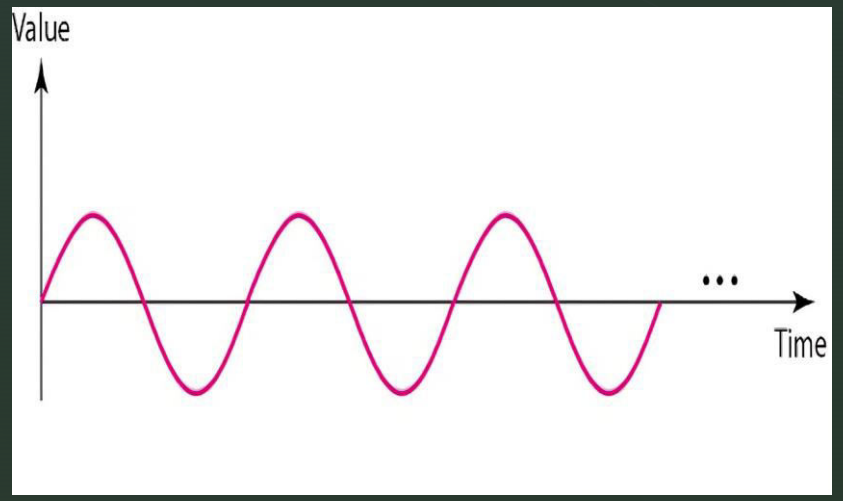
WHAT IS A SIGNAL?

- Signal is a physical quantity that varies with time and carries information.
- For example:
  - VOLTAGE & CURRENT (electrical signals)
  - ECG machine(real-time application; electrical impulse produced by heart)
  - Audio, Video, SONAR, RADAR, Speech signals etc

TWO TYPES OF SIGNALS

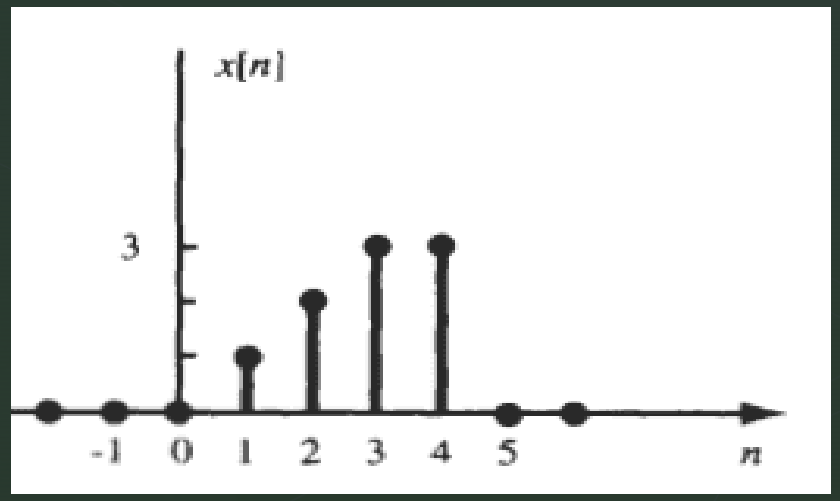
ANALOG SIGNALS

- Signal defined *continuously* for any value of an independent variable.
- When varied with time; Continuous-time signals.



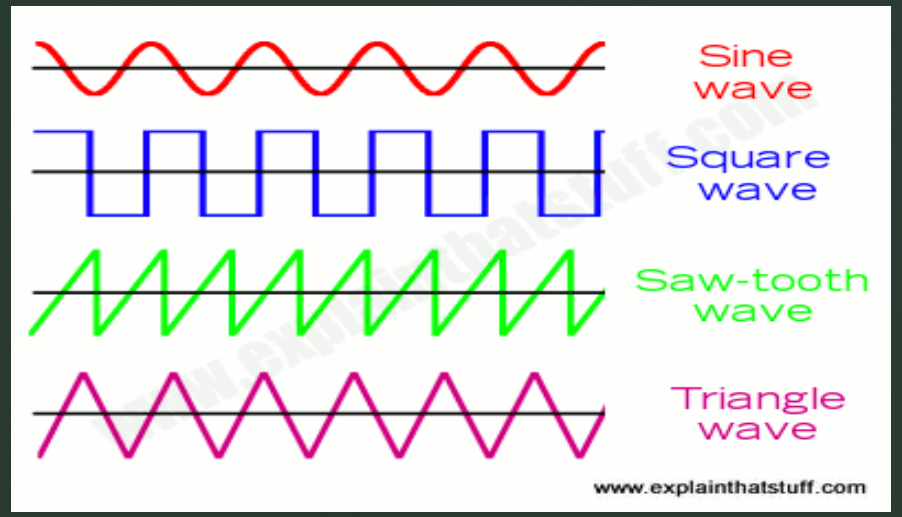
DIGITAL SIGNALS

- Signal defined for *discrete* intervals of an independent variable.
- When varied with time; Discrete-time signals.

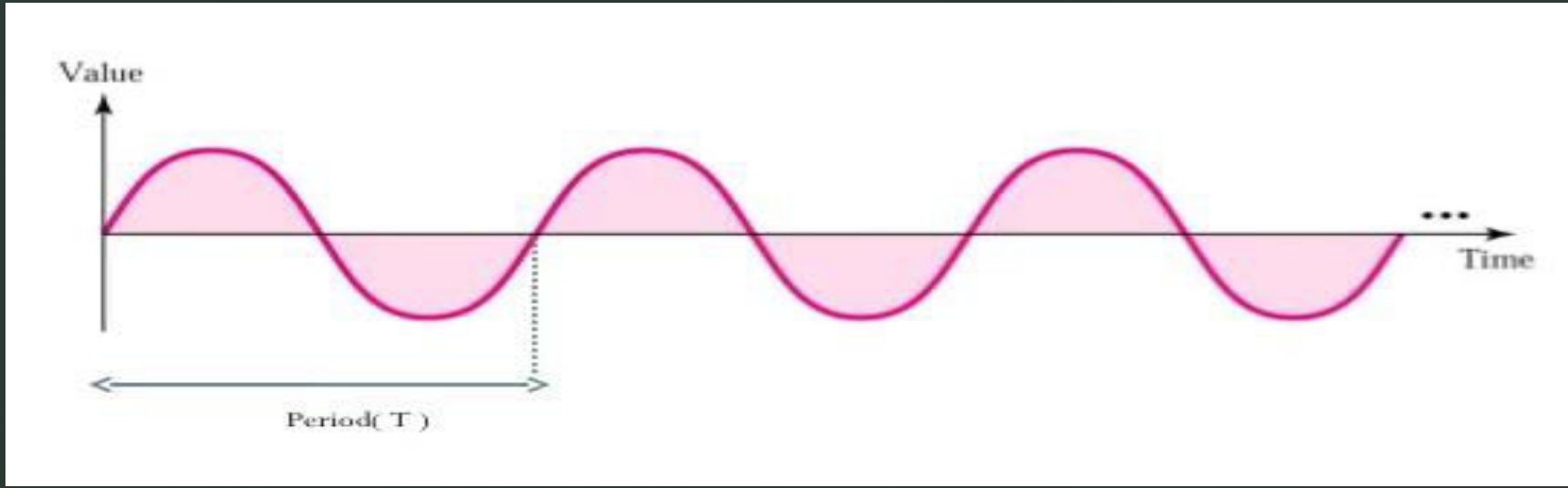


PERIODIC SIGNALS

- Signal *that repeats itself after a specific(fixed) interval of time* is called periodic signals.
- The sine and cosine functions are commonly used to represent periodic signals.
- The general form of a periodic signal can be expressed as:
  - $x(t)=A\cdot\cos(2\pi ft+\phi)$
- Signal that repeats its pattern over a period is called periodic signal (having a *definite pattern*).
- They are *Deterministic signals*.
- For example:
  - Sinusoidal
  - Square
  - Sawtooth
  - Triangle



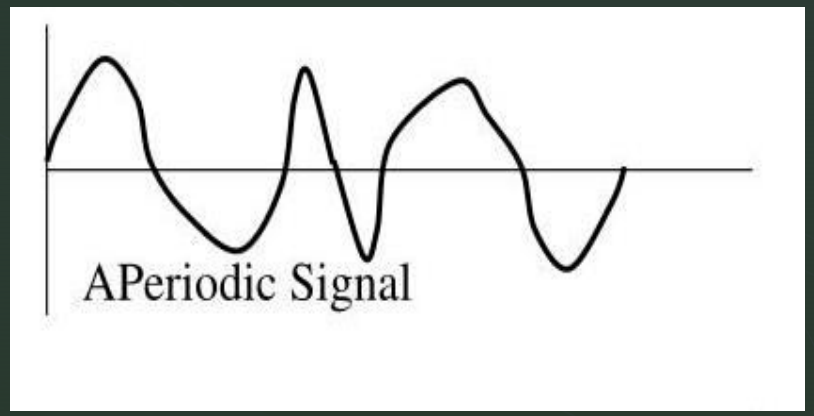
PERIODIC SIGNALS



- Fixed interval is called *Fundamental Time Period(T)* of the signal.
- It satisfies the condition; $x(t)=x(t+T)$

NON-PERIODIC SIGNALS

- Also known as *APERIODIC SIGNALS*.
- Signal *does not repeats itself after a specific(fixed) interval of time* is called aperiodic signal.
- Aperiodic signal *does not move with a definite pattern*.
- Instead of using sinusoidal functions, aperiodic signals can be represented using mathematical functions like impulses or exponential functions.
- For example:
  - Sound from Radio
  - Noise of all types
- They are *Random signals*.



Continuous-Time Signals wrt Periodicity

- A continuous-time signal is **Periodic** if it repeats its pattern over a continuous range of time.
  - Mathematically, $x(t)= x(t+T)$
    where T is the period of the signal
  - Example is sine wave
- A continuous-time signal is **Aperiodic** if it does not exhibit a repeating pattern over time.
  - No simple mathematical formula with a repeating pattern.
  - Example is an impulse function

Discrete-Time Signals wrt Periodicity

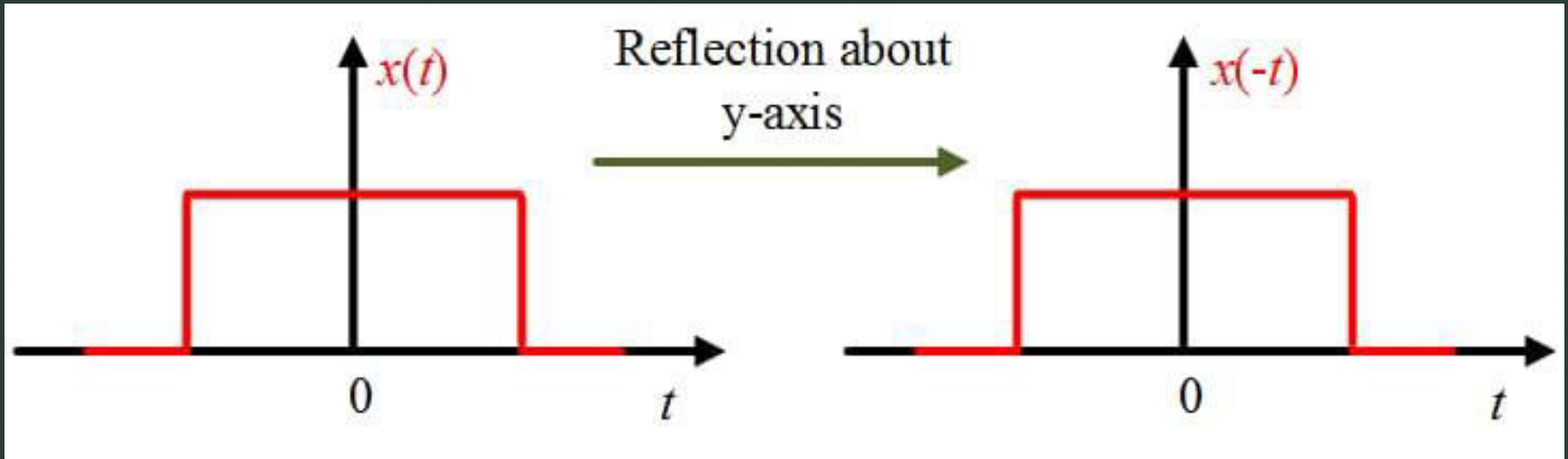
- A discrete-time signal is **Periodic** if it repeats its values after a certain number of samples.
  - Mathematically, $x[n]=x[n+N]$,
    where N is the period of the signal.
  - Example is a Discrete sine wave.
- A discrete-time signal is **Aperiodic** if it does not have a repeating pattern in its values over discrete time.
  - No simple mathematical formula with a repeating pattern in discrete time.
  - Example is a Discrete impulse sequence.

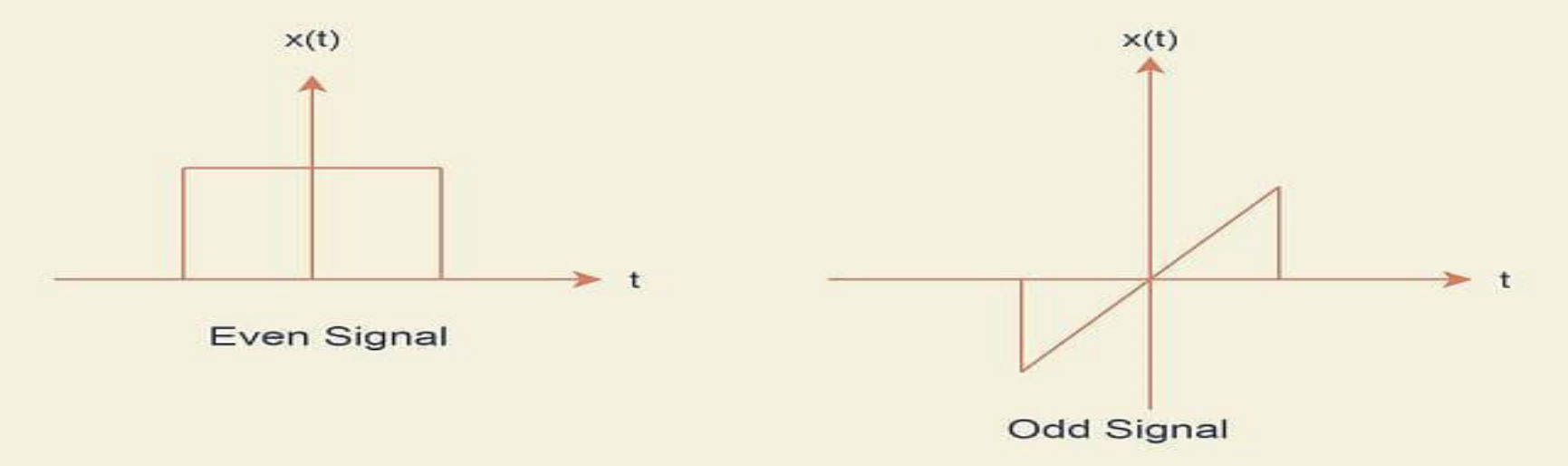
Applications of Periodic and Aperiodic waves in DSP

- **Signal Processing:**
  - **Periodic Signals:** Used in audio signal analysis, decomposed with Fourier analysis.
  - **Aperiodic Signals:** Common in real-world scenarios, analyzed for information extraction.
- **Communications:**
  - **Periodic Signals:** Utilized in modulation (AM, FM) for efficient transmission.
  - **Aperiodic Signals:** Crucial in digital communication for transmitting discrete data.

EVEN SIGNAL

- It's a type of signal that exhibits *symmetry about y-axis*.
- This means that if you have to fold the signal in half along the y-axis the *two halves would be identical* or mirror images of each other.
- *Symmetric Signal*
- In other words the signals *remain unchanged* when you reflect it across y-axis.
- **Mathematical expression x(-t) = x(t)**


x(t)
Reflection about
y-axis
x(-t)
0
t
0
t


x(t)
t
Even Signal
x(t)
t
Odd Signal

Even signal in frequency and time domain

- **Time Domain:**
  - In the time domain, an even signal exhibits symmetry around the vertical axis (usually the y-axis or t = 0). This symmetry implies that the signal's values at positive time values are equal to its values at corresponding negative time values.
  - For example, a cosine function demonstrating symmetry around the y-axis.
  - Cos (t) = cos(-t)
  - If a signal is even in the time domain, its Fourier transform will be entirely real and even.
- **Frequency Domain:** an even signal's frequency spectrum is symmetrical about the frequency axis. The positive and negative frequency components are equal in magnitude.

Applications of even signal

- Signal Processing
- Communication Systems
- Image Processing : symmetry and reflection
- Power Systems Analysis
- Audio processing: symmetric waveforms in sound engineering
- Filtering and noise reduction

ADVANTAGES OF EVEN SIGNALS

- **Simplified Mathematics :** Integration, differentiation, and other mathematical operations involving even signals often lead to simpler expressions due to the specific properties of even functions. Simplifies Analysis; by only focusing on half of the signal
- **Noise Reduction:** Exploiting the symmetric nature of even signals helps in canceling out noise components, aiding in noise reduction and improving signal-to-noise ratios.
- **Stability in Control Systems**: In control theory, even signals facilitate stability analysis and control system design due to their predictable properties and simplified mathematical representation.
- **Symmetry:** Even signals possess symmetry around the vertical axis, making calculations and analysis simpler in many mathematical operations, reducing complexity in signal processing and computations.

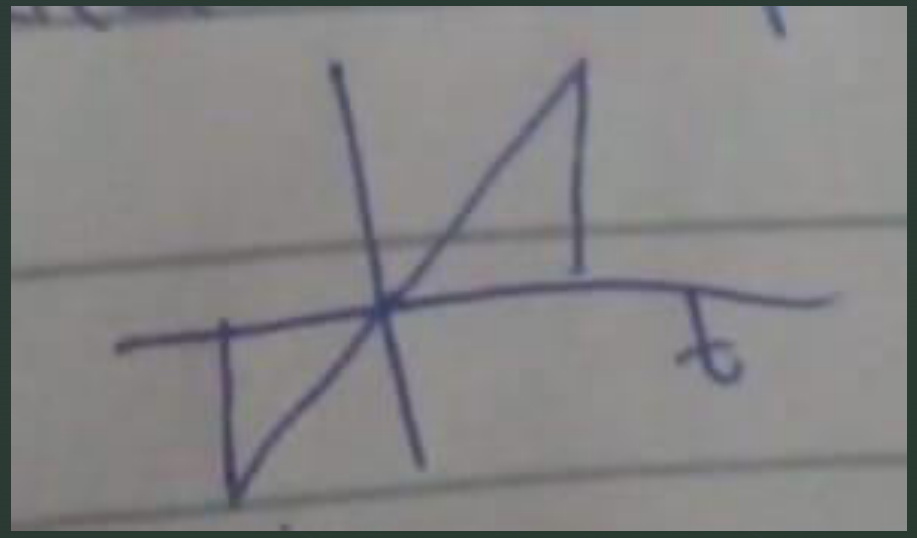
Limitations of Even signal

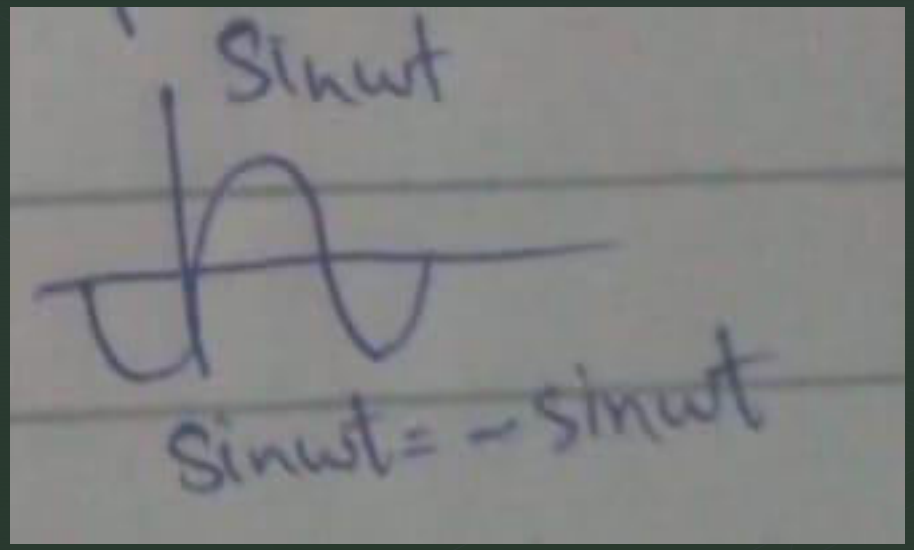
- **Limited Applicability:** Even signals represent a specific subset of signals with symmetry about the vertical axis. This limitation excludes many types of signals that do not exhibit this symmetry.
- **Information Loss:** Due to their symmetric nature, even signals might discard or overlook important information present in asymmetric or non-even signals. This can lead to potential loss of details in certain applications.
- **Complexities in Real-world Scenarios:** Real-world signals often contain both even and odd components. Over-reliance on even signals alone may oversimplify the analysis, potentially overlooking critical aspects captured by the odd components.

ODD SIGNAL

- A signal that is *symmetry with respect to it's origin* is known as odd signal or odd function.
- Therefore, the odd signals are also called *antisymmetric signals*.
- *Sine wave* is an example of odd signal.
- Mathematically, a signal x[n] is considered odd if it satisfies the condition:
  x[n]=−x[−n]

ODD SIGNAL





Filter Basics

- A filter is a system or device that is used to modify or manipulate the characteristics of a signal
- " Basically a set of mathematical equations that is applied on a signal to change it in different ways "
- Filters are commonly employed to alter specific features of a signal

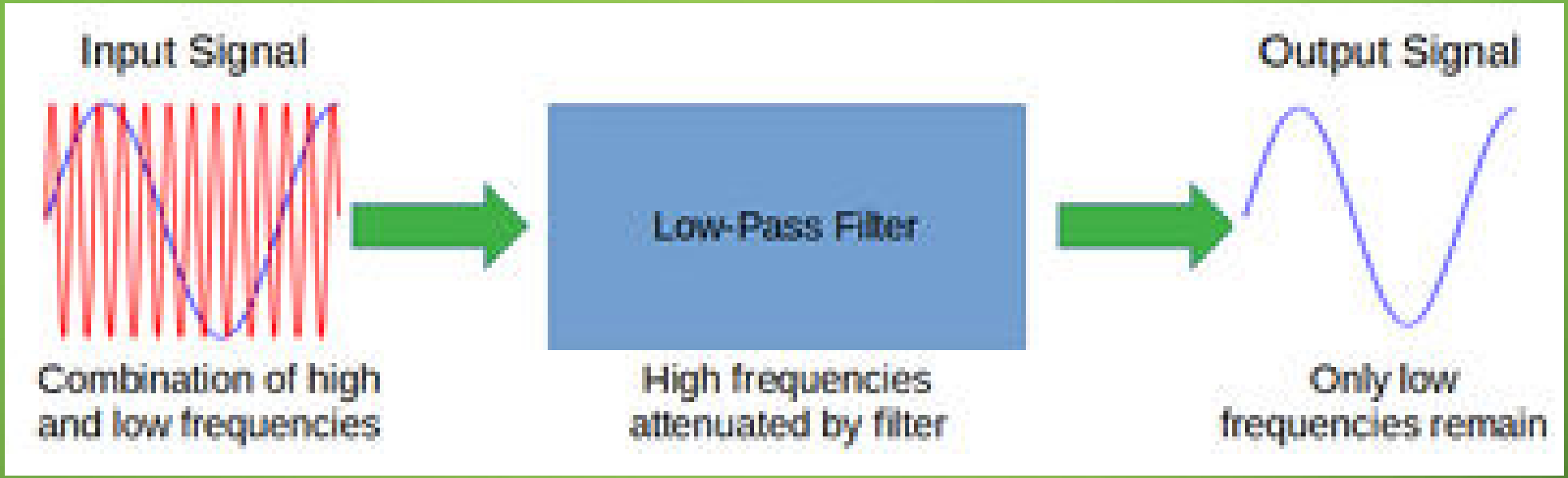
A Signal in this Context Could be Anything from an Electrical signal to Audio or Image

Types of Filters

- Low Pass-Filter
- High Pass-Filter
- Band Pass-Filter
- Band Stop-Filter
- IIR Filter
- FIR Filter

TYPES OF FILTERS

- Low Pass Filter:
- A Low Pass Filter (LPF) is a type of filter that allows low-frequency components of a signal to pass through while attenuating (reducing) the amplitude of higher-frequency components.

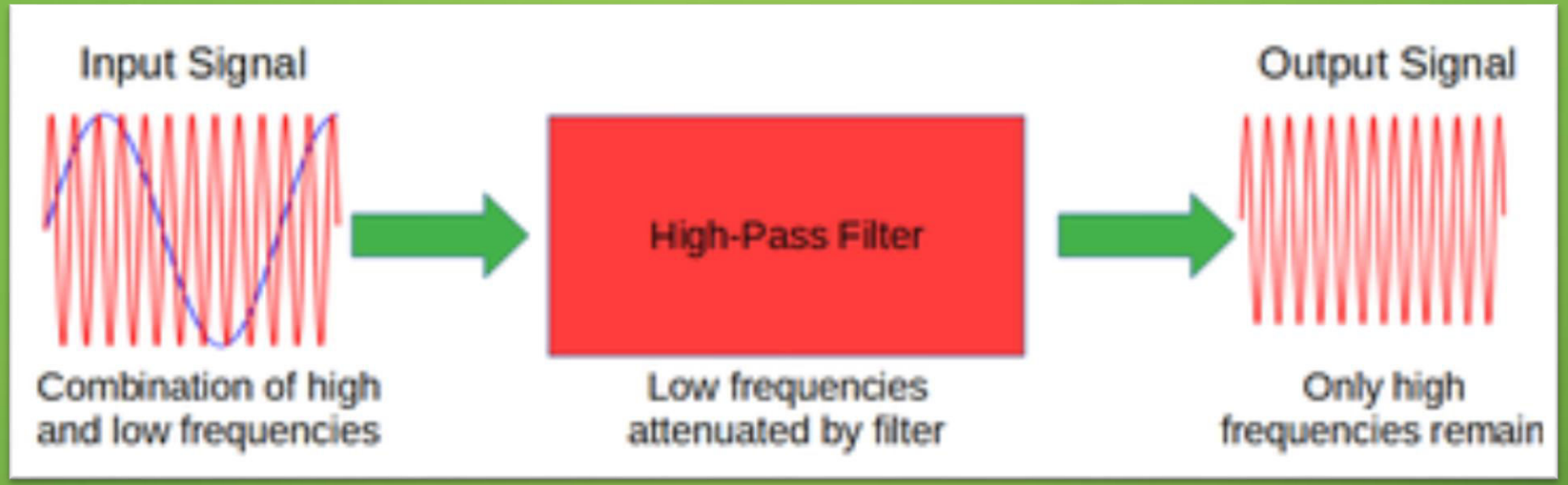


- Particularly useful when u want to keep low frequency sounds and get rid high frequency sounds
  like for example in a concert you`d like to keep the bass and tones and remove other High Pitched Noises

```
% Load audio file
[y, fs] = audioread('path/to/your/audiofile.wav');

% Design a simple low-pass filter (Butterworth filter)
cutoffFrequency = 500; % Set the cutoff frequency in Hz
order = 4; % Filter order (adjust as needed)

% Design the filter
[b, a] = butter(order, cutoffFrequency / (fs/2), 'low');

% Apply the low-pass filter
filteredAudio = filter(b, a, y);
```

HIGH PASS FILTER

- High Pass Filter:
- A High Pass Filter (HPF) is a type of filter that allows higher-frequency components of a signal to pass through while attenuating or blocking lower-frequency components.



- Could be used to Remove Background Sounds for example *Humm* sounds of the Air Conditioner or the Fan

```
% Load audio file with low-frequency hum
[y, fs] = audioread('podcast_recording.wav');

% Design a high-pass Butterworth filter
cutoffFrequency = 100; % Set the cutoff frequency above the hum frequency
order = 4; % Set the filter order

% Design the high-pass filter
[b, a] = butter(order, cutoffFrequency / (fs/2), 'high');

% Apply the high-pass filter to remove low-frequency hum
filteredAudio = filter(b, a, y);

% Save or play the filtered audio
```

BANDPASS FILTER

- BandPass Filter:
- Allows a specific "band" or Range of frequencies to pass through while attenuating frequencies outside that range

- A **bandpass filter** can be obtained by cascading both high and low pass filters. It has a huge application in audio amplifier circuits and wireless transceivers where the speaker has to play only the desired set of frequencies and ignore the rest.
  A Bandpass Filter is employed in the radio receiver to select the desired radio station's frequency band and reject others.

```
% Load an audio file
[y, fs] = audioread('your_audio_file.wav');

% Design a basic bandpass filter using the butter method
centerFrequency = 1000;  % Center frequency of the pass band
bandwidth = 200;  % Bandwidth of the pass band

% Design the bandpass filter
[b, a] = butter(2, [centerFrequency - bandwidth/2, centerFrequency + bandwidth/2]/(fs/2), 'bandpass');

% Apply the bandpass filter to the audio signal
filteredAudio = filter(b, a, y);
```

- Center Frequency : Center frequency is the specific frequency around which a filter is designed to operate.
- Bandwidth: is the range of frequencies that a filter allows to pass through or rejects.

BANDSTOP FILTER

- Bandstop Filter:
- Allow Everything Outside the Specific Range of Frequencies to pass through and stop the frequencies within that range

```
% Design a bandstop filter
centerFrequency = 1000;  % Notch frequency
bandwidth = 100;  % Width of the stop band

[b, a] = butter(2, [centerFrequency - bandwidth/2, centerFrequency + bandwidth/2]/(fs/2), 'stop');

% Apply the bandstop filter to the signal
filteredSignal = filter(b, a, inputSignal);
```

- Uses the Same Parameters as Bandpass but instead of 'pass' its 'stop'

- **Other Types of Filters**

1- Adaptive Filter

2- IIR Filter

3- FIR Filter

4- Comb Filter

5- Bell Filter

6- Low And High Shelf Filter

General Principles in Filter Design

- Frequency Response
- Filter Order
- Filter Type Selection

Filter Implementation

- **Analog Filters** If you're working with analog signals, you might implement your filter using analog components like resistors, capacitors, and inductors

- **Digital Filters-** For digital signals (common in modern electronics), filters are implemented using algorithms. Digital Signal Processors (DSP) or software (like MATLAB, Python with libraries such as SciPy) can be used.

- **Filter Realization**

- Convert the mathematical representation into a practical circuit or code. This involves choosing specific components (for analog filters) or implementing algorithms (for digital filters).

Understanding IIR Filters

- ● IIR (Infinite Impulse Response) filters are a class of digital filters used in signal processing.
- ● They are characterized by their recursive structure, which means that the output of the filter depends on both the current and past inputs.
- ● IIR filters are well-suited for applications where sharp transition bands and high attenuation levels are required

How IIR Filters Work

- ● IIR filters use a feedback loop where the output of the filter affects the input.
- ● This feedback loop gives IIR filters their recursive structure.
- ● The recursive structure of IIR filters allows them to achieve sharp transition bands and high attenuation levels.
- ● IIR filters have an infinite impulse response, which means that they will respond indefinitely to an impulse input.

Advantages of IIR Filters

- ● IIR filters can achieve high performance with low computational complexity due to their recursive structure

. ● IIR filters are efficient for applications dealing with continuous signals.

● IIR filters can be designed to have a smooth frequency response, which is important for preserving the fidelity of the signal.

Drawbacks of IIR Filters

- IIR filters can be unstable, especially if not carefully designed.
- IIR filters can introduce phase distortion into the signal.
- Phase distortion can be particularly noticeable at lower frequencies, impacting the fidelity of the filtered signal.

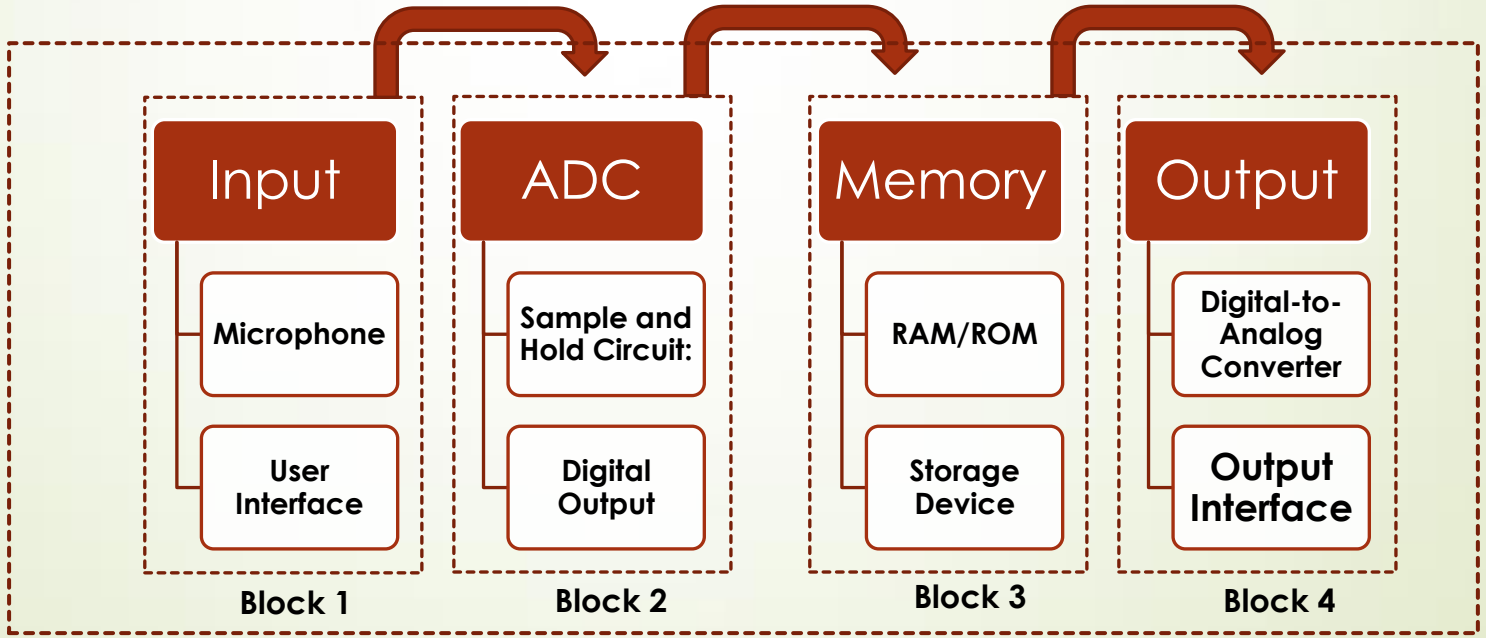
System

- Physically, a system is an interconnection of components, devices, etc., such as a computer or an aircraft or a power plant.
- Conceptually, a system is an abstract object that accepts input signals and produces output signals in response.
- A system is an operator which maps input into output.

An abstract representation of a system.

System Blocks

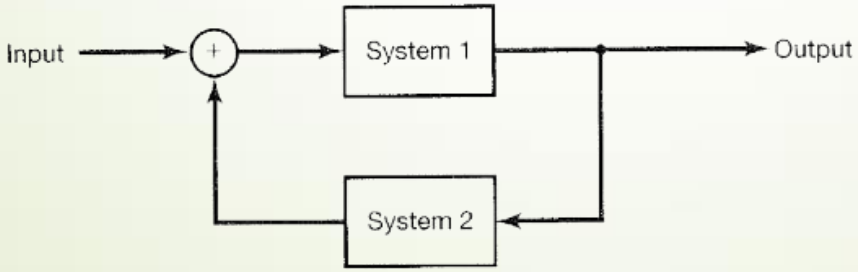
- Systems can be connected/combined to form a bigger/overall system, e.g., break down a big task into several sub-tasks and each system handles one sub-task.
  - **Input Block**
  - **ADC (Analog-to-Digital Converter)**
  - **Memory Block**
  - **Output Block**
  - **Control Block**



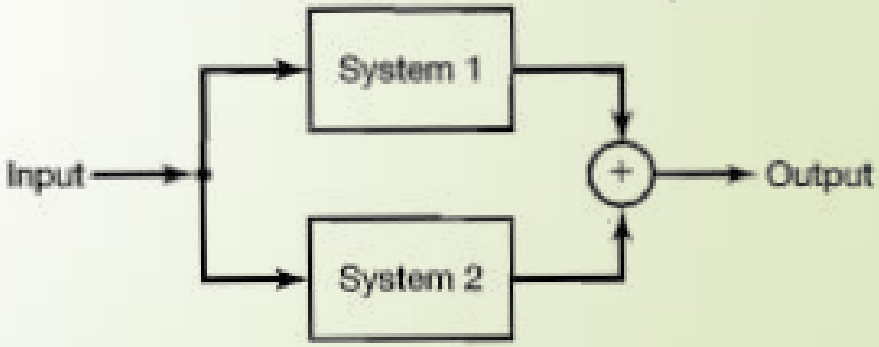
System Interconnections

- Interconnection refers to the way multiple systems are connected or linked together to form a larger, more complex system.
- In a control system, multiple subsystems (like controllers, sensors, and actuators) may be interconnected to regulate the behaviour of a more extensive system.

System Blocks In Series



System Blocks In Feedback Interconnection



System Blocks In Parallel

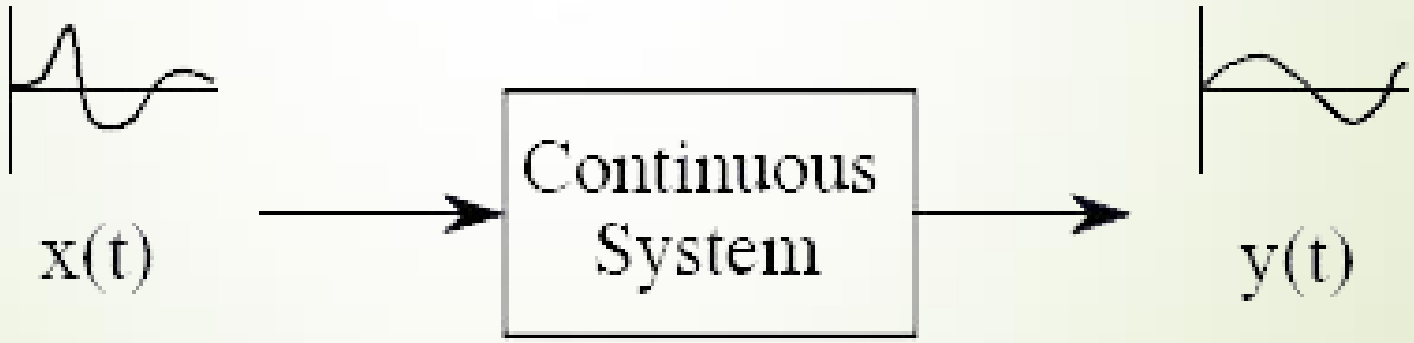
Types of Systems

There are two types of system:

- Continuous Time System
- Discrete Time System

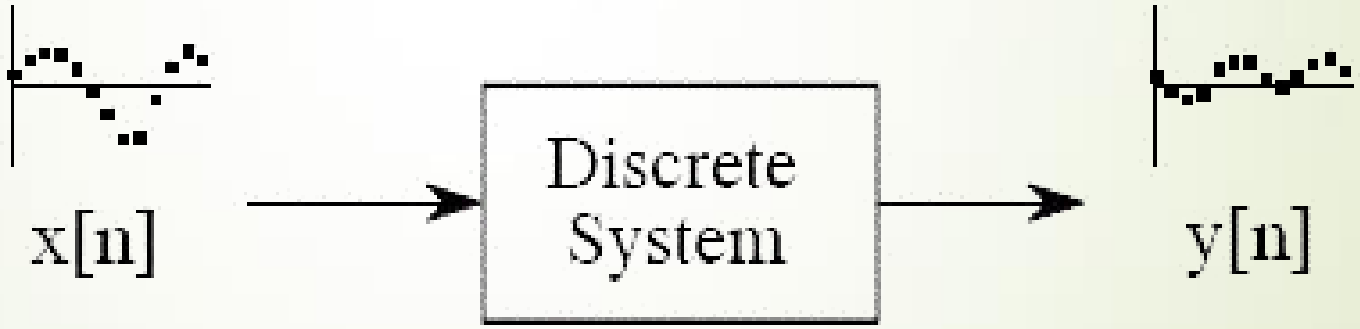
Continuous Time System

- Continuous systems are those types of systems in which input and output signals are the same at both the ends.
- In this type of system, variable changes with time and any type of variation is not found in the input and output signal.
- In response to the input signal, a continuous system generates an output signal.



Discrete Time System

- In discrete systems, both input and output signals are discrete signals. The variables in the discrete systems vary with time.
- In this type of system, the changes are predominantly discontinuous. The state of variables in discrete system changes only at a discrete set of points in time.



System Classifications Based On Properties

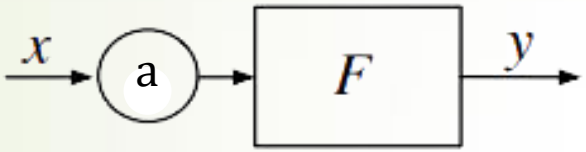
- System can be classified based on there types as:
  - **Linear** and **Non-linear** Systems
  - **Time Variant** and **Time Invariant Systems**
  - **System With Memory/Static** and **Memoryless/Dynamic Systems**
  - **Causal** and **Non-causal Systems**
  - **Invertible** and **Non-Invertible Systems**
  - **Stable** and **Unstable Systems**

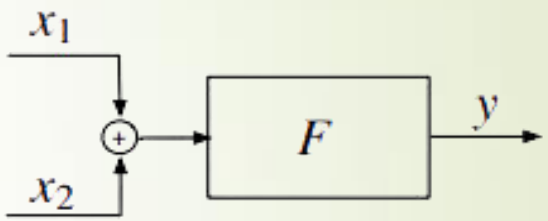
Linear and **Non-linear** Systems

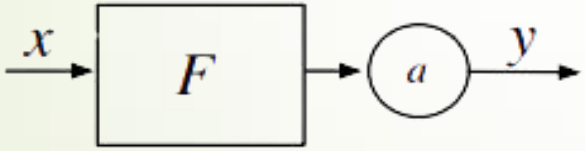
- This System Based on Linearity Property.
- A system F is linear if the following two properties hold:
  - **Homogeneity:**
    - if x is any signal and a is any scalar
    - $F(ax) = aF(x)$
  - **Superposition:**
    - if x and ˜x are any two signals,
    - $F(x + \tilde{x}) = F(x) + F(\tilde{x})$
- In words, linearity means:
  - Scaling before or after the system is the same.
  - Summing before or after the system is the same.

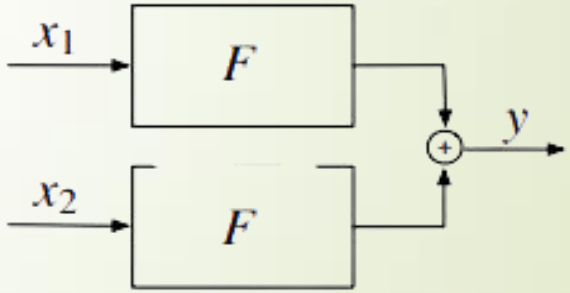
Linear and **Non-linear** Systems

- Linearity means the following pairs of block diagrams are equivalent, i.e., have the same output for any input(s)









- **Examples of linear systems:** scaling system, differentiator, integrator, running average, time shift, convolution, modulator, sampler.
- **Examples of nonlinear systems:** sign detector, multiplier (sometimes), comparator, quantizer, adaptive filter

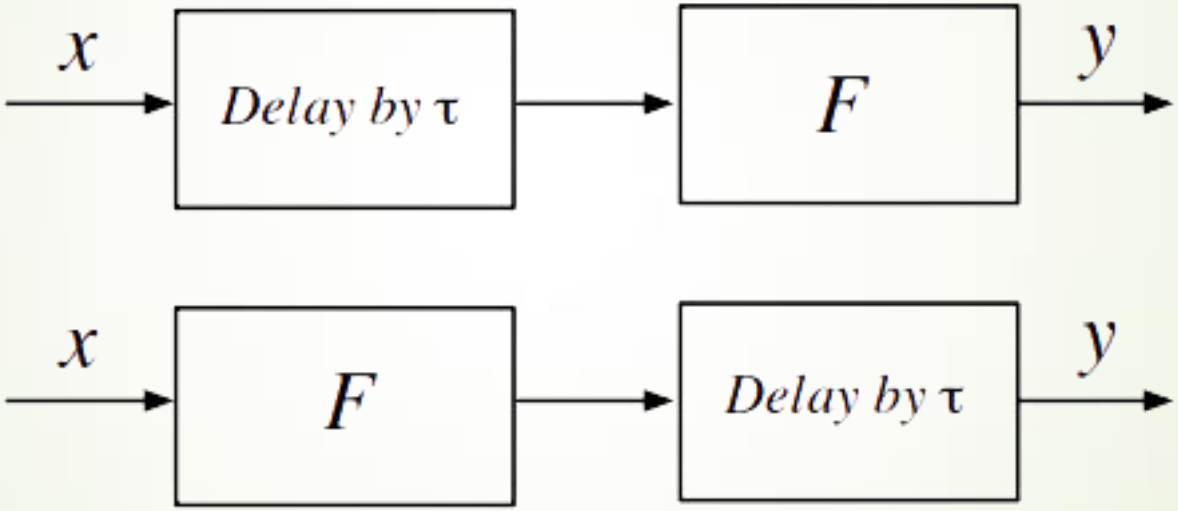
Time/Shift Variant and Time/Shift Invariant Systems

- This System Based on Time Invariance Property.
- **Time Shift:** A time shift refers to a displacement or translation of a signal along the time axis. It involves delaying or advancing the signal in time.
  - *Equation:* $y(t)=x(t-\tau)$
  - where τ is the amount of time shift.

- A system is time-invariant if a time shift in the input produces the same time shift in the output.
  - For a system F,
    - $y(t) = Fx(t)$ implies that
    - $y(t-\tau) = Fx(t-\tau)$ for any time shift τ .

Time/Shift Variant and Time/Shift Invariant Systems

- Implies that delay and the system F commute. These block diagrams are equivalent.



Causal and **Non-causal Systems**

- This System Based on Causality Property.
- **Causal:**
  - A system is said to be causal if its output depends upon present and past inputs, and does not depend upon future input.
- **Non-causal Systems**
  - For non causal system, the output depends upon future inputs also.
- **For Example:**
- **$y(n) = 0.5x(n) + 2.5x(n-2)$, for $n \geq 0$**
  - Since for $n \geq 0$, the output $y(n)$ depends on the current input $x(n)$ and its past value $x(n-2)$, the system is causal.
- **$y(n) = 0.25x(n-1) + 0.5x(n+1) - 0.4y(n-1)$, for $n \geq 0$**
  - Since for $n \geq 0$, the output $y(n)$ depends on the current input $x(n)$ and its future value $x(n+1)$, the system is noncausal.

System With Memory/ Dynamic and Memoryless/ Static Systems

- **Memoryless System:**
  - A system is memoryless if the output depends only on the present input.
    - Gain or Amplification Operation
    - If you have a system that multiplies the input signal by a constant factor, then it is a memoryless system. The output at any given time is solely determined by the current value of the input, and it does not take into account any past values.
- **System With Memory**:
  - A system with memory has an output signal that depends on inputs in the past or future.
    - FIR filters are examples of memory systems. The output at any given time depends on a weighted sum of the current and past input samples.
- A causal system has an output that depends only on past or present inputs.
  - Any real physical circuit, or mechanical system.

Invertible and **Non-Invertible Systems**

- This System Based on Invertibility Property.
- A system is invertible if the input signal can be recovered from the output signal.
- If F is an invertible system, and

$$y = Fx$$

- then there is an inverse system F INV such that

$$x = F\ INV\ y$$

$$x = F\ INV\ Fx$$

$$x = (F\ INV\ F)x$$

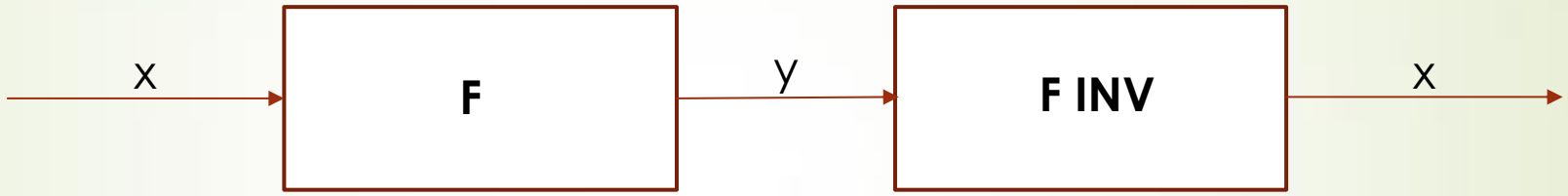
- so F INV F = I, the identity operator.

$$x = (I)x$$

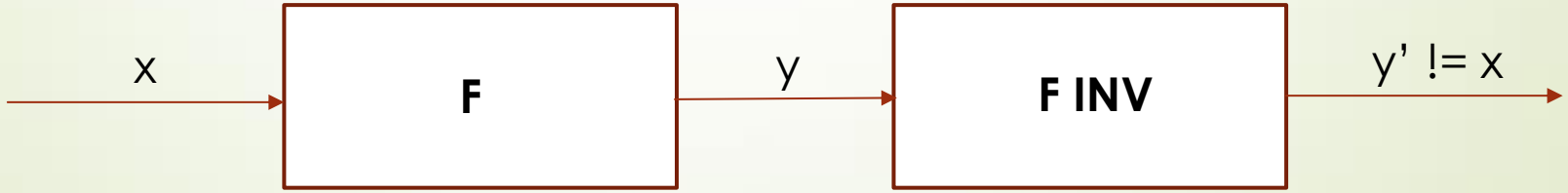
$$x=x$$

Invertible and Non-Invertible Systems

- Diagram: F is our system.



Invertible System



Non Invertible System

Stable and **Unstable Systems**

- This System Based on stability Property.
- **Stable System**
  - A system is stable if every bounded input or produces a bounded output or for all time or , or if the bounded-input bounded-output criterion is satisfied.
    - $|x(t)| \leq Mx < \infty$
  - always results in a bounded output
    - $|y(t)| \leq My < \infty$
  - where Mx and My are finite positive numbers, the system is stable.
- **Unstable System**
  - If a bounded input produces an unbounded output, then the system is unstable.

Linear-time Invariant System

Linearity and time-invariance are properties that describe the behavior of a system

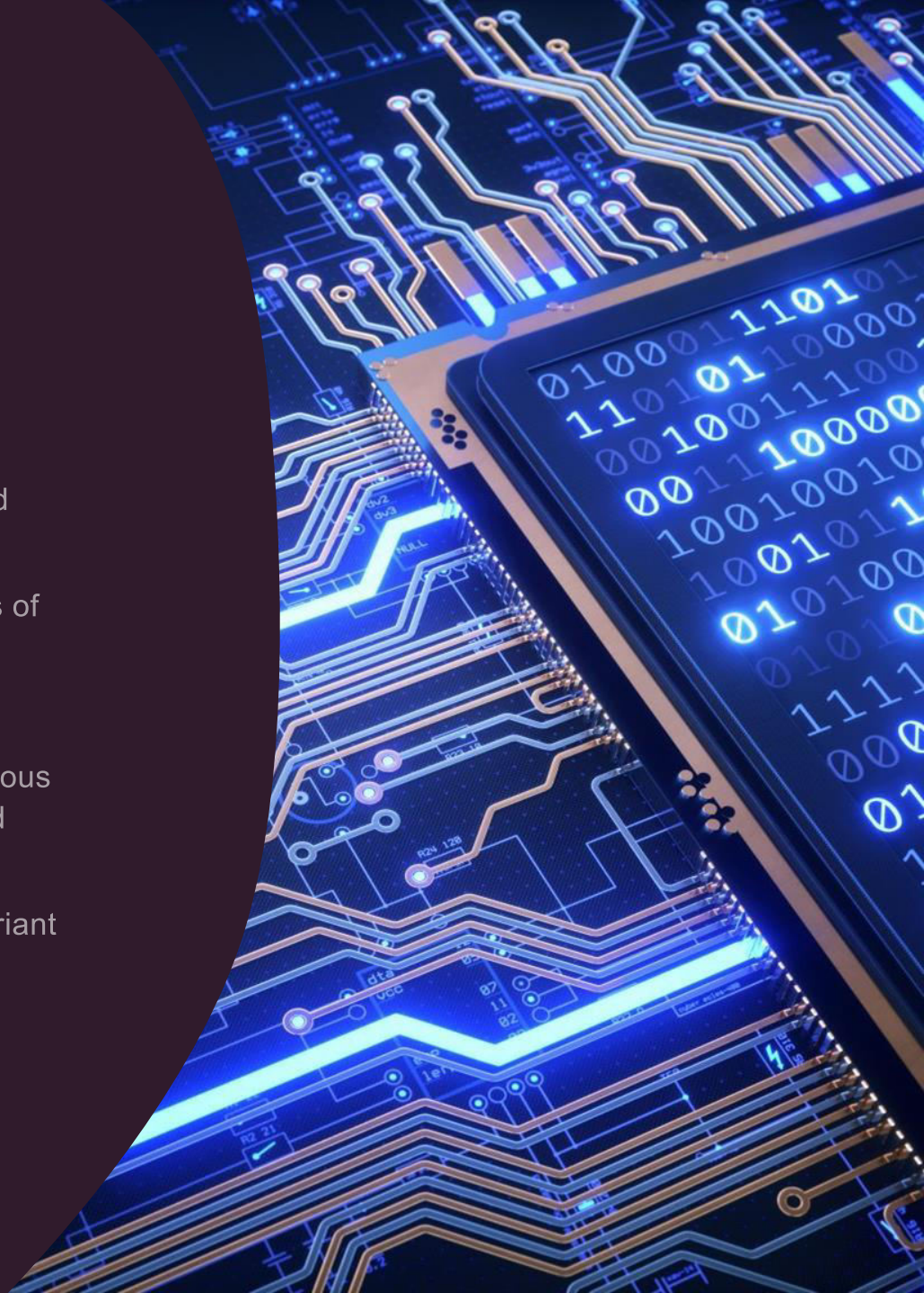
LTI systems are like reliable machines that respond proportionally to changes in input and maintain consistent behavior over time

They follow the principle of "double the input, double the output" and don't change their ways as time goes on, making them predictable and easy to understand

In simpler terms, LTI systems are steady and fair operators in the world of machines

Convolution

- Convolution is a mathematical operation that combines two signals & produces third signal.
- In linear systems, convolution is used to describe relationship between three signals of interest input signal , impulse response & output signal.
- Impulse response is the system output when the input is unit impulse.
- Convolution is a fundamental mathematical operation that plays a crucial role in various fields, including signal processing, statistics & probability, electrical engineering, and computer science.
- It is a powerful tool for analyzing and understanding the behavior of linear time-invariant systems.



Reason for using Convolution

There are two main reasons for using convolution:

- Understanding System Responses
- Signal Processing Magic

1. Understanding System Responses

- LTI System Response
- Impulse Response

2. Signal Processing Magic

- Filtering & Smoothing
- Cross Correlation

Methods to find discrete Convolution

- Tabular method
- Column method

Mathematical Representation of Discrete-time Convolution

- **The mathematical formula for discrete-time convolution is given by:**

$$y(n) = x(n) * h(n) = \sum[x(k) * h(n - k)]$$

- Convolution describes how an input sequence interacts with the impulse response of a system to generate the output sequence
  - `y ` represents the output signal at discrete time instant `n`
  - `x ` represents the input signal at discrete time instant `n`
  - `h ` represents the impulse response of the LTI system, which describes the system's response to a unit impulse input
  - `k` is the summation index
  - `*` denotes the convolution operation

Applications of Convolution

- Input-Output Relationship
- System Characterization
- System Analysis
- Signal Processing Techniques
- Filtering

Signal Processing Techniques

Convolution forms the basis for various signal processing techniques, including:

- Echo Cancellation: Convolution is used to remove echoes in communication systems by modeling the echo path and convolving it with an inverse filter
- Noise Reduction: Convolution-based noise reduction techniques aim to minimize the impact of noise on a signal by convolving the noisy signal with an inverse filter

Signal Processing Techniques

- Image Processing: Convolution operations are widely employed in image processing tasks, such as image filtering, edge detection, and image sharpening
- In summary, convolution plays a fundamental role in discrete-time systems analysis and signal processing

Advantages of Convolution

- Linear System Representation

1. mixing together
2. mixing separately

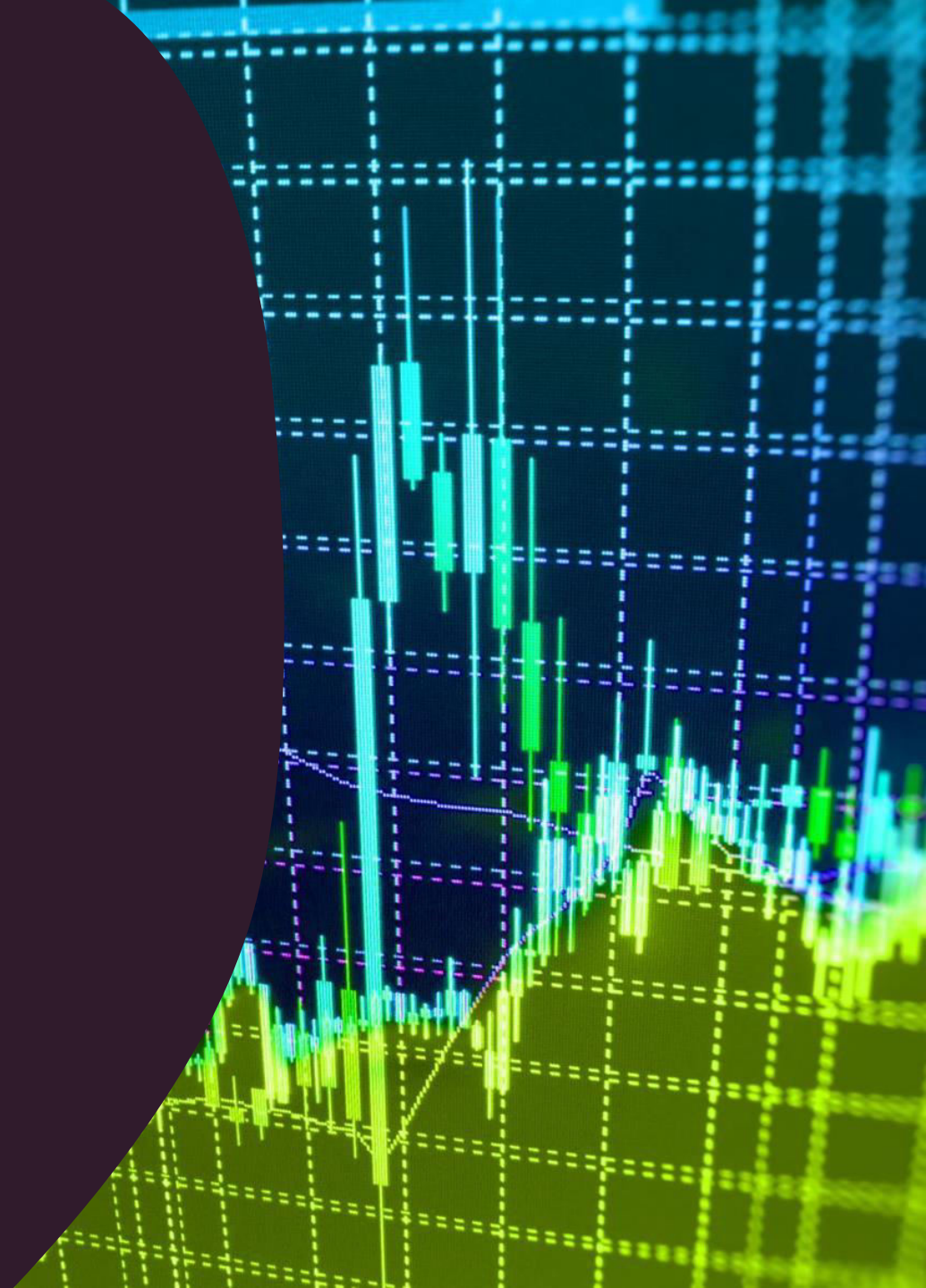
- Superposition Principle

1. analyze & understand the input

- Frequency Domain Analysis

1. time & frequency

- Filtering in Signal Processing



Advantages of Convolution

- Time-Domain Analysis
- Efficient Computation
- Impulse Response Representation
- Flexibility in System Analysis

Real-world examples

- Speech Recognition: Convolution is used in speech recognition systems to analyze and process audio signals
- Image Processing - Edge Detection: In image processing, convolution is applied for edge detection
- Medical Imaging - MRI Reconstruction: In medical imaging, convolution is employed in the reconstruction of Magnetic Resonance Imaging scans
- Financial Modeling - Time Series Analysis
- Convolution is applied in financial modeling for time series analysis
- Radar Signal Processing

What is Laplace Transformation

- Laplace transformation is a mathematical tool or technique used in engineering and physics to analyze linear time-invariant systems.
- It is a tool used to convert any signal of time domain into frequency domain.
- It is particularly useful for solving linear differential equation into a simple algebric equation.

Inverse Laplace transform

Definition:

The Inverse Laplace Transform is an operation that takes a function $F(s)$ in the Laplace domain and transforms it back to the original function $f(t)$ in the time domain.

Denoted by :

$$L-1\{F(s)\}=f(t).$$

The **general** form of the Inverse Laplace Transform is given by:

$$L-1\{F(s)\}=f(t)=2\pi i1\int\gamma-i\infty\gamma+i\infty estF(s)ds$$

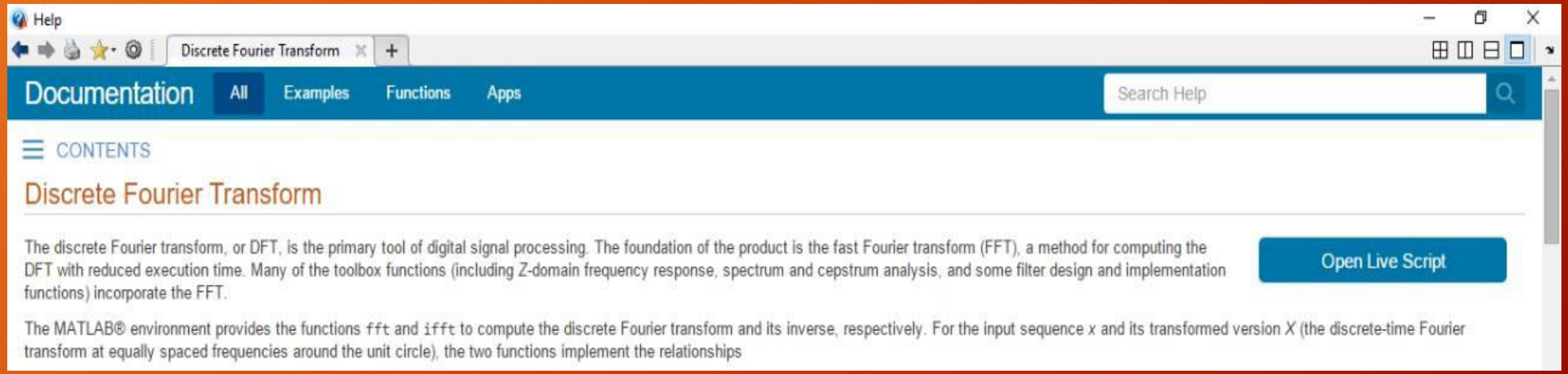
In this **expression**:

- $F(s)$ is the Laplace transform of the function $f(t)$.
- t is the time variable.
- The integration is taken along a contour in the complex plane (γ is a constant that lies to the right of all singularities of)$F(s)$).

DTFT

- The Discrete-Time Fourier Transform (DTFT) is a mathematical transformation used in signal processing to analyze the frequency content of a discrete.
- It is defined for a discrete sequence x[n], where "n" represents the discrete time index. The DTFT transforms a sequence in the time domain into a continuous function in the frequency domain.

DFT


Help
Discrete Fourier Transform
Documentation
All
Examples
Functions
Apps
Search Help
CONTENTS
Discrete Fourier Transform
The discrete Fourier transform, or DFT, is the primary tool of digital signal processing. The foundation of the product is the fast Fourier transform (FFT), a method for computing the DFT with reduced execution time. Many of the toolbox functions (including Z-domain frequency response, spectrum and cepstrum analysis, and some filter design and implementation functions) incorporate the FFT.
Open Live Script
The MATLAB® environment provides the functions fft and ifft to compute the discrete Fourier transform and its inverse, respectively. For the input sequence x and its transformed version X (the discrete-time Fourier transform at equally spaced frequencies around the unit circle), the two functions implement the relationships

Discrete Fourier Transform(DFT)

- The Discrete Fourier Transform (DFT) is a mathematical transformation that converts a finite sequence of equally spaced samples of a function into a finite sequence of complex numbers, which can be interpreted as the coefficients of a finite series of complex sinusoids.
- The DFT is widely used in signal processing and various other fields for analyzing the frequency content of discrete signals.

Properties

- Linearity
- Time shifting
- Frequency shifting

LAPLACE TRANSFORM

FORMULA

The Laplace transform of a function $f(t)$ is denoted by $F(s)$ and is defined as:

$$F(s) = \int_0^\infty e^{-st} \cdot f(t)\, dt$$

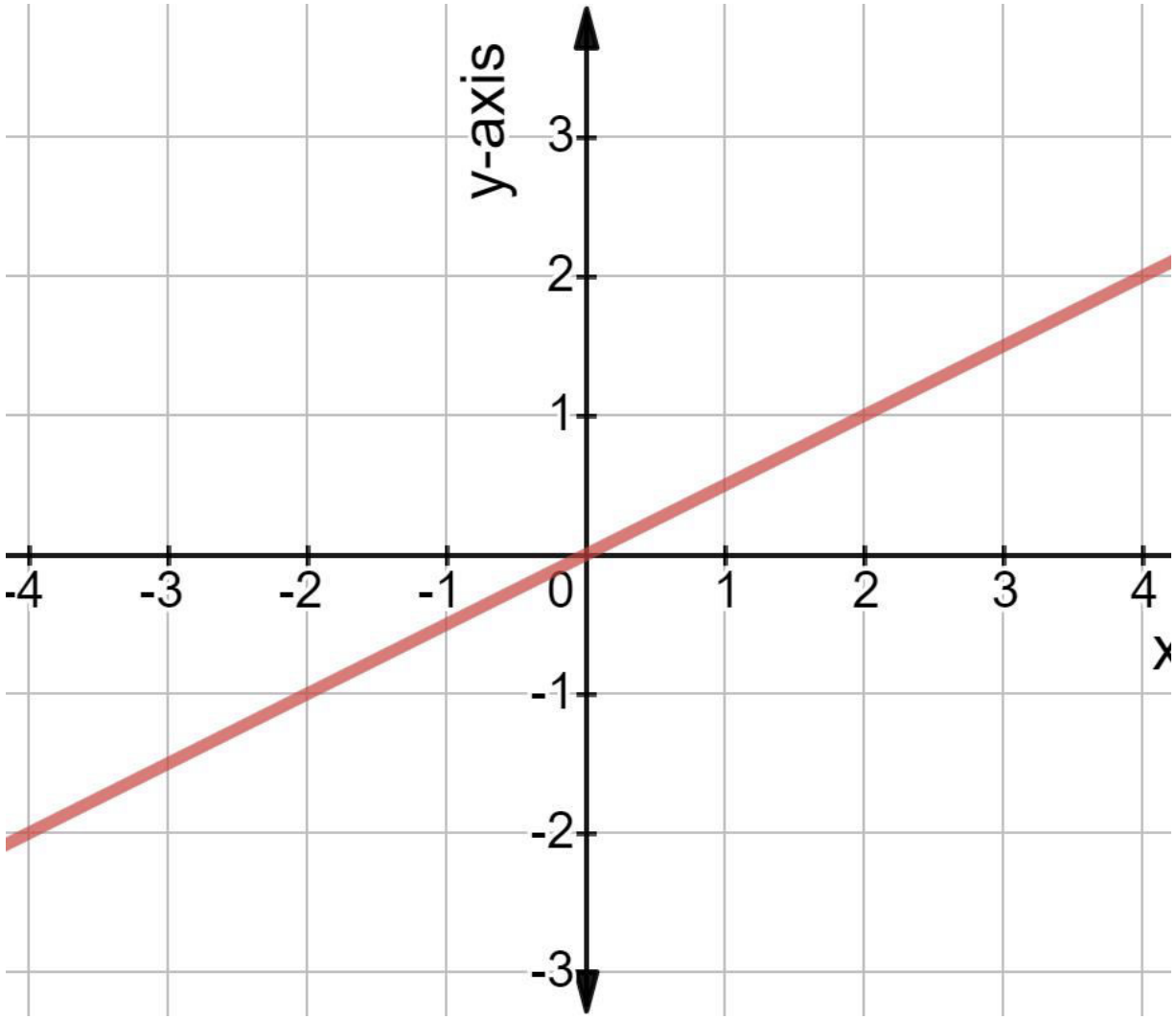
Where:

- $F(s)$ is the Laplace transform of $f(t)$.
- s is a complex variable.
- t represents time.
- e^{-st} is the exponential term that drives the transformation.
- The integral spans from 0 to ∞, indicating it's defined for functions that exist in that time range.

LINEARITY

The Laplace transform is linear, meaning it satisfies properties akin to a linear function.

$$L(a \cdot f(t) + b \cdot g(t)) = a \cdot F(s) + b \cdot G(s).$$



ROC (Region of Convergence) :

The set of values of z for which the z-transform converges is called the Region of Convergence (ROC).

In specific cases the inner radius of this ring may include the origin, and the outer radius may extend to infinity.

In digital signal processing (DSP), the Region of Convergence (ROC) is a concept associated with the analysis of signals and systems, particularly in the context of the Z-transform. The Z-transform is a mathematical transformation that converts a discrete-time signal or system representation from the time domain to the Z-domain. The Z-transform is analogous to the Laplace transform used in continuous-time signal and system analysis.

The Z-transform of a discrete-time signal x[n] is given by the expression:

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}$$

Similarly, for a discrete-time system represented by its impulse response h[n], the Z-transform is given by:

$$H(z) = \sum_{n=-\infty}^{\infty} h(n) z^{-n}$$

The Region of Convergence (ROC) is a set of values in the complex z-plane for which the Z-transform converges, meaning the sum in the Z-

transform expression converges and does not go to infinity. The ROC is essential for determining the stability and causality of a system.

The ROC is often denoted as a ring or annular region in the Z-plane. Its boundaries may be inside or outside certain circles or may extend to infinity. The specific location of the ROC depends on the properties of the discrete-time signal or system.

There are three possible **Types** of ROCs:

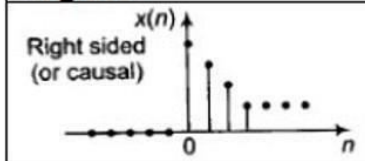
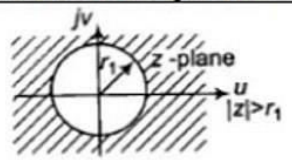
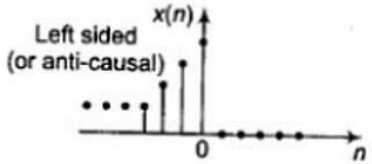
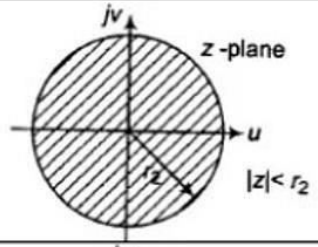
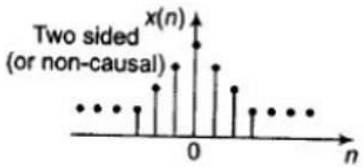
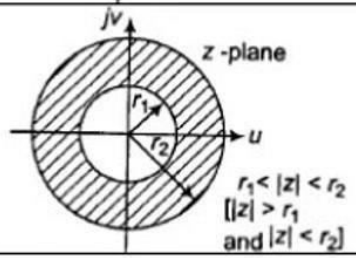
1). **Causal ROC (Right-sided ROC):** If the ROC includes the unit circle ($|z| = 1$) then ($|z| > r$) and extends outward, the system is causal.

2). **Anti-causal ROC (Left-sided ROC):** If the ROC includes the unit circle ($|z| = -1$) then ($|z| < r$) and move to origin, the system is anti-causal.

3). **Two-sided ROC:** If the ROC is a "ring" that includes the unit circle, the system is non-causal or two-sided.

Z-Transform

⇨ If **Z-Transform** of x(n) **is Rational** then ROC is bounded by poles and extended to infinity.
⇨ If **Z-Transform** of x(n) **is Rational** and **Right** sided then ROC is Z-plane outside the outermost pole.
⇨ If **Z-Transform** of x(n) **is Rational** and **left** sided then ROC is the region in Z-plane inside the innermost pole.

The ROC is crucial for understanding the convergence behavior of the Z- transform and provides insights into the stability and causality of the associated discrete-time system. Engineers and researchers use ROC analysis in the design and analysis of digital filters and other discrete-time systems in DSP applications.

Signal	ROC in z-plane
Infinite Duration Signals	
Signal	**ROC in z-plane**