Chapter 2: Discrete-time Signals and Systems[i].

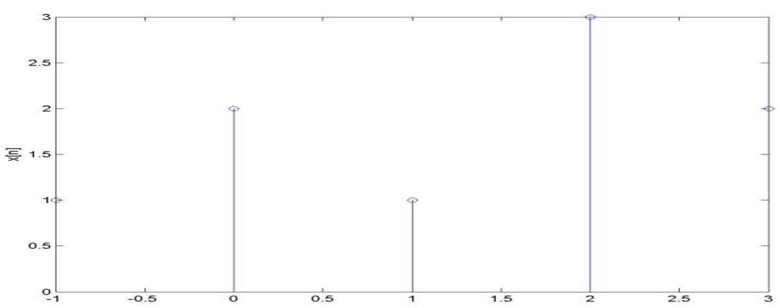
Discrete-time signals:

Digital signals are discrete in both time (the independent variable) and amplitude (the dependent variable). Signals that are discrete in time but continuous in amplitude are referred to as discrete-time signals. Discrete-time signals are data sequences. A sequence of data is denoted {x[n]} or simply x[n] when the meaning is clear. The elements of the sequence are called samples. The index n associated with each sample is an integer.

A discrete-time (D.T.) signals x[n] is a function of an independent variable that is an integer. It is important to note that a D.T. signal is not defined at instants between two successive samples. Also, it is incorrect to think that x[n] is equal to zero if n is not an integer. Simply, the signal x[n] is not defined for non-integer values of n.

Representation of D.T. Signals:

1) Graphical representation



2) Functional representation

$$x[n] = \begin{cases} 1 \ for\ n = -1, 1 \\ 2\ for\ n = 0, 3 \\ 3\ \ for\ n = 2 \\ 0\ elsewhere \end{cases}$$

3) Tabular representation

n	-1	0	1	2	3
x[n]	1	2	1	3	2

4) Sequence representation

$x[n] = \{1, \underset{\uparrow}{2}, 1, 3, 2\}$ where *"↑" sign represents the position of n =0. The arrow is often omitted if it is clear from the context which sample is x[0].*

Sample values can either be real or complex. The terms "discrete-time signals" and "sequences" are used interchangeably.

Some Elementary D.T. signals:

Elementary discrete-time signals can be defined as those signals that can be used as building blocks to build more complex signals and are normally used for the study of signals and systems. Conversely, we can decompose complex signals into elementary signals and analyze the behaviors of signals and systems. Followings are the discrete-time elementary signals we study in this course:

1) The **unit sample sequence** or unit impulse is defined as δ[n] and is defined as

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

Note: The analog signal δ(t) is defined to be zero everywhere except at t=0 and has unit area.

2) The **unit step signal** is denoted as u[n] and is defined as

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

3) The **unit ramp signal** is denoted as $u_r[n]$ and is defined as

$$u_r[n] = \begin{cases} n, for\ n \geq 0 \\ 0, for\ n < 0 \end{cases}$$

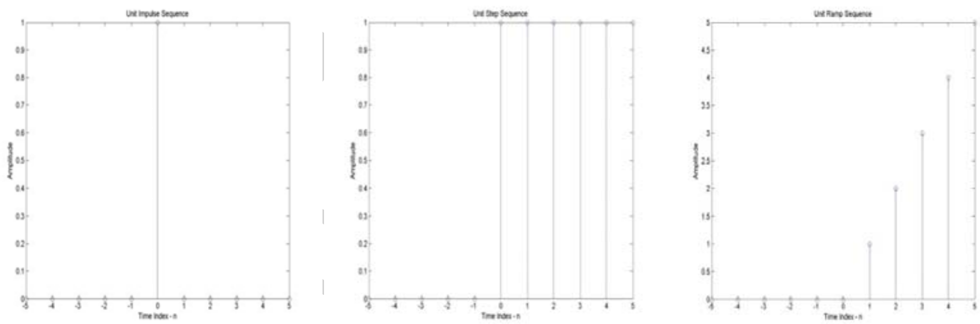


Fig: (a) Unit impulse (b) Unit Step (c) Unit Ramp sequences.

4) **Sinusoidal Signal**

$x[n] = A\cos(\omega n + \theta)$, $-\infty < n < \infty$

Where n is an integer variable, called the sample number, A is the *amplitude* of the sinusoid, ω is the frequency in radians per second, and θ is *phase* in radians.

5) **Exponential Signal**

The exponential signal is a sequence of the form

$$x[n] = a^n \text{ for all } n.$$

If the parameter a is real, then x[n] is a real signal. When the parameter a is complex valued,

$$a = re^{i\theta} \text{ (r \& } \theta \text{ are now parameters)}$$

Hence, $x[n] = r^n e^{i\theta n} = r^n(\cos\theta n + j\sin\theta n)$

The real part is $x_R[n] = r^n\cos\theta n$ and the imaginary part is $x_I[n] = r^n\sin\theta n$

Alternatively, the complex signal x[n] can be represented by the amplitude function
$|x[n]| = A(n) = r^n$
$\angle x[n] = \varphi(n) = \theta n$

Prove that

$$i. u[n] = \sum_{k=0}^{\infty} \delta[n-k]$$
$$ii. \delta[n] = u[n] - u[n-1]$$

Classification of Discrete-time signals:

1) **Energy signals & Power signals:**

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

If E is finite (i.e. $0 < E < \infty$), then x[n] is *energy signal.*

Many signals that possess infinite energy, have a finite average power. The average power of a periodic sequence with a period of N samples is defined as

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

And for non-periodic sequences, it is defined in terms of the following limit if it exists:

$$P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x[n]|^2$$

A signal with finite average power is called a power signal.

Find the average power of the unit step sequence u[n].
The unit step sequence is non-periodic, therefore the average power is

$$P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} u^2[n]$$
$$= \lim_{N\to\infty} \left(\frac{N+1}{2N+1}\right) = \frac{1}{2}$$

Therefore, the unit step sequence is a power signal. Note that its energy is infinite and so it is not an energy signal.

2) **Periodic signals & Aperiodic signals:**

The sinusoidal signal of the form
$x[n] = A\sin 2\pi f_0 n$ is periodic when f_0 is a rational number, that is if f_0 can be expressed as $f_0 = k/N$; where k and N are integers and N is the fundamental period.

3) **Symmetric(even) and antisymmetric(odd) signals:**

Even → $x[-n] = x[n]$ $x_e[n] = \frac{1}{2} \{x[n] + x[-n]\}$

Odd → $x[-n] = -x[n]$ $x_o[n] = \frac{1}{2} \{x[n] - x[-n]\}$

Transformation of the independent variable (time)

1) Shifting:

A signal x[n] may be shifted in time by replacing the independent variable n by n-k, where k is an integer. If k is positive, delay of the signal by k units of time and if k is negative, advance of the signal by k units in time.

2) Folding or a reflection of the signal about the time origin n=0:

A signal x[n] may be folded by replacing the independent variable n by –n.

Prove that the operations of folding and time shifting a signal are not commutative.

$$TD_k[x[n]] = x[n-k] \quad k>0$$

$$FD[x[n]] = x[-n]$$

Now, $TD_k[FD\{x[n]\}] = TD_k[x[-n]] = x[-n+k]$

Whereas, $FD[TD_k\{x[n]\}] = FD\{x[n-k]\} = x[-n-k]$

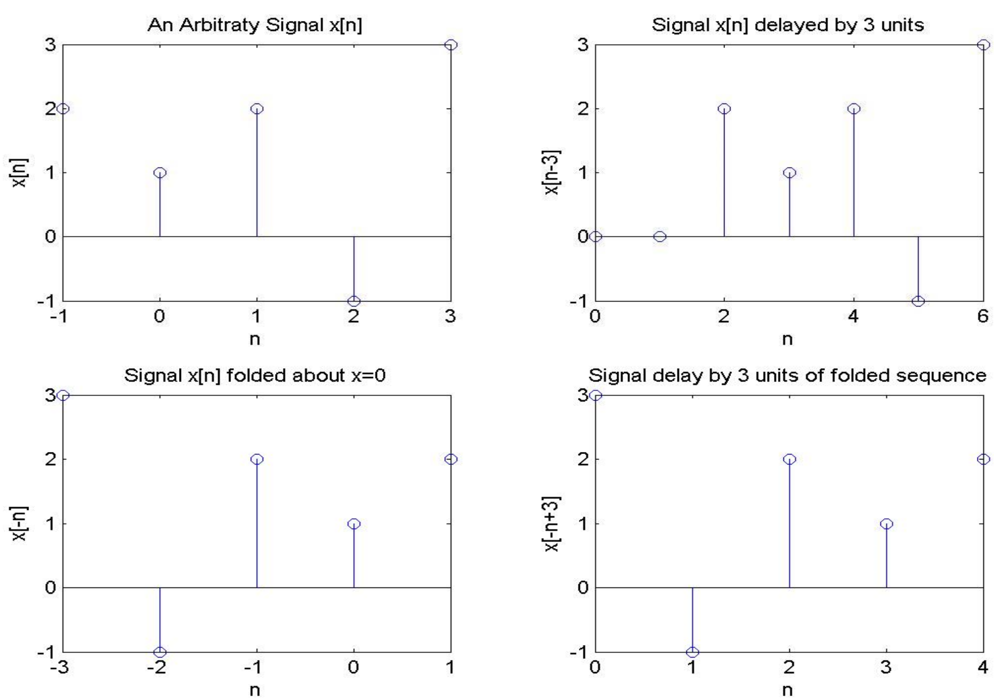
Note: Shifting of folded sequence x[-n] ---> x[-n+k]

If k positive, delay; if k negative, advance.

3) Time scaling or down-sampling:

-Replacing n by μn, where μ is an integer.

If the signal x[n] was originally obtained by sampling an analog signal $x_a(t)$, the $x[n] = x_a(nT)$, where T is the sampling interval. Now, $y[n] = x[2n] = x_a(2Tn)$. Hence the time-scaling operation is equivalent to changing the sampling rate from 1/T to 1/2T i.e. decreasing the rate by a factor of 2. This is a downsampling operation. Upsampling is not possible as we cannot obtain y[n] = x[n/2] from the signal x[n].



Note: Amplitude modification includes addition, multiplication, and scaling of D.T. signals.

Discrete-time Fourier series and properties:

In General Discrete-time Fourier series is given by,

$C_k = \frac{1}{N} \sum_{n=<N>} x[n] e^{-jr\left(\frac{2\pi}{N}\right)n}$	*Analysis equation*
$x[n] = \sum_{k=<N>} C_k\, e^{jk\omega_0 n}$	*Synthesis equation*

Where $\{C_k\}$ are spectral coefficients of x[n].

Properties of DTFS:

$$If x[n] \overset{F.S.}{\longleftrightarrow} C_k$$
$$\& y[n] \overset{F.S.}{\longleftrightarrow} D_k$$

Where x[n] and y[n] are periodic with period N & fundamental frequency $\omega_0 = 2\pi/N$. Also, C_k and D_k are periodic with period N.

There are strong similarities between the properties of discrete-time & continuous time Fourier series.

PROPERTY	PERIODIC SIGNAL	FOURIER SERIES
1. Linearity	Ax[n] + By[n]	$AC_k + BD_k$
2. Time-shifting	x[n – n_0]	$C_k e^{-\left(jk\frac{2\pi}{N}n_0\right)}$
3. Frequency-shifting	$e^{jM\left(\frac{2\pi}{N}\right)n} x[n]$	C_{k-M}
4. Conjugation	$x^*[n]$	C_{-k}^*
5. Time-reversal	$x[-n]$	C_k
6. Time scaling	$x_m[n] = \begin{cases} x\left[\frac{n}{m}\right] ifnismultipleofm \\ 0\ ifnisnotmultipleofm \end{cases}$ (period =>mN)	$\frac{1}{m} C_k$ (period =>mN)
7. Periodic convolution	$\sum_{r=<N>} x[r]y[n-r]$	NC_kD_k
8. Multiplication	$x[n]y[n]$	$\sum_{l=<N>} C_l D_{k-l}$
9. First Difference	$x[n] - x[n-1]$	$\left\{1 - e^{-jk\left(\frac{2\pi}{N}\right)}\right\} C_k$

10. Parseval's Relation for discrete-time periodic signals:

Parseval's relation for discrete-time periodic signals (Power Density Spectrum) is given by

$$\frac{1}{N} \sum_{n=<N>} |x[n]|^2 = \sum_{k=<N>} |C_k|^2$$

Where C_k are the Fourier series coefficients of x[n] and N is period.

Parsevals relation states that, "The average power in a periodic signal is equal to the sum of average powers in all of its harmonic components".

If x[n] is real (i.e. x[n] = x^*[n]), then

$$C_k^* = C_{-k}$$
$$or, |C_{-k}| = |C_k|\ \ (Even\ symmetry)$$
$$\arg(C_k) = -\arg(C_{-k})\ \ (Odd\ symmetry)$$

 # State and Prove time shifting property of DTFS

Solution: -> The DTFS is given by

$$C_k = \frac{1}{N}\sum_{n=<N>} x[n]e^{-jk\left(\frac{2\pi}{N}\right)n}$$

Time shifting property:

$$If x[n] \overset{F.s.}{\longleftrightarrow} C_k$$
$$Then x[n-l] \overset{F.s.}{\longleftrightarrow} C_k e^{-jk\left(\frac{2\pi}{N}\right)l}$$

Proof:

$$F.s.\{x[n-l]\} = \frac{1}{N}\sum_{n=<N>} x[n-l]e^{-jk\left(\frac{2\pi}{N}\right)n}$$

Let m = n – l, then

$$F.s.\{x[n-l]\} = \frac{1}{N}\sum_{m=<N>} x[m]e^{-jk\left(\frac{2\pi}{N}\right)(m+l)}$$
$$= \frac{1}{N}\sum_{m=<N>} x[m]e^{-jk\left(\frac{2\pi}{N}\right)m}e^{-jk\left(\frac{2\pi}{N}\right)l}$$
$$= e^{-jk\left(\frac{2\pi}{N}\right)l} \times \frac{1}{N}\sum_{m=<N>} x[m]e^{-jk\left(\frac{2\pi}{N}\right)m}$$
$$= e^{-jk\left(\frac{2\pi}{N}\right)l}C_k, \qquad Hence\ proved$$

When a periodic signal is shifted in time, the magnitudes of its Fourier series coefficients remain same.

Discrete-time Fourier transform and properties:

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

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

The discrete-time Fourier transform $X(e^{j\omega})$ is also known as spectrum of x[n]. The notations for Fourier transform pairs are:

$$F.T.\{x[n]\} = X(e^{j\omega})$$
$$F.T.^{-1}\{X(e^{j\omega})\} = x[n]$$
$$x[n] \overset{F.T.}{\Leftrightarrow} X(e^{j\omega})$$

Two important properties of DTFT are:

1. **Periodicity**

   The discrete-time Fourier transform $X(e^{j\omega})$ is continuous and periodic in ω with period 2π.

   $$X(e^{j\omega}) = X(e^{j(\omega+2\pi)})$$

   Implication: We need only one period of $X(e^{j\omega})$ (i.e. $\omega \in [0, 2\pi], or\ [-\pi, \pi]$ for analysis and not the whole domain $-\infty < \omega < \infty$

2. **Symmetry:**

   For real valued x[n], $X(e^{j\omega})$ is conjugate symmetry.

   $$X(e^{-j\omega}) = X^*(e^{j\omega})$$
   $$|X(e^{-j\omega})| = |X(e^{j\omega})| \quad => Even\ symmetry$$
   $$\measuredangle X(e^{-j\omega}) = -\measuredangle X(e^{j\omega}) \quad => Odd\ symmetry$$

3. Time Shifting

$$Let\ F.T.\{x[n]\} = X(k)$$

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

$$put\ m = n-k$$

$$= \sum_{n=-\infty}^{\infty} x[m]\, e^{-j\omega(m+k)}$$

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

4. Convolution

$$Let\ F.T.\{x_1[n]\} = X_1(k)\ \&\ F.T.\{x_2[n]\} = X_2(k)$$

$$F.T.\{x_1[n] * x_2[n]\} = \sum_{n=-\infty}^{\infty} \left[\sum_{k=-\infty}^{\infty} x_1[k] x_2[n-k] \right] e^{-j\omega n}$$

$$Interchanging\ the\ order\ of\ summation$$

$$= \sum_{k=-\infty}^{\infty} x_1[k] \sum_{n=-\infty}^{\infty} x_2[n-k]\, e^{-j\omega n}$$

Using shifting property

$$= \sum_{k=-\infty}^{\infty} x_1[k]\, e^{-j\omega k}. X_2(\omega) = X_1(\omega) X_2(\omega)$$

Other properties of DTFT are:

PROPERTY	Finite Energy SIGNAL/s	FOURIER Transform
Linearity	$Ax_1[n] + Bx_2[n]$	$AX_1(e^{j\omega}) + BX_2(e^{j\omega})$
Time-shifting	$x[n - n_0]$	$e^{-(j\omega n_0)} X(e^{j\omega})$
Frequency-shifting	$e^{j\omega_0 n} x[n]$	$X(e^{j(\omega-\omega_0)})$
Conjugation	$x^*[n]$	$X^*(e^{-j\omega})$
Time-reversal	$x[-n]$	$X(e^{-j\omega})$
Differentiation	$x[n] - x[n-1]$	$(1 - e^{-j\omega}) X(e^{j\omega})$
Convolution	$x[n] * h[n]$	$X(e^{j\omega}) H(e^{j\omega})$
Accumulation	$\sum_{m=-\infty}^{\infty} x[m]$	$\frac{1}{1-e^{-j\omega}} X(e^{j\omega}) + \pi X(e^{j0}) \sum_{k=-\infty}^{\infty} \delta(\omega - 2\pi k)$
Parseval's Relation	$\sum_{n=-\infty}^{\infty} \lvert x[n] \rvert^2 = \frac{1}{2\pi} \int_{2\pi} \lvert X(e^{j\omega}) \rvert^2 d\omega$	

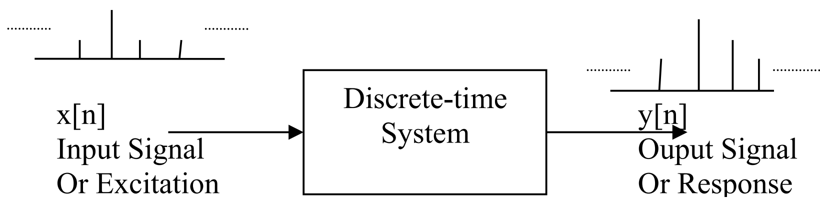
Discrete-time systems:

In many applications of digital signal processing we wish to design a device or an algorithm that performs some prescribed operation on a discrete-time signal. Such a device or algorithm is called a discrete-time system.

A discrete-time system is a device or algorithm that operates on a discrete-time signal, called the input or excitation, according to some well defined rule, to produce another discrete-time signal called the output or response of the system.

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

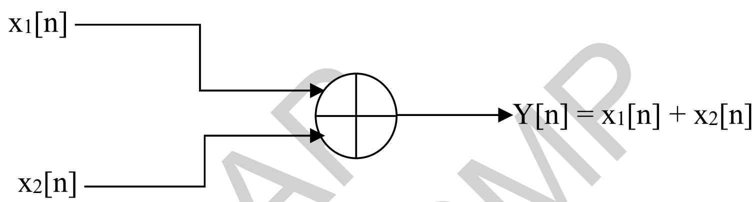
We say that the input signal x[n] is transformed by the system into a signal y[n].



Block Diagram Representation of DT Systems:

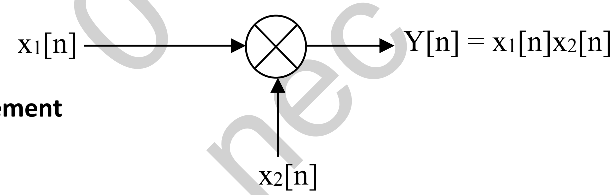
(Basic building blocks that can be interconnected to form complex systems)

- **An Adder:**



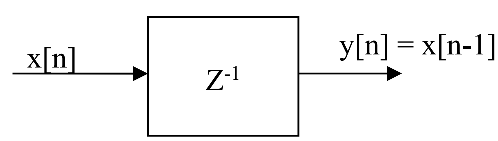
- **A Constant Multiplier**

- **A Signal Multiplier**

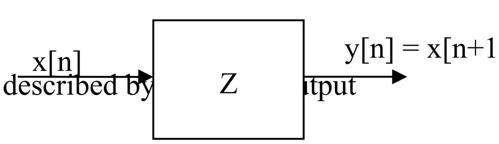


x1[n]

- **A Unit delay element**



- **A Unit advance element**



#Use basic building blocks to represent following DT system described by [illegible]tput relation:

$y[n] = 0.25y[n-1] + 0.5x[n] + 0.5x[n-1]$

Classification of Discrete-time systems:

1) Static (memoryless) vs dynamic (to have memory) systems:-

If the o/p of D.T. system at any instant n depends at most on the input sample at the same time, but not on past or future samples of the input then it is known as static system.
In any other case, the system is said to be dynamic.
$y[n] = x[n-N] \quad N \geq 0$
The system is said to have memory of duration N,
If N = 0, the system is static
$0<N<\infty$, the system is said to have finite memory.
$N = \infty$, the system is said to have infinite memory.

2) Time-invariant Vs. time-variant system:-

A system is called time-invariant if its input-output characteristics do not change with time.
Theorem:
A relaxed system T is time invariant or shift invariant if and only if $x[n] \rightarrow y[n]$
Implies $x[n-k] \rightarrow y[n-k]$
For every input signal x[n] and every time shift k.
For this we compute
$y[n, k] = T\{x[n-k]\}$ → response of delayed input and
$y[n-k]$ → delayed response
and check whether $y[n, k] = y[n-k]$ for all possible values of k.
if true → time invariant
& if $y[n, k] \neq y[n-k]$, even for one value of k, the system is time variant.

3) Linear Vs. nonlinear system:-

A linear system is one that satisfies the superposition theorem which states that, "A system is linear if and only if
$T\{a_1x_1[n] + a_2x_2[n]\} = a_1T\{x_1[n]\} + a_2T\{x_2[n]\}$
For any arbitrary i/p sequences $x_1[n]$ and $x_2[n]$, and any aribitrary constants a_1& a_2.

4) Causal Vs. noncausal systems:-

Theorem: "A system is said to be causal if the o/p of the system at any time n {i.e. y[n]} depends only on present and past inputs but doesnot depend on future inputs".
If a system doesnot satisfy this definition, it is called noncausal.

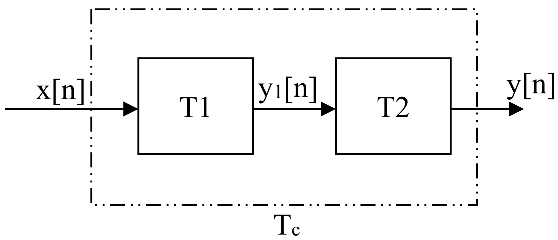
5) Stable Vs. unstable systems:-

An arbitrary relaxed system is said to be bounded input bounded output (BIBO) stable if and only if every bounded input produces a bounded output.
Mathematically,
If, $|x[n]| \leq M_x < \infty$
Then there must be $|y[n]| \leq M_y < \infty$ for all n, for the system to be stable. Where, M_x and M_y are finite numbers.

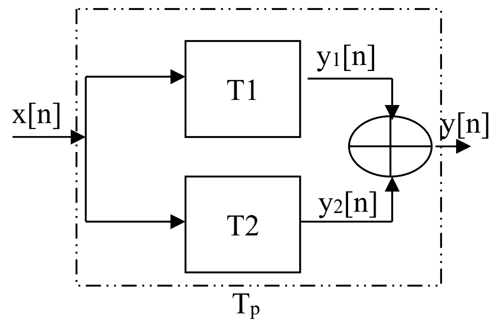
Interconnection of Discrete-time systems:

Discrete time systems can be interconnected to form larger systems. There are two basics ways in which systems can be interconnected:

1. Cascade (Series)
2. Parallel



(a) Cascade Interconnection



(b) Parallel Interconnection

In the cascade interconnection the output of the first systemis

$$y_1[n] = T_1\{x[n]\}$$

and the output of the second system is

$$y[n] = T_2\{y_1[n]\} = T_2[T_1\{x[n]\}]$$

We observe that system T_1 and T_2 can be combined or consolidated into a single overall system

$$T_c \equiv T_2T_1$$

We can express the output of the combined system as,

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

In general, for arbitrary systems,

$$T_2T_1 \neq T_1T_2$$

However, if the systems T_1 and T_2 are linear and time-invariant, then (a) T_c is time invariant and (b) $T_2T_1 = T_1T_2$

Proof of (a)

Suppose that T_1 and T_2 are time-invariant, then

$x[n-k]$ — $(T_1)\rightarrow$ $y_1[n-k]$

$y_1[n-k]$ —$(T_2)\rightarrow$ $y[n-k]$

Thus, $x[n-k]$—$(T_c)\rightarrow$ $y[n-k]$

And therefore, T_c is time invariant.

In the parallel interconnection, the output of the system T_1 is $y_1[n]$ and the output of the system T_2 is $y_2[n]$. Hence,

$y_3[n] = y_1[n] + y_2[n] = T_1\{x[n]\} + T_2\{x[n]\} =(T_1 + T_2)\{x[n]\} = T_p\{x[n]\}$.

Where, $T_p = T_1 + T_2$

In general, we can use parallel and cascade interconnection of systems to construct larger, more complex systems. Conversely, we can take a larger system and break it down into smaller subsystems for purposes of analysis and implementation.

Analysis of Discrete-time linear time-invariant(LTI) systems:

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

- **Convolution method:**

This method for analyzing the behavior of a linear system to a given input signal is first to decompose or resolve the input signal into a sum of elementary signals. Then, using the linearity property of the system, the responses of the system to the elementary signals are added to obtain the total response of the system to the given input signal.

- **Linear constant coefficient difference (LCCD) equation method:**

This method is based on the direct solution of the input-output equation for the system, and has the form
y[n] = F{y[n-1], y[n-1], ..., y[n-N], x[n], x[n-1], x[n-2], ..., x[n-M] }
Specifically, for an LTI system, the general form of the input-output relationshiop is,

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

Where $\{a_k\}$ and $\{b_k\}$ are constant parameters that specify the system and are independent of x[n] and y[n].

Resolution of a Discrete-time signal into impulses:

Suppose we have an arbitrary signal x[n] that we wish to resolve into a sum of unit sample sequences. We select the elementary signals $x_k[n]$ to be

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

where k represents the delay of the unit sample sequence.
Now suppose that we multiply the two sequences x[n] and δ[n-k], since δ[n-k] is zero everywhere except at n = k, where its value is unity, the result of this multiplication is another sequence that is zero everywhere except at n = k, where its value is x[k]. Thus
x[n]δ[n-k] = x[k]δ[n-k]
If we were to repeat the multiplication of x[n] with δ[n-m] where m is another delay (m≠k), the result will be a sequence that is zero everywhere except at n = m, where its value is x[m]. Hence,
x[n]δ[n-m] = x[m]δ[n-m]
Consequently, if we repeat this multiplication over all possible delays $-\infty < k < \infty$, and sum all the product sequences, the result will be a sequence equal to the sequence x[n], that is,

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

The right hand side of the equation gives the resolution of or decomposition of any arbitrary signal x[n] into a weighted (scale) sum of shifted unit samples sequences.
Response of LTI systems to arbitrary inputs: The Convolution Sum
We know,

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

First, we denote the response y[n ,k] of the system to the input unit sample sequence at n = k by the special symbol h[n, k]; $-\infty < k < \infty$, i.e.
y[n, k] = h[n, k] = T{δ[n-k]}
we note that n is the time index and k is a parameter showing the location of the input impulse.
If the impulse at the input is scaled by an amount C_k = x[k], the response of the system is the corresponding scaled output, that is
C_kh[n, k] = x[k]h[n, k]
The response to any arbitrary input expressed as a sum of weighted impulses, as given above is
y[n] = T{x[n]} = T{

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

Where h(n,k) is the response of unit impulses δ[n-k] $-\infty < k < \infty$.
If the response of the LTI system to the unit sample sequence δ[n] is denoted by h[n], i.e.
h[n] ≡ T{δ[n]}
then by the time-invariant property, the response of the system to the delayed unit sample sequence δ[n-k] is,
h[n-k] = T{δ[n-k]}
then,

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

The formula in above equation that gives the response y[n] of the LTI system as a function of the input signal x[n] and the unit impulse h[n] is called a convolution sum.

Methods to calculate Convolution Sum:

1. Mathematically (by using direct formula of convolution sum).
2. Graphically (by using following procedure)
3. Matrix Method

Procedure to compute the convolution sum:

Suppose that we wish to compute the output of the system at some time instant say $n = n_0$, then

$$y[n_0] = \sum_{k=-\infty}^{\infty} x[k]h[n_0-k]$$

The procedures are:

Plot x[n] and h[n] as x[k] and h[k] then,

1) Folding: Fold h[k] about k = 0 to obtain h[-k].
2) Shifting: Shift h[-k] by n_0 to the right (left) if n_0 is positive (negative), to obtain $h[n_0-k]$.
3) Multiplication: Multiply x[k] by $h[n_0-k]$ to obtain the product sequence $v_{n0}(k) \equiv x[k]h[n_0-k]$.
4) Summation: Sum all the values of the product sequence $v_{n0}(k)$ to obtain the value of the output at time $n = n_0$.
5) Repetition: Repeat steps 2 through 4, for all possible time shifts $-\infty < n < \infty$ to obtain overall response.

Properties of Convolution:

To simplify the notation, we denote the convolution operations as,

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

1) Commutative Law: x[n] * h[n] = h[n] * x[n]
2) Associative Law: $(x[n] * h_1[n]) * h_2[n] = x[n] * (h_1[n] * h_2[n])$
3) Distributive law: $x[n] * (h_1[n] + h_2[n]) = x[n] * h_1[n] + x[n] * h_2[n]$

Causal Linear Time Invariant Systems:

Causal System: whose output at time n depends only on present and past inputs but does not depends on future inputs. i.e. Output of system x[n] at instant $n = n_0$ depends on values of x[n] for $n \leq n_0$.

Causal LTI System:

Let us consider an LTI system having an output at time $n = n_0$ by

$$y[n_0] = \sum_{k=-\infty}^{\infty} h[k]x[n_0 - k]$$

Suppose that we subdivide the sum into two sets of term, one involving negative terms and other involving positive terms of values of k.

$$y[n_0] = \sum_{k=0}^{\infty} h[k]x[n_0 - k] + \sum_{k=-\infty}^{-1} h[k]x[n_0 - k]$$

$=\{h[0]x[n_0] + h[1]x[n_0 - 1]+ h[2]x[n_0 - 2] + - - - - - - \} + \{h[-1]x[n_0 + 1]+ h[-2]x[n_0 +2] + - - - \}$

We observe that the terms in the first sum involve $x[n_0]$, $x[n_0 - 1]$, - - - which are the present and past values. On the other hand, the terms in the second sum involves the input signals $x[n_0 + 1]$, $x[n_0 + 2]$, - - -. Now, if the output at time $n = n_0$ is to depend only on the present and past values, then it is clear that

$h[n] = 0$ for $n < 0$

It is both a necessary and a sufficient condition for causality. Hence "*An LTI system is causal if and only if its impulse response is zero for negative values of n.*"

Thus for causal LTI system,

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

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

Stability of Linear Time Invariant Systems:

Stable System: An arbitrary relaxed system is BIBO stable if and only if its output sequence y[n] is bounded for every bounded input x[n].

For an LTI system:

Suppose we have an LTI system,

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

Taking absolute values on both sides,

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

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

If the input is bounded, there exists a finite number M_x such that $|x[n]| \leq M_x$

$$|y[n]| \leq M_x \sum_{k=-\infty}^{\infty} |h[n]|$$

The output is bounded if the impulse response of the system satisfies the condition

$$S_h \equiv \sum_{k=-\infty}^{\infty} |h[n]| < \infty$$

"*A linear time-invariant system is stable if its impulse response is absolutely summable*". This is the necessary and sufficient condition for stable LTI system.

Systems with Finite-Duration and Infinite Duration Impulse Response:

Finite Duration Impulse Response System (FIR system):

h[n] = 0 for n < 0 and n ≥ M,

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

FIR has a finite memory of length M-samples.

Infinite Duration Impulse Response System (IIR System)

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

IIR has an infinite memory.

Discrete-time system described by Difference Equations:

The convolution summation formula is given by

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

Above equation suggests a means for the realization of the system. In the case of FIR systems, such a realization involves additions, multiplications, and a finite number of memory location, which is readily implemented directly.

If the system is IIR, however, its practical implementation as given by convolution is clearly impossible, since it requires an infinite number of memory locations, multiplications, and additions.

There is a practical and computationally efficient means for implementing a family of IIR systems, within the general class of IIR systems; this family of discrete-time systems is more conveniently described by difference equations.

Recursive & Non-Recursive Discrete-time systems:

If the response of any discrete-time systems depends only on the input signals (i.e. terms of the input signal) then the system is known as non-recursive discrete time system.

If we can express the output of the system not only in terms of the present and past values of the input, but also in terms of the past output values, then that system is known as recursive system.

Eg: the cumulative average of a signal x[n] in the interval 0 ≤ k ≤ n defined as,

$$y[n] = \frac{1}{n+1}\sum_{k=0}^{n} x[k] \qquad n = 0, 1, 2, \ldots..$$

The realization of above equation requires the storage of all the input samples. Since n is increasing, our memory requirements grow linearly with time.

However, y[n] can be computed more efficiently by utilizing the previous output value y[n-1]. By a simple algebraic rearrangement, we obtain,

$$(n+1)y[n] = \sum_{k=0}^{n-1} x[k] + x[n] = ny[n-1] + x[n]$$

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

Now the cumulative average y[n] can be computed recursively by multiplying the previous output value y[n-1] by n/(n+1), multiplying the present input x[n] by 1/(n+1), and adding the two products.

This is an example of recursive system. In general whose output y[n] at time n depends on any number of past output values y[n-1], y[n-2], ……, is called a recursive system.

The output of a causal and practically realizable recursive system can be expressed in general as,

y[n] = F{y[n-1], y[n-2], ……, y[n-N], x[n], x[n-1], ….., x[n], x[n-1], ….., x[n-M]}

If y[n] depends only on the present and past inputs, then

y[n] = F{ x[n], x[n-1], ….., x[n], x[n-1], ….., x[n-M]}

Such a system is called non-recursive.

In recursive system, we need to compute all the previous values y[0], y[1], y[2], …., $y[n_0 - 1]$ to compute $y[n_0]$ but in non-recursive system, we can compute the $y[n_0]$ immediately without having $y[n_0-1]$, $y[n_0-2]$, ……. This feature is desirable in some practical applications.

Difference Equations:

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

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k]\ a_0 \equiv 1$$

Or, equivalently,

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

If $a_N \neq 0$, then the difference equation is of order N. This equation describes a recursive approach for computing the current output, given the input values & previously computed output-values.

If the system described by difference equation has a constant coefficient (independent of time) then it is known as linear constant coefficient difference (LCCD) equation.

Consider the first order system (i.e. N = 1) & M = 0.

y[n] = ay[n-1] + x[n]

Now,

y[0] = ay[-1] + x[0]

y[1] = ay[0] + x[1] = a^2y[-1] + ax[0] + x[1]

y[2] = ay[1] + x[2] = a^3y[-1] + a^2x[0] + ax[1] + x[2]

⋮ ⋮ ⋮ ⋮

y[n] = ay[n-1] + x[n] = a^{n+1}y[-1] +a^nx[0] + a^{n-1}x[1] + ….

$$= a^{n+1}y[-1]\sum_{k=0}^{n} a^k x[n-k] \qquad n \geq 0$$

the response contain two parts; the first part is the result of the initial condition y[-1] of the system and second part is the response of the system to the input signal x[n].

if the system is initially relaxed at time n = 0, then its memory should be zero. Hence y[-1] = 0 (state = output of the delay element).

In this case, we say that the system is at zero state and its corresponding output is called zero-state response or forced response and is denoted by y_{zs}.

$$y_{zs}[n] = \sum_{k=0}^{n} a^k x[n-k] \qquad n \geq 0$$

We note that above equation is a convolution summation involving the input signal convolved with the impulse response $h[n] = a^n u[n]$.

We obtained the result that the relaxed recursive system described by the first order difference equation is a linear time-invariant IIR system with impulse response given by $h[n] = a^n u[n]$.

Now, suppose the system is initially non-relaxed (i.e. $y[-1] \neq 0$) and the input $x[n] = 0$ for all n. the output of the system with zero input is called the zero-input response or natural response and is denoted by $y_{zi}[n]$.

$$y_{zi}[n] = a^{n+1} y[-1] \quad n \geq 0$$

We observe that a recursive system with nonzero initial condition is non relaxed in the sense that it can produce an output without being excited.

Linearity, time invariance and Stability of the system described by LCCD equation

A system is linear if it satisfies the following three requirements:

1. The total response is equal to the sum of the zero-input and zero-state responses. (i.e. $y[n] = y_{zi}[n] + y_{zs}[n]$)
2. The principle of superposition applies to the zero-state response.
3. The principle of superposition applies to the zero-input response.

Else the system is non-linear.

In general, recursive systems described by the constant-coefficient difference equation is linear and time-invariant. (because the coefficients a_k and b_k are constants)

References:

1. J. G. Proakis, D. G. Manolakis, "Digital Signal Processing, Principles, Algorithms and Applications", 3rd Edition, Prentice-hall, 2000. Chapter 2.