}{4}\right) = \cos\left(\frac{\pi n}{4} + 2n\pi\right) = \cos\left(\frac{9\pi n}{4}\right) = \cos\left(\frac{w_2}{1000}\right)$$

As a result , we have

$$\frac{9\pi n}{4} = \frac{n_2 n}{1000} = w_2 = \frac{9000\pi}{4} = 2250\,\pi$$

There are 2001 samples in 0.002 s and interpolating the successive based on plot yields good approximations

Quantization



- It is quite apparent that the quantized signal is not exactly the same as the original analog signal.
- There is a fair degree of quantization error here.

Quantization step Δ ⇨ Quantization resoluation

Quantization

The notations and general rules for quantization are as follows:

$$\Delta = \frac{(x_{max} - x_{min})}{L}$$

The symbol Δ is the step size of the ADC resolution.
Where:

x_{max} and x_{min} are the maximum and minimum values, respectively, of the analogue input signal x

The symbol L denotes the number of quantization levels, which is determined by equation

$$L = 2^m$$

Where m is the number of bits used in ADC

$$i = round\left(\frac{x - x_{min}}{\Delta}\right)$$

Quantization

$$x_q = x_{min} + i\Delta \qquad i = 0, 1, \cdots, L-1$$

x_q indicates the quantization level ,
i is an index corresponding to the binary code .
When the DAC outputs the analog amplitude x_q with finite precision , it introduces quantization error defined as:-

$$e_q = x_q - x$$

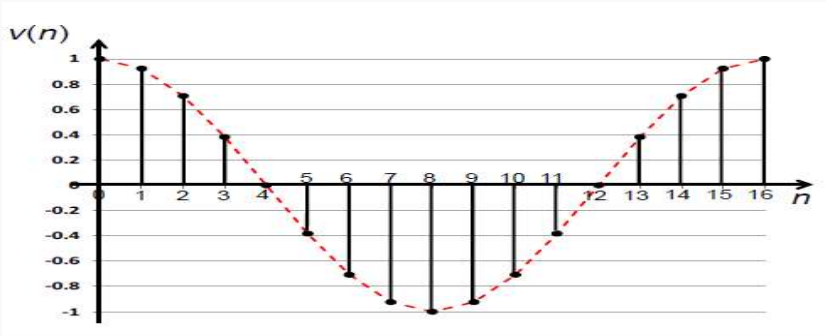
Note :
It is important that the quantization errors due to rounding is within the range

$$-\frac{\Delta}{2} \le e_q(n) \le \frac{\Delta}{2}$$

Quantization

Example :

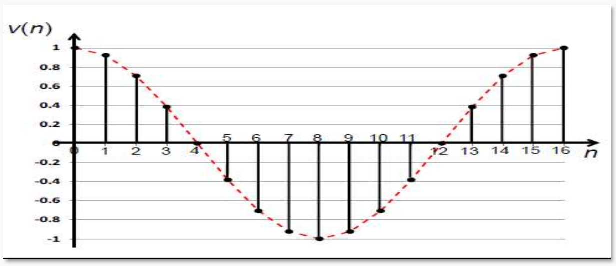
The discrete – time signal $V(n) = \cos\left(\frac{2\pi}{16}\right)$ below can be quantized by mapping each value of $V(n)$ to the 11 quantization levels { -1, -0.8 , -0.6 , …… , 0.6 , 0.8 , 1 } as shown in the figure .

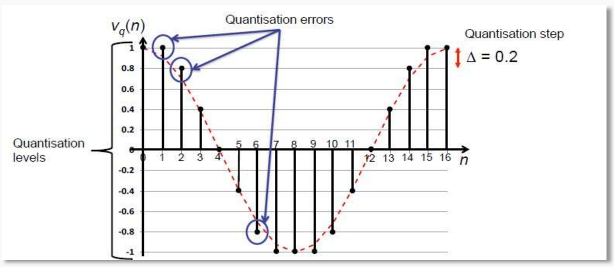


L = 11 so the minimum length of the binary vectors must be 4 bits (since $2^4 > 11$)

Quantization

Some quantization errors can clearly be seen on the digital signal below





Encoding

Therefore, for the quantized sample values
1, 1, 0.8, 0.4, 0,…….
The binary sequence will be
0101 0101 0100 0010 0000 …
This completes the conversion of
a continuous signal to a binary signal

1	**0101**
0.8	0100
0.6	0011
0.4	0010
0.2	0001
0	0000
-0.2	1001
-0.4	1010
-0.6	1011
-0.8	1100
-1	1101

Tutorial

Example1

Assuming that a 3-bit ADC channel accepts analog input ranging from 0 to 5 volts , determine

a. The number of quantization levels
b. The step size of the quantizer or resolution
c. The quantization level when the analog voltage is 3.2 volts
d. The binary code produced by the ADC .

Solution :

since the range is from 0 to 5 volts a 3-bit ADC is used , we have

$x_{min} = 0$ volt , $x_{max} = 5$ volts , and
m=3 bits

a. Using equation of quantization level, we get the number of quantization levels as

L = $2^m = 2^3 = 8$

b. Applying Equation yields

$\Delta = \frac{5-0}{8} = 0 \cdot 625$ volt

c. When x= 3.2, from equation we get

i = round ($\frac{x - x_{min}}{\Delta}$) = round (5.12) = 5

From equation, we determine the quantization level as

$x_q = 0 + 5\Delta = 5 \times 0.625 = 3.125$ volts

Tutorial

Example2

Using the previous example, determine the quantization error when the analog input is 3.2 volts .

Solution :
Using equation , we obtain

$$e_{\mathrm{q}} = x_q - x = 3.125 - 3.2 = -0.075\ volt$$

Note that the quantization error is less than the half of the step size , that is

$$e_q \;= 0.075 < \Delta \mathrm{T}2 = 0.3125 \text{ volt}$$

Ninevah University
College of Electronics Engineering
Control and Systems Engineering
Third Stage / 1st semester 2025-2026
Digital Signal Processing (DSP)

Lecture3: Discrete Time Systems

Instructor:
Asst. Prof. Dr. Ahmed Jameel Abdulqader

Place and Date: Mosul / College of Electronics Engineering, /10/2025

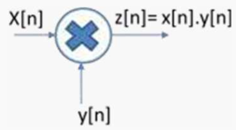
Discrete Time Systems

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

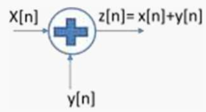
  where the symbol H denotes the transformation (also called an **operator**), or processing performed by the system on x[n] to produce y[n].

Operations on Sequences

Modulation



Addition



Scaling

Time Shifting

Discrete Time Systems

Example: consider the following sequences of length (5) defined for (0≤n≤4).
x[n] ={ 3.2 41 36 -9.5 0 }
y[n] ={ 1.7 -0.5 0 0.8 1 }

Find

a) x[n] . y[n]

b) x[n] + y[n]

c) 7/2 x[n]

solution:

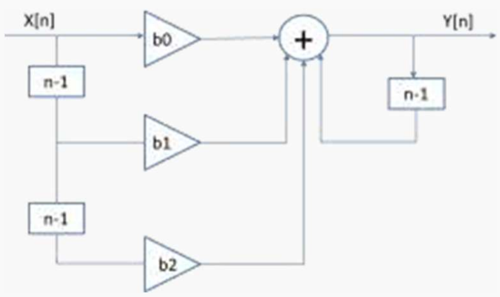
x[n] . y[n]={ 5.44 -20.5 0 -7.6 0 }

x[n]+y[n]={ 4.9 40.5 36 -8.7 1}

7/2x[n]={ 11.2 143.5 126 -33.25 0}

Example: Analyze the discrete-time system shown below to determine the sequence y[n].

Solution:
y[n]= b_0x[n] +b_1x[n-1] +b_2x[n-2] +y[n-1]
This formula is known as ((***difference equation***))



Discrete -Time Systems (Digital Processors):-

The function of a discrete - time system is to process a given input sequence to generate an output sequence.

Classification of Discrete Time Systems

The classification of DTS is based on the input - output relation of the system.

Classification of Discrete Time Systems

1- Linear System: It is the system for which the superposition principle always holds.

2- Shift -Invariant System: (Time - invariant system)

If y[n] is the response to an input x[n], then the response to x $[n-n_0]$ is y $[n-n_0]$

3- Linear Time-Invariant System: (LTI)

It is the system that satisfies both the linearity and the time - invariance properties. Such systems are mathematically easy to analyze ,and easy to design

4- Static and Dynamic System

5- Causal System

In causal system, the output signal depends only on present and /or previous values of the input. The practical signal processors are always causal, because they cannot anticipate the future

6- Invertible System

If a digital system with input x[n] gives an output y[n], then its inverse would produce x [n] if fed with y [n]. Most practical systems are invertible.

The LTI systems are also causal and invertible.

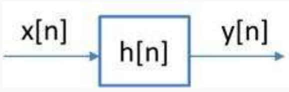
Time Domain Analysis

Introduction

In this topic we develop the basic techniques for describing digital signals in the time domain. Such techniques are: impulse response, step response, and digital convolution.

1. The Impulse Response:

The response of digital system to sequence ($x[n]=\delta[n]$) is called the **unit sample response** or simply "**the impulse response**", and is denoted as ($h[n]$).



The Impulse Response

Example: Find the impulse response of the system :

$$y[n]=\frac{1}{3}x[n+1]+\frac{1}{3}x[n]+\frac{1}{3}x[n-1]$$

Sol : we set $x[n] = \delta[n]$

$$y[n]=h[n]=\frac{1}{3}\delta[n+1]+\frac{1}{3}\delta[n]+\frac{1}{3}\delta[n-1]$$

for $n=-2$ $\quad y[n]=\frac{1}{3}\delta[-1]+\frac{1}{3}\delta[-2]+\frac{1}{3}\delta[-3]=0$

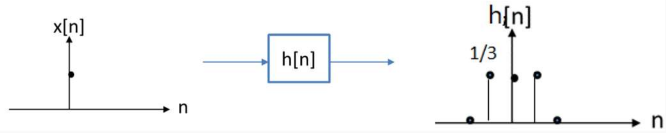
for $n=-1$ $\quad y[n]=\frac{1}{3}\delta[0]+\frac{1}{3}\delta[-1]+\frac{1}{3}\delta[-2]=1/3$

for $n=0$ $\quad y[n]=\frac{1}{3}\delta[1]+\frac{1}{3}\delta[0]+\frac{1}{3}\delta[-1]=1/3$

for $n=1$ $\quad y[n]=\frac{1}{3}\delta[2]+\frac{1}{3}\delta[1]+\frac{1}{3}\delta[0]=1/3$

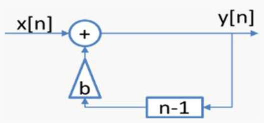
for $n=2$ $\quad y[n]=\frac{1}{3}\delta[3]+\frac{1}{3}\delta[2]+\frac{1}{3}\delta[1]=0$

for $n\le-2$ and $n\ge2$ ➡ $y[n]=0$



The Impulse Response

Example : Find the impulse response for the system shown below. Given **b=-0.9**



Sol:

$y[n]=-0.9\,y[n-1]+x[n]$

the impulse response $=h[n]=-0.9\,h[n-1]+\delta[n]$

$h[-1]=-0.9\,h[-2]+\delta[-1]=0+0=0$

$h[0]=-0.9\,h[-1]+\delta[0]=0+1=1=(-0.9)^0$

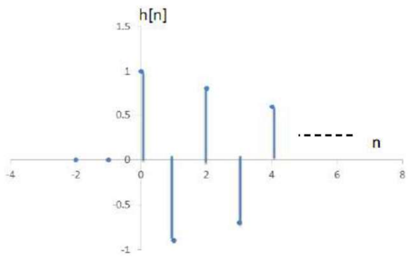
$h[1]=-0.9\,h[0]+\delta[1]=-0.9 \quad =(-0.9)^1$

$h[2]=-0.9\,h[1]+\delta[2]=0.81 \quad =(-0.9)^2$

$h[3]=-0.9\,h[2]+\delta[3]=-0.729 \quad =(-0.9)^3$

$h[4]=-0.9\,h[3]+\delta[4]=0.656 \quad =(-0.9)^4$

⋮ ⋮ ⋮



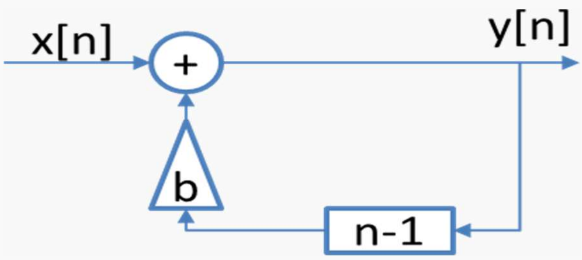
we can also find that $h[n]=(-0.9)^n\,u[n]$ or in general: $\mathbf{h[n]=b^n\,u[n]}$

2. The Step Response:

The response of a discrete-time system to a unit step sequence(x[n]=u[n]) is called the **unit step response** or simply the "**step response**", and is denoted as **S[n]**.

Example:

Find and sketch the step response for the system shown below. Given **b=0.8**.
•Find the response to the rectangular pulse input bandlimited by ($0\le n\le 3$).



The Step Response:

Sol: a) y[n]=0.8 y[n-1] + x[n]

For n<0 y[n]=0

For n=0 y[0]=0.8 y[-1] + x[0]=0+1=1

For n=1 y[1]=0.8 y[0] + x[1]=0.8(1)+1=1.8

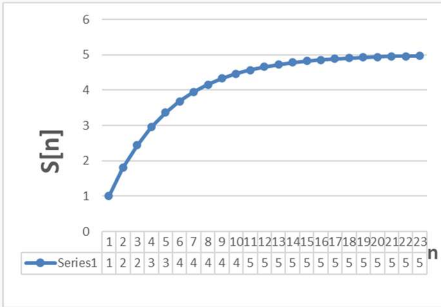
For n=2 y[2]=0.8 y[1] + x[2]=0.8(1.8)+1=2.44

For n=3 y[3]=0.8 y[2] + x[3]=0.8(2.44)+1=2.952

For n=∞ $y[\infty]=1+0.8^1+0.8^2+0.8^3+.......+0.8^\infty$

$=0.8^0+0.8^1+0.8^2+0.8^3+.......+0.8^\infty$

$=\sum_{n=0}^{\infty}(0.8)^n = \frac{1}{1-0.8} = 5$ =steady state value



The Step Response:

b) y[n]=0.8 y[n-1] +x[n]

for n<0 y[n]=0

n=0 y[n]=1

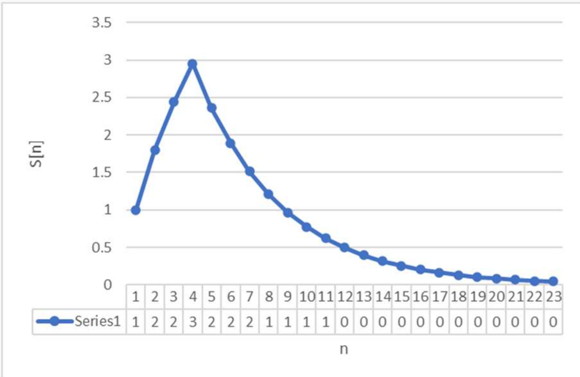
n=1 y[n]=1.8

n=2 y[n]=2.44

n=3 y[n]=2.952

n=4 y[n]=2.362

n=5 y[n]=1.89



- **Note** that increasing the value of **b** will increase the durationof the transient (the rise time).
- **Transient response:** it is the part of a response that vanishesas sample number approaches infinity.
- **Steady state response:** it is the part of the response that does not vanish as sample number approaches infinity.

Tutorial

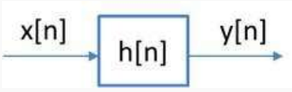
1- Find the first three sample values of the impulse response h[n]for the system defined by the following difference equation $y[n]= 1.5\, y[n-1] - 0.85\, y[n-2] + x[n]$ Assuming the system is a causal.

2- Find the first four sample values of the impulse response h[n] for the system defined by the following equation

$$y[n] = x[n] + x[n-1] + x[n-2] + \cdots$$

Discrete Time Convolution

We have seen how to characterize LTI processors by their impulse or step responses. In practical cases, we need a general computer-based method to estimate a system's response to any form of input signal. The method which will do this is known as **"digital convolution"**



=x[n]*h[n]

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

Which can be alternately as $\boldsymbol{y[n]} = \sum_{\boldsymbol{k}=-\infty}^{\infty} \boldsymbol{x[n-k]\ \ h[k]}$

Discrete Time Convolution

Example 1: convolution of two finite-duration sequence:

$$x[n] = \begin{cases} 1 & for \quad -1 \le n \le 1 \\ 0 & otherwise \end{cases} \qquad h[n] = \begin{cases} 1 & for \quad -1 \le n \le 1 \\ 0 & otherwise \end{cases}$$

Sol:

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

for n=-3 y[-3]=$x[-1]\ h[-2] + x[0]\ h[-3] + x[1]\ h[-4]$=0

for n=-2 y[-2]=$x[-1]\ h[-1] + x[0]\ h[-2] + x[1]\ h[-3]$=1

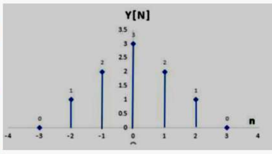
for n=-1 y[-1]=$x[-1]\ h[0] + x[0]\ h[-1] + x[1]\ h[-2]$=2

for n=0 y[0]=$x[-1]\ h[1] + x[0]\ h[0] + x[1]\ h[-1]$=3

for n=1 y[1]=$x[-1]\ h[2] + x[0]\ h[1] + x[1]\ h[0]$= 2

for n=2 y[2]=$x[-1]\ h[3] + x[0]\ h[2] + x[1]\ h[1]$= 1

for n=3 y[3]=$x[-1]\ h[4] + x[0]\ h[3] + x[1]\ h[2]$= 0



Discrete Time Convolution

Example 2: Find x[n] * h[n] where:

x[n]=[1 2 3 -1]
↑
h[n]=[1 -1 2]

There are three solution methods

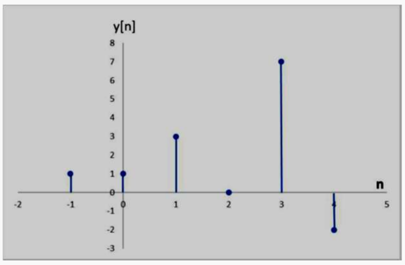
Sol 2: Also this problem can be solved using **the multiplication method** as shown below:

```
x[n]        1  2  3 -1
h[n]     ×  1 -1  2
-----------------------
            2  4  6 -2
        -1 -2 -3  1
  +   1  2  3 -1
-----------------------
y[n]= 1  1  3  0  7 -2
         ↑
```

**and it can also be solved by using the 4th method tabulation method

	h[0]	h[1]	h[2]
x[-1]	1	-1	2
x[0]	2	-2	4
x[1]	3	-3	6
x[2]	-1	1	-2

Y[n]	Y[-1]	Y[0]	Y[1]	Y[2]	Y[3]	Y[4]
=	1	1	3	0	7	-2



Discrete Time Convolution

Digital Convolution Properties

The convolution operation satisfies several useful properties:

1. Commutative: $x_1[n] \otimes x_2[n] = x_2[n] \otimes x_1[n]$
2. Associative: $(x_1[n] \otimes x_2[n]) \otimes x_3[n] = x_1[n] \otimes (x_2[n] \otimes x_3[n])$
3. Distributive: $x_1[n] \otimes (x_2[n] + x_3[n]) = x_1[n] \otimes x_2[n] + x_1[n] \otimes x_3[n]$

Discrete Time Convolution

H.W: Find the convolution

1- $x[n]=[\underset{\uparrow}{0}\ 1\ -2\ 3\ -4]$, $h[n]=[0.5\ 1\ \underset{\uparrow}{2}\ 1\ 0.5]$

2- $x[n]= 0.5n[u[n]-u[n-6]]$, $h[n]=2\sin(n\pi/2)[u[n+3]-u[n-4]]$

Ninevah University
College of Electronics Engineering
Control and Systems Engineering
Third Stage / 1st semester 2025-2026
Digital Signal Processing (DSP)

Lecture3: Frequency Analysis of Discrete-Time Signals

Instructor:
Asst. Prof. Dr. Ahmed Jameel Abdulqader

Place and Date: Mosul / College of Electronics Engineering, /10/2025

The Fourier Series and Fourier Transform for Discrete-Time Signals

- The Fourier series is used to represent a periodic function by a discrete sum of complex exponentials.
- While the Fourier transform is then used to represent a general, nonperiodic function by a continuous superposition or integral of complex exponentials.

Discrete Fourier Series (DFS)

Suppose that we are given a periodic sequence with period N. The Fourier series representation for x[n] consists of N harmonically related exponential functions

$e^{j2\pi kn/N}$, k = 0, 1,2,……,N-1

and is expressed as

$$x[n] = \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N}$$

where the coefficients c_k can be computed as:

$$c_k = \frac{1}{N} \sum_{n=0}^{\infty} x[n] e^{-j2\pi kn/N}$$

C_k is spectra or DFS coefficient

Discrete Fourier Series (DFS)

Example: Determine the spectra of the following signals:

(a) x[n] = [1, 1, 0, 0], x[n] is periodic with period of 4 **(b)** x[n] = cosπn/3

Solution: (a) $x[n] = [\underset{\uparrow}{1}, 1, 0, 0]$

$$\because c_k = \frac{1}{N}\sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} = \frac{1}{4}\sum_{n=0}^{3} x[n] e^{-j2\pi kn/N}$$

Now $c_0 = \frac{1}{4}\sum_{n=0}^{3} x[n] = \frac{1}{4}\left[x[0]+x[1]+x[2]+x[3]\right] = \frac{1}{4}[1+1+0+0] = \frac{1}{2}$

$$c_1 = \frac{1}{4}\sum_{n=0}^{3} x[n] e^{-j2\pi n/4} = \frac{1}{4}\sum_{n=0}^{3} x[n] e^{-j\pi n/2} = \frac{1}{4}\left[x[0]+x[1]e^{-j\pi/2}+0+0\right]$$

$$= \frac{1}{4}\left[1+1\left(\cos\tfrac{\pi}{2} - j\sin\tfrac{\pi}{2}\right)\right] = \frac{1}{4}\left[1+(0-j)\right] = \frac{1}{4}(1-j)$$

Discrete Fourier Series (DFS)

$$c_2 = \frac{1}{4}\sum_{n=0}^{3} x[n]e^{j2\pi 2n/4} = \frac{1}{4}\sum_{n=0}^{3} x[n]e^{j\pi n} = \frac{1}{4}\left[1 + 1.e^{j\pi}\right]$$

$$= \frac{1}{4}\left[1 + \cos\pi - j\sin\pi\right] = 0$$

$$c_3 = \frac{1}{4}\sum_{n=0}^{3} x[n]e^{-j2\pi n3/4} = \frac{1}{4}\left[1 + \cos(3\pi/2) - j\sin(3\pi/2)\right] = \frac{1}{4}\left[1 + 0 + j\right] = \frac{1}{4}\left[1 + j\right]$$

The magnitude spectra are:

$$|c_0| = \frac{1}{2} \quad |c_1| = \frac{\sqrt{2}}{4} \quad |c_2| = 0 \quad |c_3| = \frac{\sqrt{2}}{4}$$

and the phase spectra are:

$$\Phi_0 = 0 \quad \Phi_1 = \frac{-\pi}{4} \quad \Phi_2 = undefined \quad \Phi_3 = \frac{\pi}{4}$$

Discrete Fourier Series (DFS)

(b) x[n] = cosπn/3

Solution: In this case, $f_0 = 1/6$ and hence x[n] is periodic with fundamental period N = 6.

Now

$$c_k = \frac{1}{6}\sum_{n=0}^{5} x[n]e^{-j2\pi kn/N} = \frac{1}{6}\sum_{n=0}^{5}\cos\frac{\pi n}{3}e^{-j2\pi kn/6} = \frac{1}{6}\sum_{n=0}^{5}\cos\frac{\pi n}{3}e^{-j\pi kn/3}$$

$$= \frac{1}{6}\sum_{n=0}^{5}\frac{1}{2}\left[e^{j\pi n/3} + e^{-j\pi n/3}\right]e^{-j\pi kn/3} = \frac{1}{12}\sum_{n=0}^{5}\left[e^{j\frac{\pi n}{3}(1-k)} + e^{-j\frac{\pi n}{3}(1+k)}\right]$$

$$\therefore c_0 = \frac{1}{12}\sum_{n=0}^{5} 2\cos\frac{\pi n}{3} = \frac{1}{6}\sum_{n=0}^{5}\cos\frac{\pi n}{3}$$

$$= \frac{1}{6}\left[\cos 0 + \cos\tfrac{\pi}{3} + \cos\tfrac{2\pi}{3} + \cos\tfrac{3\pi}{3} + \cos\tfrac{4\pi}{3} + \cos\tfrac{5\pi}{3}\right] = 0$$

Similarly, $c_2 = c_3 = c_4 = 0$, $c_1 = c_5 = ½$.

Discrete Fourier Series (DFS)

Example: The periodic signal

$$x(t) = \sin(2\pi t)$$

is sampled using the sampling rate $f_s = 4$ Hz.

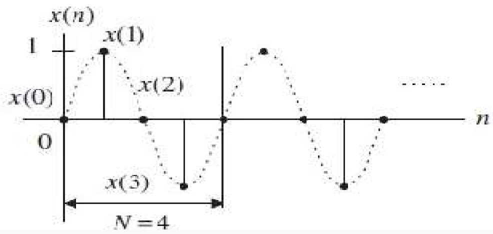
a. Compute the spectrum c_k using the samples in one period.
b. Plot the two-sided amplitude spectrum $|c_k|$ over the range from -2 to 2 Hz.

Solution:

a. From the analog signal, we can determine the fundamental frequency $\omega_0 = 2\pi$ radians per second and $f_0 = \frac{\omega_0}{2\pi} = \frac{2\pi}{2\pi} = 1$ Hz, and the fundamental period $T_0 = 1$ second.
Since using the sampling interval $T = 1/f_s = 0.25$ second, we get the sampled signal as

$$x(n) = x(nT) = \sin(2\pi nT) = \sin(0.5\pi n)$$

and plot the first eight samples as shown in Figure 4.4.



Discrete Fourier Series (DFS)

$$e^{-j2\pi/N} = \cos\left(\frac{2\pi}{N}\right) - j\sin\left(\frac{2\pi}{N}\right).$$

Choosing the duration of one period, $N = 4$, we have the sample values as follows

$$x(0) = 0;\ x(1) = 1;\ x(2) = 0;\ \text{and}\ x(3) = -1.$$

$$c_0 = \frac{1}{4}\sum_{n=0}^{3} x(n) = \frac{1}{4}(x(0) + x(1) + x(2) + x(3)) = \frac{1}{4}(0 + 1 + 0 - 1) = 0$$

$$c_1 = \frac{1}{4}\sum_{n=0}^{3} x(n)e^{-j2\pi\times 1n/4} = \frac{1}{4}\left(x(0) + x(1)e^{-j\pi/2} + x(2)e^{-j\pi} + x(3)e^{-j3\pi/2}\right)$$

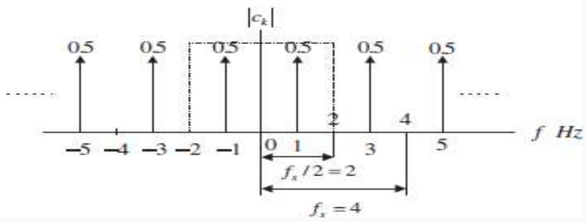
$$= \frac{1}{4}(x(0) - jx(1) - x(2) + jx(3) = 0 - j(1) - 0 + j(-1)) = -j0.5.$$

Similarly, we get

$$c_2 = \frac{1}{4}\sum_{k=0}^{3} x(n)e^{-j2\pi\times 2n/4} = 0,\ \text{and}\ c_3 = \frac{1}{4}\sum_{n=0}^{3} x(k)e^{-j2\pi\times 3n/4} = j0.5.$$

Discrete Fourier Series (DFS)

b. The amplitude spectrum for the digital signal is sketched in Figure 4.5.



As we know, the spectrum in the range of -2 to 2 Hz presents the information of the sinusoid with a frequency of 1 Hz and a peak value of $2|c_1|=1$, which is converted from two sides to one side by doubling the spectral value. Note that we do not double the direct-current (DC) component, that is, c_o.

Discrete Fourier Transform (DFT)

We determine the Fourier series coefficients using one-period ***N*** data samples and previous equation Then we multiply the Fourier series coefficients by a factor of ***N*** to obtain.

$$X(K) = NC_k = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi kn}{N}} \qquad , K = 0, 1, \ldots. N-1$$

Where ***X(k)*** constitutes the DFT coefficients. Notice that the factor of N is a constant and does not affect the relative magnitudes of the DFT coefficients ***X(K)***. As shown in the last plot, applying DFT with ***N*** data samples of ***x(n)*** sampled at a rate of *fs* (sampling period is T=1/ *fs*) produces ***N*** complex DFT Coefficients X(k).

Discrete Fourier Transform (DFT)

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi kn}{N}} = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad , for\ k = 0, 1, \ldots, N-1$$

This equation can be expanded as:-

$X(k)$ = x(0) W_N^{k0} +x(1) W_N^{k1} +x(2) W_N^{k2} +…….+x(N-1) $W_N^{k(N-1)}$ for k=0,1,…..,N-1

Where the factor w_N (called the twiddle factor in some textbooks) is defined as:-

$$W_N = e^{-j\frac{2\pi kn}{N}} = \cos\left(\frac{2\pi}{N}\right) - j\sin\left(\frac{2\pi}{N}\right)$$

Discrete Fourier Transform (DFT)

Example:

Given a sequence $x(n)$ for $0 \le n \le 3$, where $x(0) = 1$, $x(1) = 2$, $x(2) = 3$, and $x(3) = 4$,

a. Evaluate its DFT $X(k)$.

Solution:

a. Since $N = 4$ and $W_4 = e^{-j\frac{\pi}{2}}$, using Equation (4.7) we have a simplified formula,

$$X(k) = \sum_{n=0} x(n)W_4^{kn} = \sum_{n=0} x(n)e^{-j\frac{\pi kn}{2}}.$$

Thus, for $k = 0$

$$X(0) = \sum_{n=0}^{3} x(n)e^{-j0} = x(0)e^{-j0} + x(1)e^{-j0} + x(2)e^{-j0} + x(3)e^{-j0}$$
$$= x(0) + x(1) + x(2) + x(3)$$
$$= 1 + 2 + 3 + 4 = 10$$

for $k = 1$

$$X(1) = \sum_{n=0}^{3} x(n)e^{-j\frac{\pi n}{2}} = x(0)e^{-j0} + x(1)e^{-j\frac{\pi}{2}} + x(2)e^{-j\pi} + x(3)e^{-j\frac{3\pi}{2}}$$
$$= x(0) - jx(1) - x(2) + jx(3)$$
$$= 1 - j2 - 3 + j4 = -2 + j2$$

Discrete Fourier Transform (DFT)

for $k = 2$

$$X(2) = \sum_{n=0}^{3} x(n)e^{-j\pi n} = x(0)e^{-j0} + x(1)e^{-j\pi} + x(2)e^{-j2\pi} + x(3)e^{-j3\pi}$$
$$= x(0) - x(1) + x(2) - x(3)$$
$$= 1 - 2 + 3 - 4 = -2$$

and for $k = 3$

$$X(3) = \sum_{n=0}^{3} x(n)e^{-j\frac{3\pi n}{2}} = x(0)e^{-j0} + x(1)e^{-j\frac{3\pi}{2}} + x(2)e^{-j3\pi} + x(3)e^{-j\frac{9\pi}{2}}$$
$$= x(0) + jx(1) - x(2) - jx(3)$$
$$= 1 + j2 - 3 - j4 = -2 - j2$$

Discrete Fourier Transform (DFT)

The Fourier Transform of a finite energy discrete time signal x[n] is defined as

The following table shows some useful DFT pairs:

X[n]	X[k]
δ[n]	1
δ[n-a]	$e^{\frac{-j2\pi k}{N}a}$
$a^n u[n]$	$\frac{1}{1-ae^{\frac{-j2\pi k}{N}}}$
u[n]	$\frac{1}{1-e^{\frac{-j2\pi k}{N}}}$
$e^{an}u[n]$	$\frac{1}{1-e^{a}e^{\frac{-j2\pi k}{N}}}$

Properties of Discrete Fourier Transform

1- Linearity:

DFT ($Ax_1[n] + Bx_2[n]$) = $AX_1[k] + BX_2[k]$

2- Convolution:

DFT ($x[n] \otimes y[n]$) = X[k] . Y[k]

3- Modulation:

DFT (x[n] . y[n]) = $X[k] \otimes Y[k]$

4- Periodicity:

X[k] = X[k+N]

5- Time Shifting:

DFT (x[n-a]) = X[k] $e^{-\frac{j2\pi k\ a}{N}}$

6- Frequency Shifting:

DFT (x[n] $e^{-\frac{j2\pi k\ a}{N}}$) = X[(k-a)N] = $\begin{cases} x[k-a] & a \le k \le N-1 \\ x[k-a+N] & 0 \le k \le a \end{cases}$

7- Parseval's Theorem

The power in discrete time domain is the same one in the discrete frequency domain or

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |x[k]|^2$$

The inverse DFT (IDFT):

To transfer the frequency response into the corresponding discrete time sequence, we use the following formula:

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

The inverse DFT (IDFT):

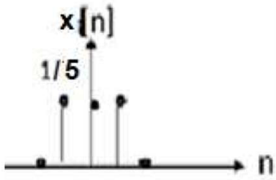
Ex: compute the IDFT for $H[k] = \frac{1+2\cos\frac{2\pi k}{N}}{5}$

Sol: we have $h[n] = \frac{1}{N}\sum_{n=0}^{N-1} H[k]\, e^{j2\pi kn/N}$

Or we can directly find the IDFT from the table

Since x[k]=$\frac{1}{5}+\frac{2}{5}\left(\frac{e^{\frac{j2\pi k}{N}}+e^{-\frac{j2\pi k}{N}}}{2}\right)$

$=\frac{1}{5}+\frac{1}{5}e^{\frac{j2\pi k}{N}}+\frac{1}{5}e^{-\frac{j2\pi k}{N}}$

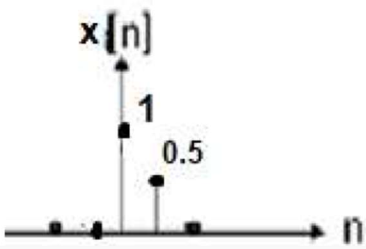


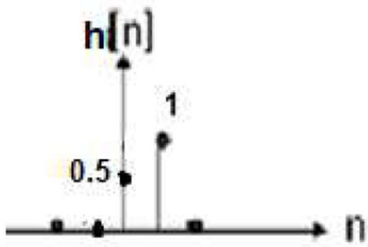
x[n]= $\frac{1}{5}\delta[n]+\frac{1}{5}\delta[n+1]+\frac{1}{5}\delta[n-1]$

The inverse DFT (IDFT):

Ex: Perform the linear convolution with DFT.

$$x[n] = \begin{cases} 1 & for\ n = 0 \\ 0.5 & for\ n = 1 \\ 0 & otherwise \end{cases} \qquad h[n] = \begin{cases} 0.5 & for\ n = 0 \\ 1 & for\ n = 1 \\ 0 & otherwise \end{cases}$$





The inverse DFT (IDFT):

Sol.

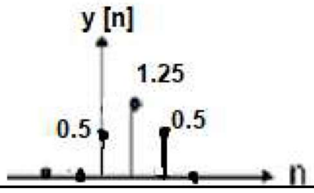
$$X[k] = \sum_{n=0}^{1} x[n]\, e^{-\frac{j2\pi kn}{N}}$$

$= x[0]+x[1]\, e^{-\frac{j2\pi k}{N}}$

$X[k]=1+0.5e^{-\frac{j2\pi k}{N}}$ Also $H[k]=0.5+e^{-\frac{j2\pi k}{N}}$

$Y[k] = X[k] \,.\, H[k] = 0.5+1.25e^{-\frac{j2\pi k}{N}} + 0.5e^{-\frac{j2\pi k(2)}{N}}$

$y[n] = IDFT\ (Y[k]) = 0.5\delta[n] + 1.25\ \delta[n-1] + 0.5\delta[n-2]$



The inverse DFT (IDFT):

Example : Find the DFT of the following sequence

[1 0 0 1]

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

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

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

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

$$X[2] = \sum_{n=0}^{3} x[n]e^{-j\pi n} = x[0]+x[3]e^{-j3\pi} = 1+1.[\cos(3\pi) - j\sin(3\pi)] = 0$$

The inverse DFT (IDFT):

Example : Find the IDFT of the sequence

[2 1+j 0 1-j]

Sol.:

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

Now $x[0] = \frac{1}{4}\sum_{k=0}^{N-1} X[k] = \frac{1}{4}\left[X[0] + X[1] + X[2] + X[3]\right]$

$$x[1] = \frac{1}{4}\sum_{k=0}^{3} X[k]e^{jk2\pi/4} = \frac{1}{4}\sum_{k=0}^{3} X[k]e^{jk\pi/2}$$

$$= X[0] + X[1]e^{j\pi/2} + X[2]e^{j\pi} + X[3]e^{j3\pi/2} = 0$$

Similarly,

X[2] = 0 and X[3] = 1

Ninevah University
College of Electronics Engineering
Control and Systems Engineering
Third Stage / 1st semester 2025-2026
Digital Signal Processing (DSP)

Lecture5: Fast Fourier Transform

Instructor:
Asst. Prof. Dr. Ahmed Jameel Abdulqader

Place and Date: Mosul / College of Electronics Engineering, /11/2025

Fast Fourier Transform (FFT)

FFT is a very efficient algorithm in computing DFT coefficients and can reduce a very large amount of computational complexity (multiplications). Without loss of generality.

Cooley and Tukey found that the DFT operation could be decomposed into a number of other DFTs of shorter lengths.

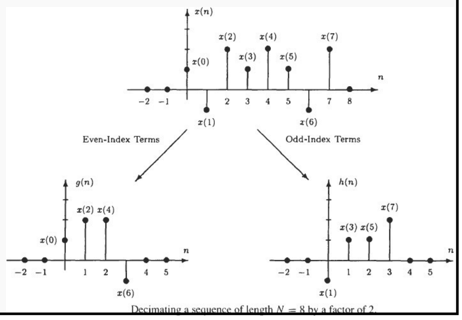
we consider the digital sequence *x(n)* consisting of 2^m samples, where m is a positive integer—the number of samples of the digital sequence x(n) is a power of 2, N = 2, 4, 8,16, etc. If *x*(*n*) does not contain 2^m samples, then we simply append it with zeros until the number of the appended sequence is equal to an integer of a power of 2 data points.

Derivation of the FFT

The decomposition of the DFT is achieved by breaking a signal $x[n]$ down into shorter sequences i.e. even numbered points $x[2m]$, and the odd numbered points $x[2m+1]$.

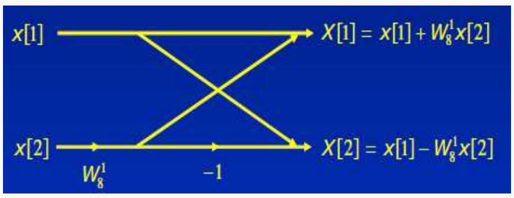
Keep on doing this until a *pair* of even-odd numbers is reached.

$$X[k] = \sum_{m=0}^{\frac{N}{2}-1} x[2m]\cdot W_N^{2mk} + \sum_{m=0}^{\frac{N}{2}-1} x[2m+1]\cdot W_N^{(2m+1)k}$$



Radix-2 Butterfly

- DFT operation is decomposed into DFTs of shorter lengths called **‘butterfly’**.
- Originally known as the **Radix-2 butterfly**:



The basic strategy that is used in the FFT algorithm is one of “divide and conquer”. Which involves decomposing an N-point DFT into successively smaller DFTs. To see how this works, suppose that the length of x(n) is even(i.e., N is divisible by 2). If x(n) is decimated into two sequences of length N/2, computing the N/2-point DFT of each these sequences requires approximately $(N/2)^2$ multiplications and the same number of additions .

Radix-2 Butterfly

. Thus, the two DFTs require $2(N/2)^2=1/2\ N^2$ multiplies and adds. Therefore, if it is possible to find the N-point DFT of x(n) from these two N/2-point DFTs in fewer than $N^2/2$ operations. A saving has been realized.

- A single butterfly requires **1** multiplication and **2** additions.
- N-point DFT can be computed using $(N/2)\log_2 N$ butterfly.

	No. of multiplication	No. of addition
DFT	N^2	$N(N-1)$
Radix-2 FFT	$(N/2)\log_2 N$	$N\log_2 N$

Radix-2 Butterfly

There are two format of FFT:-

1- Decimation in frequency

2- Decimation in Time

Decimation-in-Frequency FFT

This algorithm may be derived by decimating the output sequence X(k) into smaller and smaller subsequences. These algorithms are called decimation-in-frequency FFTs and may be derived as follows. Let N be a power of 2, $N = 2^m$. and consider separately evaluating the even-index and odd-index samples of X(k).

$$DFT\{x(n)\ with\ N\ \text{points}\} = \begin{cases} DFT\{a(n)\ with\ (N/2)\ \text{points}\} \\ DFT\{b(n)W_N^n\ with\ (N/2)\ \text{points}\} \end{cases}$$

Radix-2 Butterfly

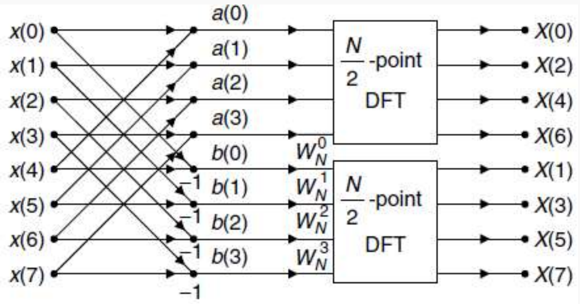


FIGURE 4.24 The first iteration of the eight-point FFT.

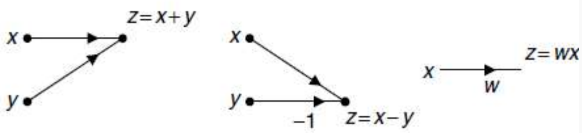


FIGURE 4.25 Definitions of the graphical operations.

$$W_N^{nk} = e^{\frac{-j2\pi nk}{N}}$$

$$W_N^0 = 1$$

$$W_N^1 = 0.707 - j0.707$$

$$W_N^2 = -j$$

$$W_N^3 = -0.707 - j0.707$$

Radix-2 Butterfly

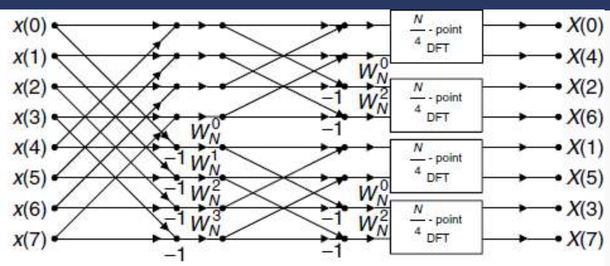


FIGURE 4.26 The second iteration of the eight-point FFT.

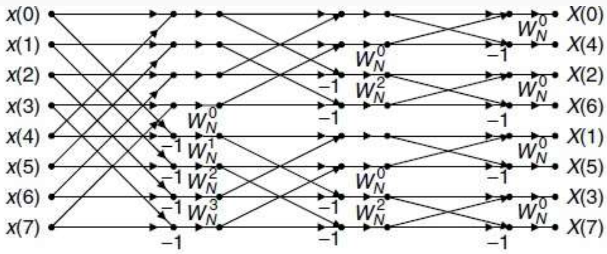


FIGURE 4.27 Block diagram for the eight-point FFT (total twelve multiplications).

Bit-reversal

Examination of the final chart shows that it is necessary to shuffle the order of the input data. This data shuffle is usually termed but-reversal for reasons that are clear if the indicates of the shuffled data are written in binary.

Binary	Bit Reverse	Decimal
000	000	0
001	100	4
010	010	2
011	110	6
100	001	1
101	101	5
110	011	3
111	111	7

Example

Given a sequence $x(n)$ for $0 \leq n \leq 3$, where $x(0) = 1$, $x(1) = 2$, $x(2) = 3$, and $x(3) = 4$,

a. Evaluate its DFT $X(k)$ using the decimation-in-frequency FFT method.

b. Determine the number of complex multiplications.

Solution:

a. Using the FFT block diagram in Figure 4.27, the result is shown in Figure 4.30.

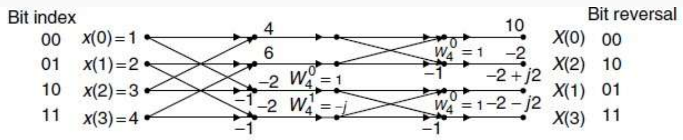


FIGURE 4.30 **Four-point FFT block diagram in Example 4.12.**

b. From Figure 4.30, the number of complex multiplications is four, which can also be determined by

$$\frac{N}{2}\log_2(N) = \frac{4}{2}\log_2(4) = 4.$$

Decimation-in-Time FFT

The decimation-in-time FFT algorithm is based on splitting (decimating) x(n) into smaller sequences and finding ***X (k)*** from the DFTs of these decimated sequences. This section describes how this decimation leads to an efficient algorithm when the sequence length is a power of 2.

Let x(n) be a sequence of length $N = 2^m$, and suppose that x(n) is split (decimated) into two subsequences, each of length N/2.

$$X(k) = \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2mk} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^k W_N^{2mk},$$

for $k = 0, 1, \ldots, N-1$.

The index for each input sequence element can be achieved by bit reversal of the frequency index in a sequential order.

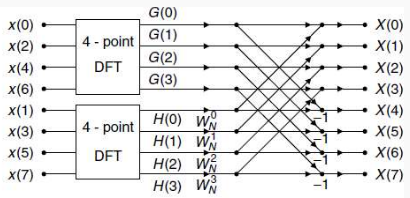


FIGURE 4.32 The first iteration.

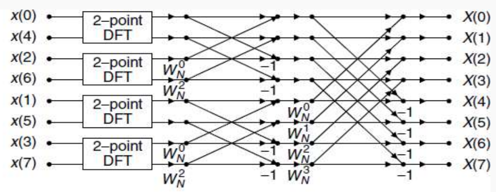


FIGURE 4.33 The second iteration.

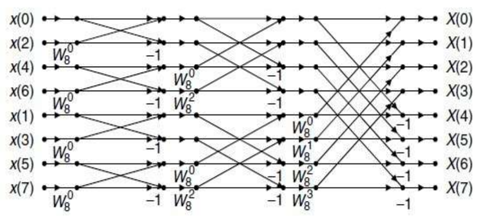


FIGURE 4.34 The eight-point FFT algorithm using decimation-in-time (twelve complex multiplications).

Example

Given a sequence $x(n)$ for $0 \le n \le 3$, where $x(0) = 1$, $x(1) = 2$, $x(2) = 3$, and $x(3) = 4$,

a. Evaluate its DFT $X(k)$ using the decimation-in-time FFT method.

Solution:

a. Using the block diagram in Figure 4.34 leads to

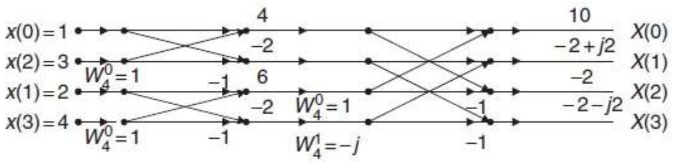


FIGURE 4.36 **The four-point FFT using decimation in time.**

The Inverse FFT (IFFT)

- It uses positive power of $W_{\tilde{N}} = W_N^{-1}$ instead of negative ones.
- There is an additional division of each output value by N.

Any FFT algorithm can be modified to perform the IDFT by

- Using positive powers instead of negatives
- Multiplying each component of the output by 1/N

- Hence the algorithm is the same but computational load increases due to N extra multiplications.

The Inverse FFT (IFFT)

after we change W_N to $W^{\sim N}$ in Figure 4.34 and multiply the output sequence by a factor of 1/N, we derive the inverse FFT block diagram for the eight-point inverse FFT in Figure

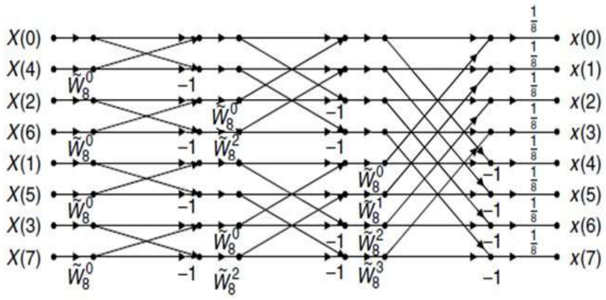


FIGURE 4.35 The eight-point IFFT using decimation-in-time.

Example

Given the DFT sequence $X(k)$ for $0 \le k \le 3$ computed in Example 4.12,

a. Evaluate its inverse DFT $x(n)$ using the decimation-in-frequency FFT method.

Solution:

a. Using the inverse FFT block diagram in Figure 4.28, we have

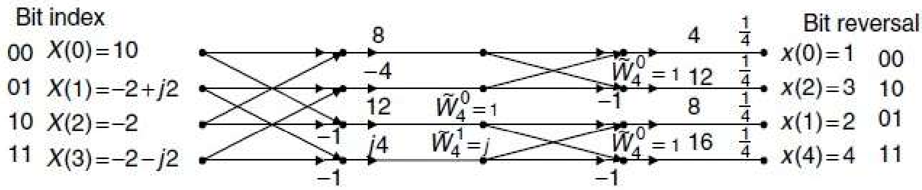


FIGURE 4.31 Four-point inverse FFT block diagram in Example 4.13.

Example

Given the DFT sequence $X(k)$ for $0 \le k \le 3$ computed in Example 4.14,

a. Evaluate its inverse DFT $x(n)$ using the decimation-in-time FFT method.

Solution:

a. Using the block diagram in Figure 4.35 yields

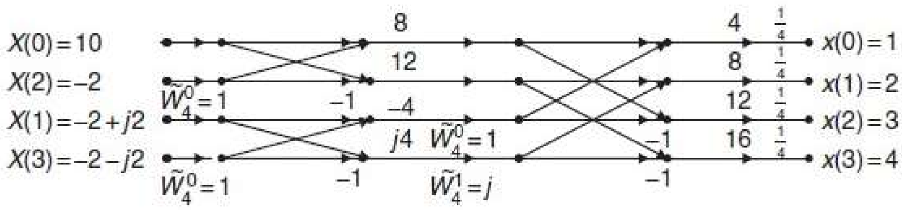


FIGURE 4.37 **The four-point IFFT using decimation in time.**

Example

Assume that a complex multiply takes 1 μs and that the amount of time to compute a DFT is determined by the amount of time it takes to perform all of the multiplications.

(*a*) How much time does it take to compute a 1024-point DFT directly?

(*b*) How much time is required if an FFT is used?

(*c*) Repeat parts (*a*) and (*b*) for a 4096-point DFT.

(*a*) Including possible multiplications by ± 1, computing an N-point DFT directly requires N^2 complex multiplications. If it takes 1 μs per complex multiply, the direct evaluation of a 1024-point DFT requires

$$t_{\text{DFT}} = (1024)^2 \cdot 10^{-6}\text{ s} \approx 1.05\text{ s}$$

(*b*) With a radix-2 FFT, the number of complex multiplications is approximately $(N/2)\log_2 N$ which, for $N = 1024$, is equal to 5120. Therefore, the amount of time to compute a 1024-point DFT using an FFT is

$$t_{\text{FFT}} = 5120 \cdot 10^{-6}\text{ms} = 5.12\text{ ms}$$

Example

(*c*) If the length of the DFT is increased by a factor of 4 to $N = 4096$, the number of multiplications necessary to compute the DFT directly increases by a factor of 16. Therefore, the time required to evaluate the DFT directly is

$$t_{\text{DFT}} = 16.78 \text{ s}$$

If, on the other hand, an FFT is used, the number of multiplications is

$$2{,}048 \cdot \log_2 4{,}096 = 24{,}576$$

and the amount of time to evaluate the DFT is

$$t_{\text{FFT}} = 24.576 \text{ ms}$$

Ninevah University
College of Electronics Engineering
Control and Systems Engineering
Third Stage / 1st semester 2025-2026
Digital Signal Processing (DSP)

Lecture6: Circular Convolution

Instructor:
Asst. Prof. Dr. Ahmed Jameel Abdulqader

Place and Date: Mosul / College of Electronics Engineering, /11/2025

Applications of circular convolution

1.Designing digital filters

2.Processing signals in communication systems

3.Audio signal processing

4.Processing images

5.Convolutional neural networks

6.Periodic data analysis

7.Error detection and correction in coding theory

8.Cryptographic algorithms

9.Radar signal processing

10.Biomedical signal processing

Circular convolution (Cyclic Convolution)

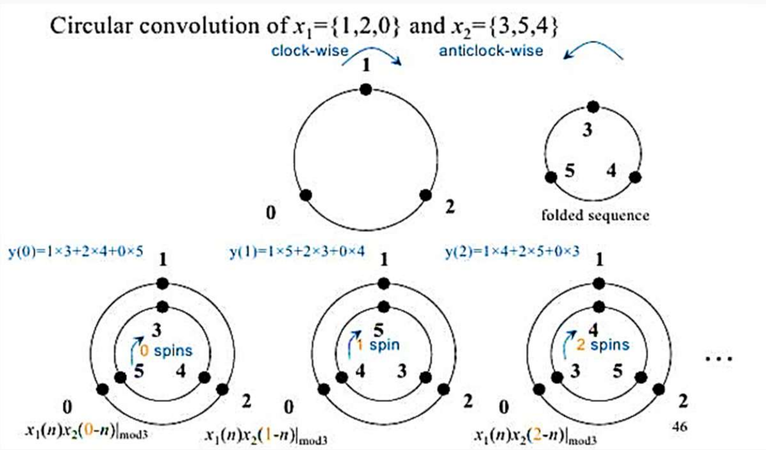
- It is a special case of periodic convolution, which is the convolution of two periodic functions that have the same period.
- Circular convolution is essentially the same process as linear convolution. Just like linear convolution, it involves the operation of folding a sequence, shifting it, multiplying it with another sequence, and summing the resulting products. However, in circular convolution, the signals are all periodic. Thus, the shifting can be thought of as actually being a rotation. Since the values keep repeating because of the periodicity. Hence, it is known as circular convolution.
- Circular convolution is also applicable for both continuous and discrete-time signals. We can represent Circular Convolution as

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

Circular convolution (Cyclic Convolution)

Here ***y(n)*** is a periodic output, ***x(n)*** is a periodic input, and ***h(n)*** is the periodic impulse response of the LTI system.
In circular convolution, both the sequences (input and impulse response) must be of equal sizes. They must have the same number of samples. Thus, the output of a circular convolution has the same number of samples as the two inputs.

Method-1



Circular convolution (Cyclic Convolution)

Method-2

Circular convolution of $x_1 = \{1, 2, 0\}$ and $x_2 = \{3, 5, 4\}$.

clock-wise → anticlock-wise ←

k	-2	-1	0	1	2	3	
$x_1(k)$			1	2	0		
$x_2(-k)\|_3$	4	5	3	4	5	3	$y(0) = 11$
$x_2(1-k)\|_3$	3	4	5	3	4	5	$y(1) = 11$
$x_2(2-k)\|_3$	5	3	4	5	3	4	$y(2) = 14$
$x_2(3-k)\|_3 = x_2(-k)\|_3$	4	5	3	4	5	3	$y(3) = 11$

Circular convolution (Cyclic Convolution)

- For the given example, circular convolution is possible only after modifying the signals via a method known as *zero padding*. In zero padding, zeroes are appended to the sequence that has a lesser size to make the sizes of the two sequences equal. Thus, for the given sequence, after zero-padding:

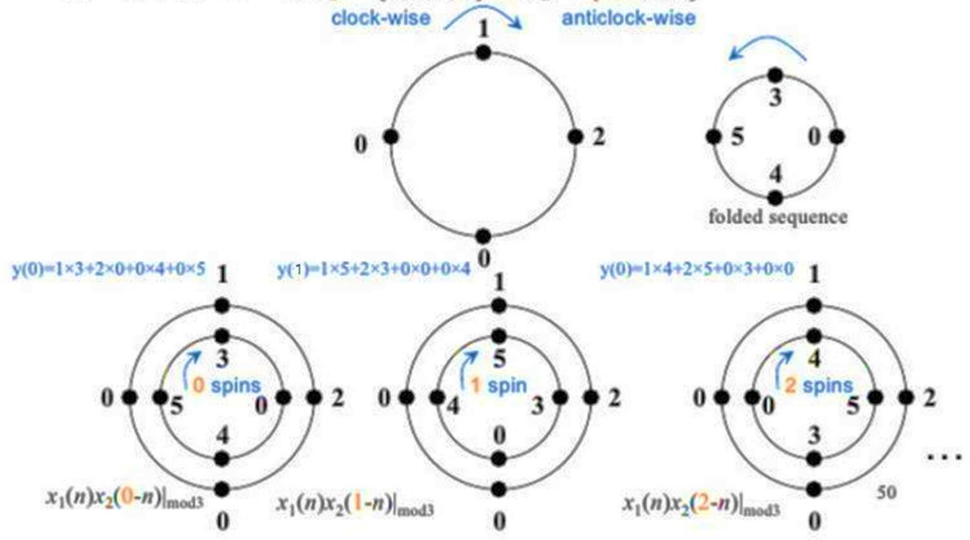
  x(n) = [1,2,3,0,0]

- Now both x(n) and h(n) have the same lengths. So circular convolution can take place. And the output of the circular convolution will have the same number of samples. i.e., 5.
- Graphically, when we perform circular convolution, there is a circular shift taking place. Alternatively, we can call it rotation.
- The output of a circular convolution is always periodic, and its period is specified by the periods of one of its inputs.

Circular convolution (Cyclic Convolution)

Example

Convolution of $x_1 = \{1, 2\}$ $(M = 2)$ and $x_2 = \{3, 5, 4\}$ $(L = 3)$,
$N = L + M - 1 = 4$, $x_1 = \{1, 2, 0, 0\}$ & $x_2 = \{3, 5, 4, 0\}$



Circular convolution (Cyclic Convolution)

Example

Convolution of $x_1 = \{1, 2\}$ $(M = 2)$ and $x_2 = \{3, 5, 4\}$ $(L = 3)$,
$N = L + M - 1 = 4$, $x_1 = \{1, 2, 0, 0\}$ & $x_2 = \{3, 5, 4, 0\}$

k	-2	-1	0	1	2	3	
$x_1(k)$			1	2	0	0	
$x_2(-k)\|_4$			3	0	4	5	$y(0) = 3$
$x_2(1-k)\|_4$			5	3	0	4	$y(1) = 11$
$x_2(2-k)\|_4$			4	5	3	0	$y(2) = 14$
$x_2(3-k)\|_4$			0	4	5	3	$y(3) = 8$

Circular convolution (Cyclic Convolution)

Method-3 **Matrix Multiplication Method**

Matrix method represents the two-given sequence $\boldsymbol{x_1(n)}$ and $\boldsymbol{x_2(n)}$ in matrix form.

- One of the given sequences is repeated via circular shift of one sample at a time to form a ***N X N*** matrix.
- The other sequence is represented as column matrix.
- The multiplication of two matrices given the result of circular convolution.

Circular convolution (Cyclic Convolution)

Matrix Method

$$\begin{bmatrix} x_2(0) & x_2(2) & x_2(1) \\ x_2(1) & x_2(0) & x_2(2) \\ x_2(2) & x_2(1) & x_2(0) \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \end{bmatrix} = \begin{bmatrix} x_3(0) \\ x_3(1) \\ x_3(2) \end{bmatrix}$$

Example

Input: x[n]=[1,2,4,2], h[n]={1,1,1}
Output: 7 5 7 8

Input: x[n]=[5,7,3,2],h[n]={1,5}
Output: 15 32 38 17

Circular convolution (Cyclic Convolution)

Example

Find the circular convolution of $x_1[n]$={2,1,2,1}
$x_2[n]$={1,2,3,4}

$$\begin{bmatrix}1 & 4 & 3 & 2\\ 2 & 1 & 4 & 3\\ 3 & 2 & 1 & 4\\ 4 & 3 & 2 & 1\end{bmatrix}\begin{bmatrix}2\\ 1\\ 2\\ 1\end{bmatrix} = \begin{bmatrix}2+4+6+2\\ 4+1+8+3\\ 6+2+2+4\\ 8+3+4+1\end{bmatrix} = \begin{bmatrix}14\\ 16\\ 14\\ 16\end{bmatrix}$$

Ninevah University
College of Electronics Engineering
Control and Systems Engineering
Third Stage / 1st semester 2025-2026
Digital Signal Processing (DSP)

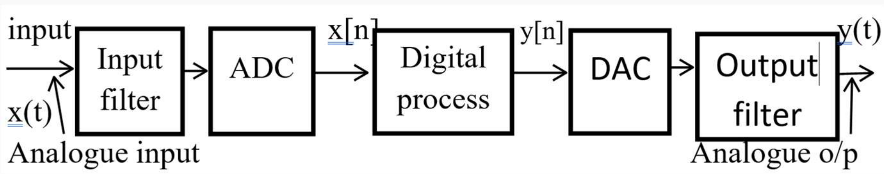
Lecture7: A framework for digital filter design

Instructor:
Asst. Prof. Dr. Ahmed Jameel Abdulqader

Place and Date: Mosul / College of Electronics Engineering, /11/2025

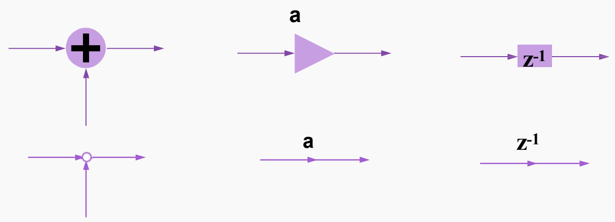
Introduction to digital filter

- A filter is essentially a system or network that selectively changes the wave shape, amplitude and or phase-frequency characteristics of a signal in a desired manner.
- Common filtering objectives are to improve the quality of a signal .... (to remove or reduce noise), to extract information from signals or to separate two or more signals previously combined to make, for example, efficient use of an available communication channel.
- A digital filter is a mathematical algorithm implemented in hardware and /or software that operate on a digital input signal to produce a digital output signal of achieving a filter objective.
- A simplified block diagram of a real time digital filter with analogue input and output signals is shown in figure.

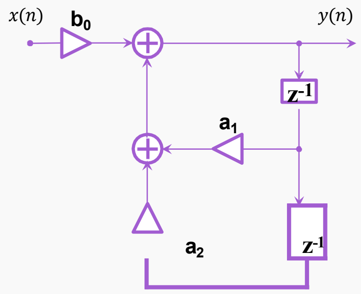


BASIC ELEMENTS OF DIGITAL FILTER STRUCTURES

- Adder has two inputs and one output.
- Multiplier (gain) has single-input, single-output.
- Delay element delays the signal passing through it by one sample. It is implemented by using a shift register.



BASIC ELEMENTS OF DIGITAL FILTER STRUCTURES



$$y(n) = b_0x(n) + a_1y(n-1) + a_2y(n-2)$$

Type of digital filter: FIR and IIR

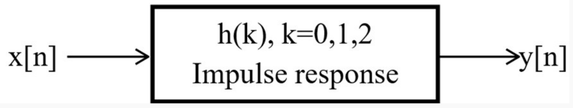
- Digital filters are broadly divided into two classes:-
  1) Infinite impulse response (IIR)
  2) Finite impulse response (FIR)
- Either type of filter can be represented by its impulse response sequence, h(k) (k=0, 1, 2, …) as shown in figure.
- The input and output signals to the filter are related by the convolution sum.
- For IIR filters, the impulse response is finite duration whereas for FIR it is of finite duration.

$$y[n] = \sum_{k=0}^{\infty} h(k)\, x\,(n-k) \quad \text{…. for IIR}$$

$$y[n] = \sum_{k=0}^{N-1} h(k)\, x\,(n-k) \quad \text{…. for FIR}$$

$$H(k) = \sum_{k=0}^{N-1} h(k)\, z^{-k}$$

$$H(z) = \frac{\sum_{k=0}^{N} a_k z^{-k}}{1+\sum_{k=0}^{M} b_k z^{-k}}$$



Representation of a filter by a suitable structure (Realization)

Realization involves converting a given transfer function H(z) into a suitable filter structure.

1- For FIR filter three structure are used:-
- (a) Direct form (or transversal)
- (b) Frequency sampling
- (c) Fast convolution.

2- For IIR filters, three structures are used:-
- (a) Direct form
- (b) Cascade form
- (c) Parallel form

FIR (FINITE IMPULSE RESPONSE) FILTER STRUCTURES

- The characteristics of the FIR filter
- FIR filters have Finite-duration Impulse Responses; thus, they can be realized by means of DFT
- The system function H(z) has the ROC of system $|z| > 0$, thus, it is a causal system
- An FIR filter is a non recursive system
- FIR filters can be designed to have a linear-phase response

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n}$$

It has N-1 order poles at $z = 0$ and N-1 zeros in $|z| > 0$

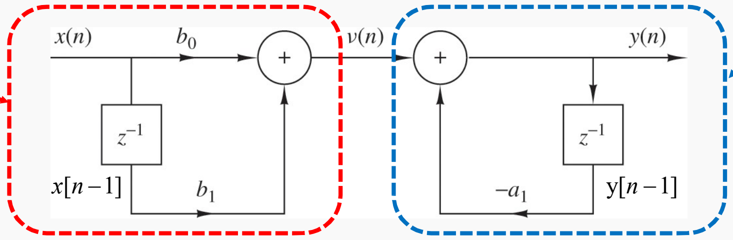
The order of such an FIR filter is N-1

Block Diagrams for First-Order System

$$y[n] = -a_1 y[n-1] + b_0 x[n] + b_1 x[n-1]$$

From this we see that we can use one delay to get $y[n\text{-}1]$ and a second delay to get $x[n\text{-}1]$.

Then we add the various terms together to create $y[n]$

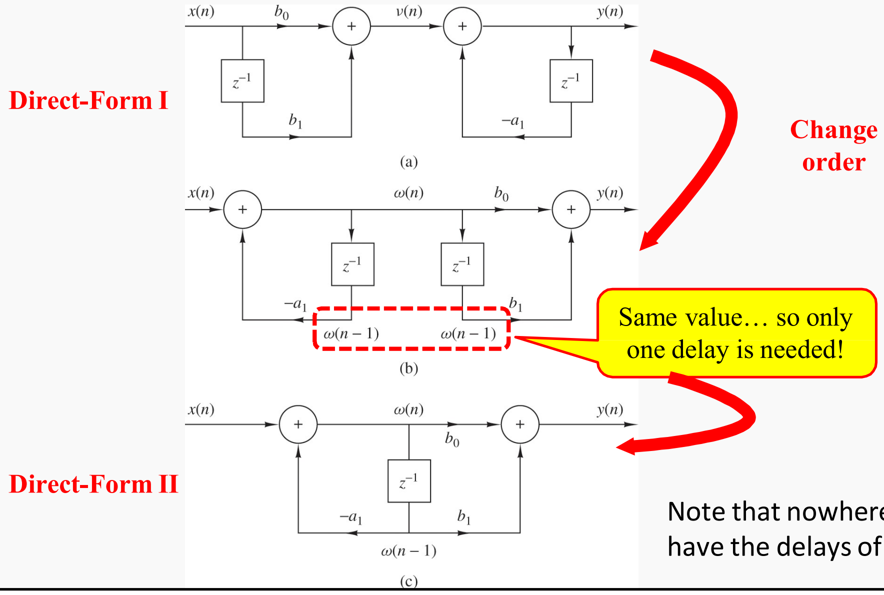


This form… that has separate delays for the input and for the output…. is called **Direct-Form I**

It is possible to reduce the number of delays with a "trick".

Trick to Get Direct-Form II – which has reduced number of delays

For LTI systems we can interchange their order without changing their overall mathematical result. So…



Note that nowhere in this do we have the delays of input or output!

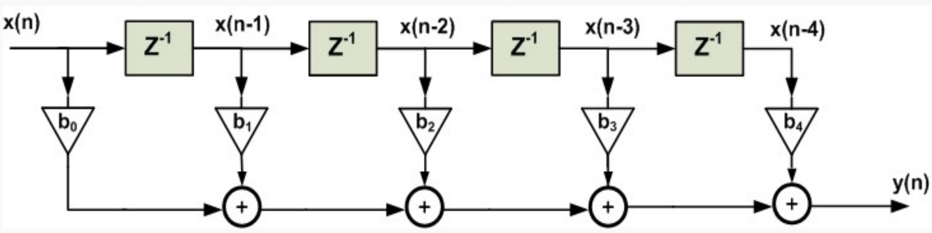
FIR FILTER STRUCTURES

- Direct form

In this form the difference equation is implemented directly as given:

$$y(n) = \sum_{m=0}^{N-1} h(m)x(n-m)$$

It requires N multiplications



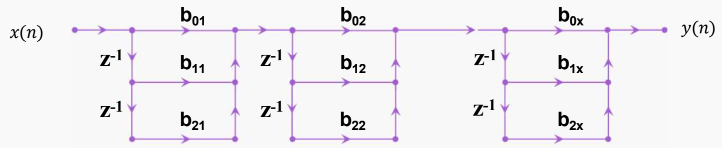
FIR FILTER STRUCTURES

- Cascade form

In this form the system function H(z) is converted into products of second- order sections with real coefficients

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n} = \prod_{k=1}^{\left[\frac{N}{2}\right]} (b_{0k} + b_{1k} z^{-1} + b_{2k} z^{-2})$$

It requires (3N/2) multiplications

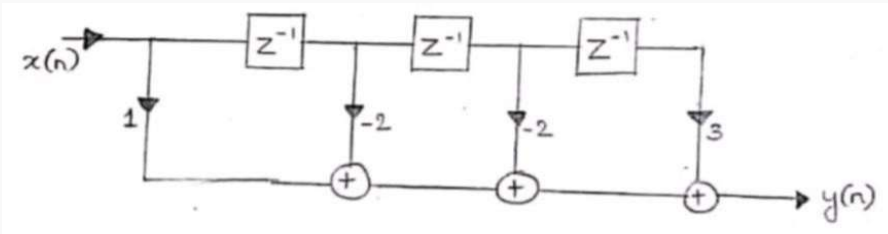


EXAMPLE

Draw the direct form structure for the FIR filter represented by the following difference equation

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

Sol:



EXAMPLE

Draw the direct form structure for the FIR filter represented by the following transfer function

$$H[z]= 4+2z^{-1}-2\ z^{-2}+3\ z^{-3}$$

Sol:

$$H(z)=\frac{Y(z)}{X(z)}=4+2z^{-1}-2z^{-2}+3z^{-3}$$

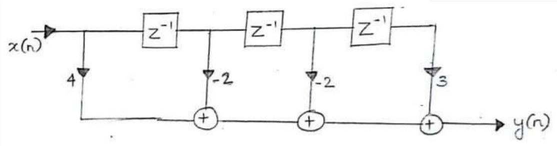
Time shifting

$$x(n-k) \overset{z}{\leftrightarrow} z^{-k}X(z)$$

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

Taking Inverse z-Transform

$$y(n)=4x(n)+2x(n-1)-2x(n-2)+3x(n-3)$$



EXAMPLE

Using cascade structure realize the FIR filter represented by the following transfer function

$$H[z]= (1+\frac{1}{2}z^{-1}+ z^{-2})\ (1+\frac{1}{4}z^{-1}+ z^{-2})$$

Sol:

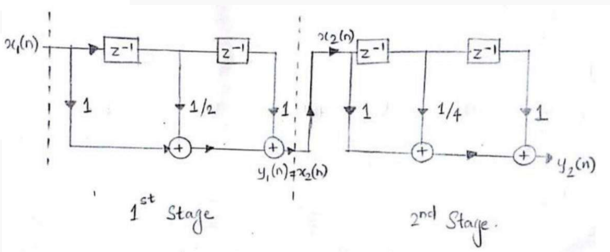
$$H(z)=H_1(z).H_2(z)$$

$$H_1(z)=(1+\frac{1}{2}z^{-1}+z^{-2}) \qquad H_2(z)=(1+\frac{1}{4}z^{-1}+z^{-2})$$

$$H_1(z)=\frac{Y_1(z)}{X_1(z)}=(1+\frac{1}{2}z^{-1}+z^{-2}) \qquad H_2(z)=\frac{Y_2(z)}{X_2(z)}=(1+\frac{1}{4}z^{-1}+z^{-2})$$

$$Y_1(z)=X_1(z)+\frac{1}{2}z^{-1}X_1(z)+z^{-2}X_1(z) \qquad Y_2(z)=X_2(z)+\frac{1}{4}z^{-1}X_2(z)+z^{-2}X_2(z)$$

$$y_1(n)=x_1(n)+\frac{1}{2}x_1(n-1)+x_1(n-2) \qquad y_2(n)=x_2(n)+\frac{1}{4}x_2(n-1)+x_2(n-2)$$



Ninevah University
College of Electronics Engineering
Control and Systems Engineering
Third Stage / 1st semester 2025-2026
Digital Signal Processing (DSP)

Lecture 8: A framework for digital filter design-IIR

Instructor:
Asst. Prof. Dr. Ahmed Jameel Abdulqader

Place and Date: Mosul / College of Electronics Engineering, /12/2025

IIR(INFINITE IMPULSE RESPONSE) FILTER STRUCTURES

- The characteristics of the IIR filter
- IIR filters have Infinite-duration Impulse Responses
- The system function H(z) has poles in $0 < |z| < \infty$
- An IIR filter is a recursive system

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 - (a_1 z^{-1} + \cdots + a_N z^{-N})}$$

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

The order of such an IIR filter is called N if $a_N \neq 0$

IIR FILTER STRUCTURES

- Direct form

In this form the difference equation is implemented directly as given. There are two parts to this filter, namely the moving average part and the recursive part (or the numerator and denominator parts). Therefore, this implementation leads to two versions: direct form I and direct form II structures

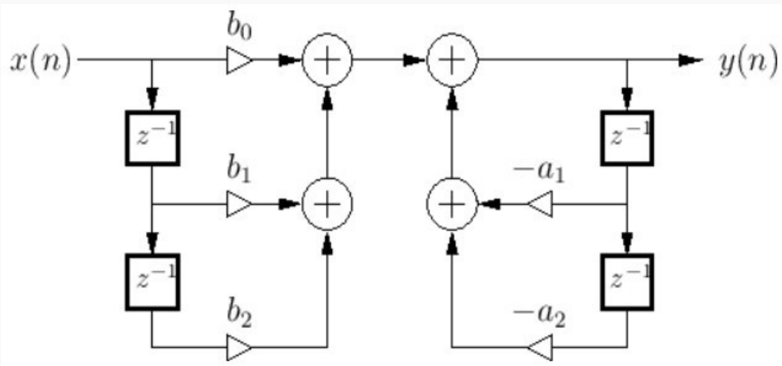
$$y(n) = \sum_{k=0}^{M} b_k x(n-k) + \sum_{k=1}^{N} a_k y(n-k)$$

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}}$$

IIR FILTER STRUCTURES

- Direct form I

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

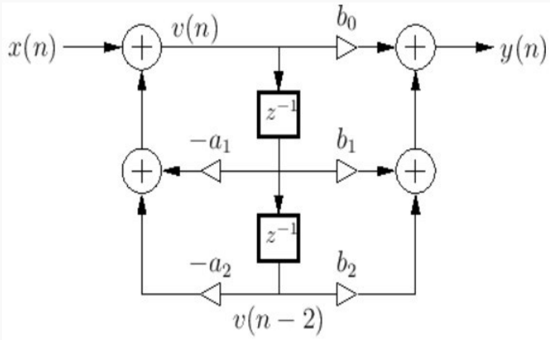


IIR FILTER STRUCTURES

- Direct form II

For an LTI cascade system, we can change the order of the systems without changing the overall system response

$$v(n) = x(n) - a_1 v(n-1) - a_2 v(n-2)$$
$$y(n) = b_0 v(n) + b_1 v(n-1) + b_2 v(n-2)$$

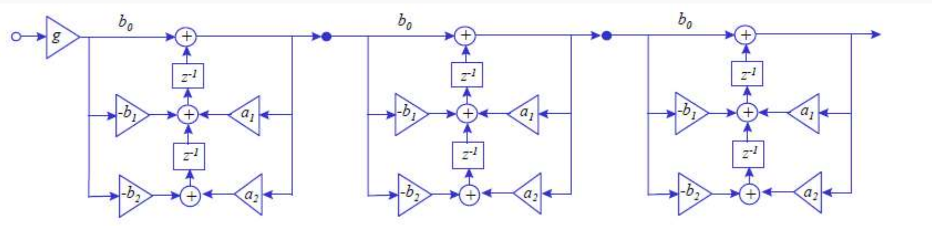


IIR FILTER STRUCTURES

- Cascade form

For an IIR filter each second order cascade section has the form

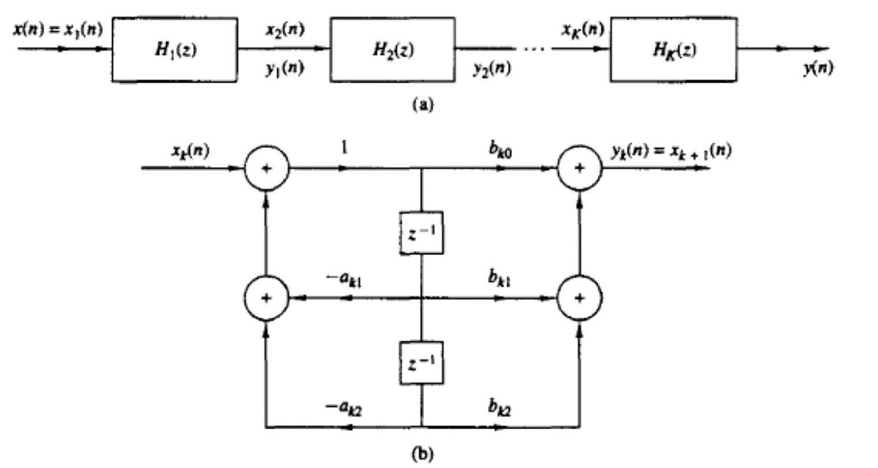
$$H_k(z) = \frac{1 + b_k z^{-1} + b_k z^{-2}}{1 + a_{k1} z^{-1} + a_{k2} z^{-2}}$$



Realization of an IIR cascade structure from second order sections.

IIR FILTER STRUCTURES

- Cascade form

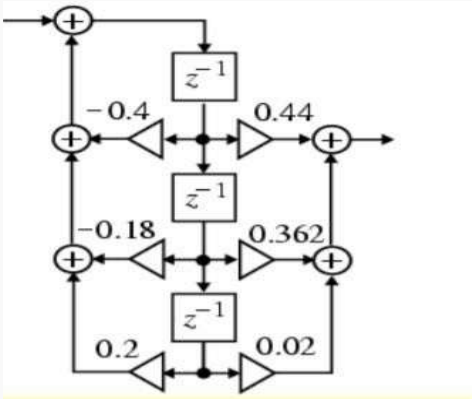


EXAMPLE

Draw the direct form II realization of the following transfer function

$$H(z)=\frac{0.44z^{-1}+0.362z^{-2}+0.02z^{-3}}{1+0.4z^{-1}+0.18z^{-2}-0.2z^{-3}}$$

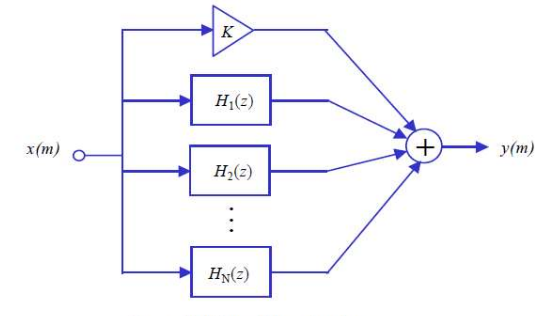
Sol:



IIR FILTER STRUCTURES

- Parallel form
- In this structure, the input signal is processed separately by a different subsystems.
- An IIR Transfer function can be realized in a parallel form by making use of the partial fraction of expansion of the Transfer function.

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1+\sum_{k=1}^{N} a_k z^{-k}} = \frac{\prod_{k=1}^{M}(1-\beta_k z^{-1})}{\prod_{k=1}^{N}(1-\alpha_k z^{-1})}$$



A parallel-form filter structure.

EXAMPLE

Obtain Cascade and parallel structures for the following system:

$$y(n) = \frac{3}{4}y(n-1) - \frac{1}{8}y(n-2) + x(n) + \frac{1}{3}x(n-1)$$

Sol:

Taking Z-Transform on both sides results in

$$Y(z) - \frac{3}{4}z^{-1}Y(z) + \frac{1}{8}z^{-2}Y(z) = X(z) + \frac{1}{3}z^{-1}X(z)$$

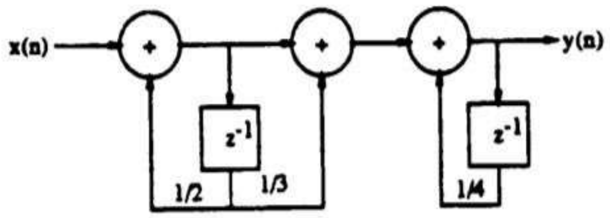
The corresponding Transfer Function is

$$H(z) = \frac{1+\frac{1}{3}z^{-1}}{1-\frac{3}{4}z^{-1}+\frac{1}{8}z^{-2}}$$

EXAMPLE

Cascade Form:

$$H(z) = \frac{1+\frac{1}{3}z^{-1}}{1-\frac{3}{4}z^{-1}+\frac{1}{8}z^{-2}} = \frac{1+\frac{1}{3}z^{-1}}{(1-\frac{1}{2}z^{-1})}\cdot\frac{1}{(1-\frac{1}{4}z^{-1})}$$
$$= H_1(z).H_2(z)$$



EXAMPLE

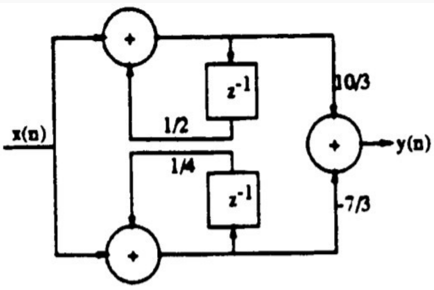
Parallel Form:

By using partial fraction expansion

$$H(z) = \frac{1+\frac{1}{3}z^{-1}}{1-\frac{3}{4}z^{-1}+\frac{1}{8}z^{-2}} = \frac{A}{(1-\frac{1}{2}z^{-1})}+\frac{B}{(1-\frac{1}{4}z^{-1})}$$

Solving for A and B,

$$H(z) = \frac{10}{3}\frac{1}{(1-\frac{1}{2}z^{-1})} - \frac{7}{3}\frac{1}{(1-\frac{1}{4}z^{-1})}$$



Discrete-Time System Relationships