UNIT - 1
SIGNALS AND SYSTEM

INTRODUCTION

A **SIGNAL** is defined as any physical quantity that changes with time, distance, speed, position, pressure, temperature or some other quantity. A **SIGNAL** is physical quantity that consists of many sinusoidal of different amplitudes and frequencies.

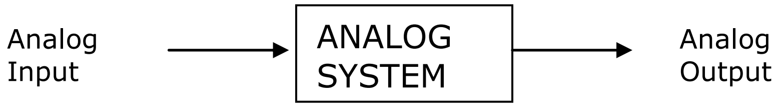
Ex $x(t) = 10t$

$X(t) = 5x^2+20xy+30y$

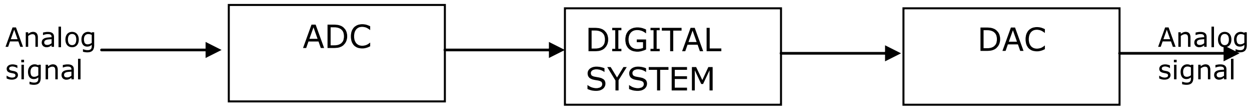
A **System** is a physical device that performs an operations or processing on a signal. Ex Filter or Amplifier.

1.1 CLASSIFICATION OF SIGNAL PROCESSING

1) ASP (Analog signal Processing) : If the input signal given to the system is analog then system does analog signal processing. Ex Resistor, capacitor or Inductor, OP-AMP etc.



2) DSP (Digital signal Processing) : If the input signal given to the system is digital then system does digital signal processing. Ex Digital Computer, Digital Logic Circuits etc. The devices called as ADC (analog to digital Converter) converts Analog signal into digital and DAC (Digital to Analog Converter) does vice-versa.



Most of the signals generated are analog in nature. Hence these signals are converted to digital form by the analog to digital converter. Thus AD Converter generates an array of samples and gives it to the digital signal processor. This array of samples or sequence of samples is the digital equivalent of input analog signal. The DSP performs signal processing operations like filtering, multiplication, transformation or amplification etc operations over these digital signals. The digital output signal from the DSP is given to the DAC.

ADVANTAGES OF DSP OVER ASP

1. Physical size of analog systems is quite large while digital processors are more compact and light in weight.
2. Analog systems are less accurate because of component tolerance ex R, L, C and active components. Digital components are less sensitive to the environmental changes, noise and disturbances.
3. Digital system is most flexible as software programs & control programs can be easily modified.
4. Digital signal can be stores on digital hard disk, floppy disk or magnetic tapes. Hence becomes transportable. Thus easy and lasting storage capacity.
5. Digital processing can be done offline.

6. Mathematical signal processing algorithm can be routinely implemented on digital signal processing systems. Digital controllers are capable of performing complex computation with constant accuracy at high speed.
7. Digital signal processing systems are upgradeable since that are software controlled.
8. Possibility of sharing DSP processor between several tasks.
9. The cost of microprocessors, controllers and DSP processors are continuously going down. For some complex control functions, it is not practically feasible to construct analog controllers.
10. Single chip microprocessors, controllers and DSP processors are more versatile and powerful.

Disadvantages of DSP over ASP

1. Additional complexity (A/D & D/A Converters)
2. Limit in frequency. High speed AD converters are difficult to achieve in practice. In high frequency applications DSP are not preferred.

1.2CLASSIFICATION OF SIGNALS

1. Single channel and Multi-channel signals
2. Single dimensional and Multi-dimensional signals
3. Continuous time and Discrete time signals.
4. Continuous valued and discrete valued signals.
5. Analog and digital signals.
6. Deterministic and Random signals
7. Periodic signal and Non-periodic signal
8. Symmetrical(even) and Anti-Symmetrical(odd) signal
9. Energy and Power signal

1.2.1 Single channel and Multi-channel signals

If signal is generated from single sensor or source it is called as single channel signal. If the signals are generated from multiple sensors or multiple sources or multiple signals are generated from same source called as Multi-channel signal. Example ECG signals. Multi-channel signal will be the vector sum of signals generated from multiple sources.

1.2.2Single Dimensional (1-D) and Multi-Dimensional signals (M-D)

If signal is a function of one independent variable it is called as single dimensional signal like speech signal and if signal is function of M independent variables called as Multi-dimensional signals. Gray scale level of image or Intensity at particular pixel on black and white TV is examples of M-D signals.

1.2.3 Continuous time and Discrete time signals.

S No	Continuous Time (CTS)	Discrete time (DTS)
1	This signal can be defined at any time instance & they can take all values in the continuous interval(a, b) where a can be -∞ & b can be ∞	This signal can be defined only at certain specific values of time. These time instance need not be equidistant but in practice they are usually takes at equally spaced intervals.
2	These are described by differential equations.	These are described by difference equation.
3	This signal is denoted by x(t).	These signals are denoted by x(n) or

		notation x(nT) can also be used.
4	The speed control of a dc motor using a tacho generator feedback or Sine or exponential waveforms.	Microprocessors and computer based systems uses discrete time signals.

1.2.4 Continuous valued and Discrete Valued signals.

S No	Continuous Valued	Discrete Valued
1	If a signal takes on all possible values on a finite or infinite range, it is said to be continuous valued signal.	If signal takes values from a finite set of possible values, it is said to be discrete valued signal.
2	Continuous Valued and continuous time signals are basically analog signals.	Discrete time signal with set of discrete amplitude are called digital signal.

1.2.5 Analog and digital signal

Sr No	Analog signal	Digital signal
1	These are basically continuous time & continuous amplitude signals.	These are basically discrete time signals & discrete amplitude signals. These signals are basically obtained by sampling & quantization process.
2	ECG signals, Speech signal, Television signal etc. All the signals generated from various sources in nature are analog.	All signal representation in computers and digital signal processors are digital.

Note: Digital signals (**DISCRETE TIME & DISCRETE AMPLITUDE**) are obtained by sampling the **ANALOG** signal at discrete instants of time, obtaining **DISCRETE TIME** signals and then by quantizing its values to a set of discrete values & thus generating **DISCRETE AMPLITUDE** signals.

Sampling process takes place on x axis at regular intervals & quantization process takes place along y axis. Quantization process is also called as rounding or truncating or approximation process.

1.2.6 Deterministic and Random signals

Sr No	Deterministic signals	Random signals
1	Deterministic signals can be represented or described by a mathematical equation or lookup table.	Random signals that cannot be represented or described by a mathematical equation or lookup table.
2	Deterministic signals are preferable because for analysis and processing of signals we can use mathematical model of the signal.	Not Preferable. The random signals can be described with the help of their statistical properties.
3	The value of the deterministic signal can be evaluated at time (past, present or future) without certainty.	The value of the random signal can not be evaluated at any instant of time.
4	Example Sine or exponential waveforms.	Example Noise signal or Speech signal

1.2.7 Periodic signal and Non-Periodic signal

The signal x(n) is said to be periodic if x(n+N)= x(n) for all n where N is the fundamental period of the signal. If the signal does not satisfy above property called as Non-Periodic signals.

Discrete time signal is periodic if its frequency can be expressed as a ratio of two integers. f= k/N where k is integer constant.

a) cos (0.01 ∏ n)	Periodic N=200 samples per cycle.
b) cos (3 ∏ n)	Periodic N=2 samples
c) sin(3n)	Non-periodic
d) cos(n/8) cos(∏n/8)	Non-Periodic

1.2.8 Symmetrical(Even) and Anti-Symmetrical(odd) signal

A signal is called as symmetrical(even) if x(n) = x(-n) and if x(-n) = -x(n) then signal is odd. X1(n)= cos(ωn) and x2(n)= sin(ωn) are good examples of even & odd signals respectively. Every discrete signal can be represented in terms of even & odd signals.

X(n) signal can be written as

$$X(n)= \frac{X(n)}{2} + \frac{X(n)}{2} + \frac{X(-n)}{2} - \frac{X(-n)}{2}$$

Rearranging the above terms we have

$$X(n)= \frac{X(n) + X(-n)}{2} + \frac{X(n) - X(-n)}{2}$$

Thus $X(n)= X_e(n) + X_o(n)$

Even component of discrete time signal is given by

$$X_e(n) = \frac{X(n) + X(-n)}{2}$$

Odd component of discrete time signal is given by

$$X_o(n) = \frac{X(n) - X(-n)}{2}$$

Test whether the following CT waveforms is periodic or not. If periodic find out the fundamental period.

a) 2 sin(2/3)t + 4 cos (1/2)t + 5 cos((1/3)t Ans: Period of x(t)= 12

b) a cos(t √2) + b sin(t/4) Ans: Non-Periodic

a) Find out the even and odd parts of the discrete signal $x(n)=\{2,\underline{4},3,2,1\}$

b) Find out the even and odd parts of the discrete signal $x(n)=\{\underline{2},2,2,2\}$

1.2.9 Energy signal and Power signal

Discrete time signals are also classified as finite energy or finite average power signals. The energy of a discrete time signal x(n) is given by

$$E= \sum_{n=-\infty}^{\infty} |x^2 (n)|$$

The average power for a discrete time signal x(n) is defined as

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

If Energy is finite and power is zero for x(n) then x(n) is an energy signal. If power is finite and energy is infinite then x(n) is power signal. There are some signals which are neither energy nor a power signal.

a) Find the power and energy of u(n) unit step function.

b) Find the power and energy of r(n) unit ramp function.

c) Find the power and energy of a^n u(n).

1.3 DISCRETE TIME SIGNALS AND SYSTEM

There are three ways to represent discrete time signals.

1) Functional Representation

$$x(n)= \begin{cases} 4 & \text{for } n=1,3 \\ -2 & \text{for } n=2 \\ 0 & \text{elsewhere} \end{cases}$$

2) Tabular method of representation

n	-3	-2	-1	0	1	2	3	4	5
x(n)	0	0	0	0	4	-2	4	0	0

3) Sequence Representation

$$X(n) = \{ \underset{\substack{\uparrow \\ n=0}}{0} , 4 , -2 , 4 , 0 ,\ldots\ldots\}$$

1.3.1STANDARD SIGNAL SEQUENCES

1) **Unit sample signal (Unit impulse signal)**

$$\delta (n) = \begin{cases} 1 & n=0 \\ 0 & n\neq 0 \end{cases} \qquad \text{i.e } \delta(n)=\{1\}$$

2) **Unit step signal**

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

3) **Unit ramp signal**

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

4) **Exponential signal**

$$x(n) = a^n = (re^{j\O})^n = r^n \; e^{j\O n} = r^n (\cos \O n + j \sin \O n)$$

5) **Sinusoidal waveform**

$$x(n) = A \sin wn$$

1.3.2 PROPERTIES OF DISCRETE TIME SIGNALS

1) **Shifting** : signal x(n) can be shifted in time. We can delay the sequence or advance the sequence. This is done by replacing integer n by n-k where k is integer. If k is positive signal is delayed in time by k samples (Arrow get shifted on left hand side) And if k is negative signal is advanced in time k samples (Arrow get shifted on right hand side)

$$X(n) = \{ 1, -1 , 0 , 4 , \underset{\underset{n=0}{\uparrow}}{-2} , 4 , 0 ,......\}$$

Delayed by 2 samples : $X(n-2)= \{ 1, -1 , \underset{\underset{n=0}{\uparrow}}{0} , 4 , -2 , 4 , 0 ,......\}$

Advanced by 2 samples : $X(n+2) = \{ 1, -1 , 0 , 4 , -2 , 4 , \underset{\underset{n=0}{\uparrow}}{0} ,......\}$

2) **Folding / Reflection** : It is folding of signal about time origin n=0. In this case replace n by –n.

Original signal:

$$X(n) = \{ 1, -1 , 0 , 4 , \underset{\underset{n=0}{\uparrow}}{-2} , 4 , 0\}$$

Folded signal:

$$X(-n) = \{ 0 , 4 , -2 , 4 , \underset{\underset{n=0}{\uparrow}}{0} , -1 , 1\}$$

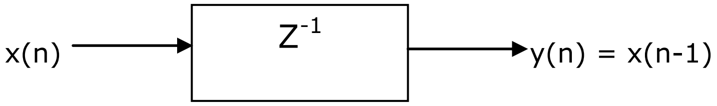
3) **Addition** : Given signals are x1(n) and x2(n), which produces output y(n) where y(n) = x1(n)+ x2(n). Adder generates the output sequence which is the sum of input sequences.

4) **Scaling**: Amplitude scaling can be done by multiplying signal with some constant. Suppose original signal is x(n). Then output signal is A x(n)

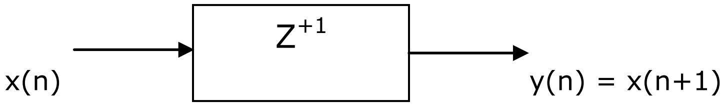
4) **Multiplication** : The product of two signals is defined as y(n) = x1(n) * x2(n).

1.3.3 SYMBOLS USED IN DISCRETE TIME SYSTEM

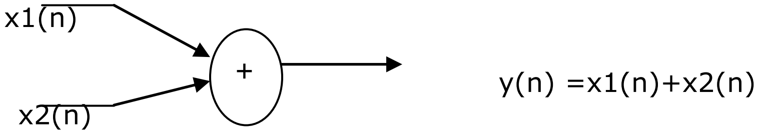
1. Unit delay



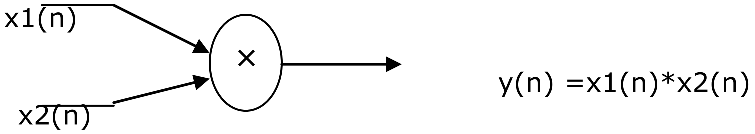
2. Unit advance



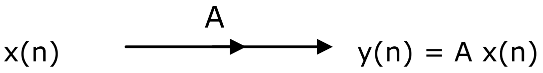
3. Addition



4. Multiplication



5. Scaling (constant multiplier)



1.3.4 CLASSIFICATION OF DISCRETE TIME SYSTEMS

1) STATIC v/s DYNAMIC

Sr No	**STATIC**	**DYNAMIC (Dynamicity property)**
1	Static systems are those systems whose output at any instance of time depends at most on input sample at same time.	Dynamic systems output depends upon past or future samples of input.
2	Static systems are memory less systems.	They have memories for memorize all samples.

It is very easy to find out that given system is static or dynamic. Just check that output of the system solely depends upon present input only, not dependent upon past or future.

Sr No	System [y(n)]	Static / Dynamic
1	x(n)	Static
2	A(n-2)	Dynamic
3	$X^2(n)$	Static
4	$X(n^2)$	Dynamic
5	$n\ x(n) + x^2(n)$	Static
6	X(n)+ x(n-2) +x(n+2)	Dynamic

2) TIME INVARIANT v/s TIME VARIANT SYSTEMS

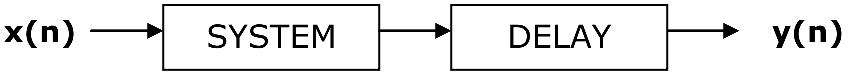
Sr No	TIME INVARIANT (TIV) / SHIFT INVARIANT	TIME VARIANT SYSTEMS / SHIFT VARIANT SYSTEMS (Shift Invariance property)
1	A System is time invariant if its input output characteristic do not change with shift of time.	A System is time variant if its input output characteristic changes with time.
2	Linear TIV systems can be uniquely characterized by Impulse response, frequency response or transfer function.	No Mathematical analysis can be performed.
3	a. Thermal Noise in Electronic components b. Printing documents by a printer	a. Rainfall per month b. Noise Effect

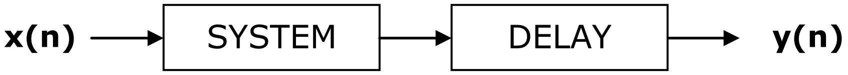
It is very easy to find out that given system is Shift Invariant or Shift Variant.
Suppose if the system produces output y(n) by taking input x(n)

x(n) → y(n)

If we delay same input by k units x(n-k) and apply it to same systems, the system produces output y(n-k)

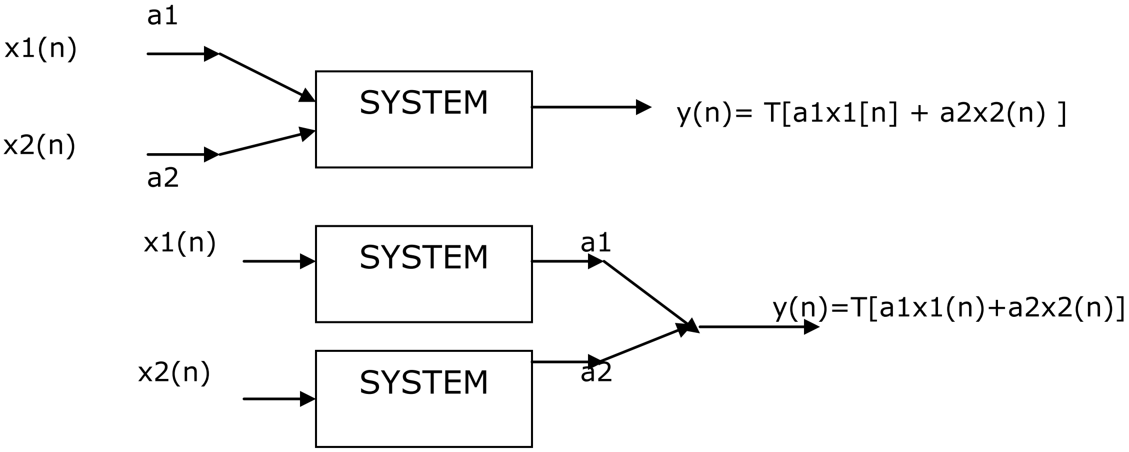
x(n-k) → y(n-k)





3) LINEAR v/s NON-LINEAR SYSTEMS

Sr No	LINEAR	NON-LINEAR (Linearity Property)
1	A System is linear if it satisfies superposition theorem.	A System is Non-linear if it does not satisfies superposition theorem.
2	Let x1(n) and x2(n) are two input sequences, then the system is said to be linear if and only if T[a1x1(n) + a2x2(n)]=a1T[x1(n)]+a2T[x2(n)]	



hence T [a1 x1(n) + a2 x2(n)] = T [a1 x1(n)] + T [a2 x2(n)]

It is very easy to find out that given system is Linear or Non-Linear.

Response to the system to the sum of signal = sum of individual responses of the system.

Sr No	**System y(n)**	**Linear or Non-Linear**
1	$e^{x(n)}$	Non-Linear
2	x^2 (n)	Non-Linear
3	m x(n) + c	Non-Linear
4	cos [x(n)]	Non-Linear
5	X(-n)	Linear
6	Log $_{10}$ (|x(n)|)	Non-Linear

4) CAUSAL v/s NON CAUSAL SYSTEMS

Sr No	**CAUSAL**	**NON-CAUSAL (Causality Property)**
1	A System is causal if output of system at any time depends only past and present inputs.	A System is Non causal if output of system at any time depends on future inputs.
2	In Causal systems the output is the function of x(n), x(n-1), x(n-2)….. and so on.	In Non-Causal System the output is the function of future inputs also. X(n+1) x(n+2) .. and so on
3	Example Real time DSP Systems	Offline Systems

It is very easy to find out that given system is causal or non-causal. Just check that output of the system depends upon present or past inputs only, not dependent upon future.

Sr No	**System [y(n)]**	**Causal /Non-Causal**
1	x(n) + x(n-3)	Causal
2	X(n)	Causal
3	X(n) + x(n+3)	Non-Causal
4	2 x(n)	Causal
5	X(2n)	Non-Causal
6	X(n)+ x(n-2) +x(n+2)	Non-Causal

5) STABLE v/s UNSTABLE SYSTEMS

Sr No	STABLE	UNSTABLE (Stability Property)
1	A System is BIBO stable if every bounded input produces a bounded output.	A System is unstable if any bounded input produces a unbounded output.
2	The input x(n) is said to bounded if there exists some finite number M_x such that $\lvert x(n)\rvert \le M_x < \infty$ The output y(n) is said to bounded if there exists some finite number M_y such that $\lvert y(n)\rvert \le M_y < \infty$	

STABILITY FOR LTI SYSTEM

It is very easy to find out that given system is stable or unstable. Just check that by providing input signal check that output should not rise to ∞.

The condition for stability is given by

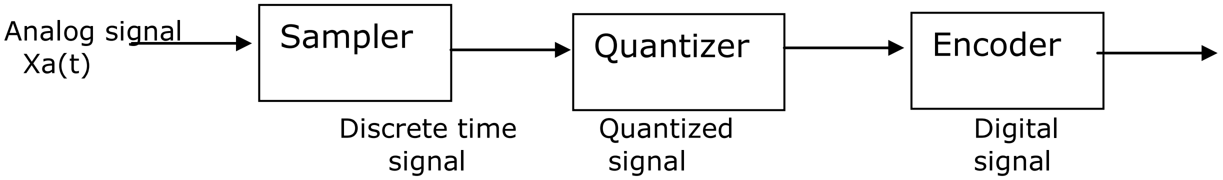
$$\sum_{k=-\infty}^{\infty} | h(k) | < \infty$$

Sr No	System [y(n)]	Stable / Unstable
1	Cos [x(n)]	Stable
2	x(-n+2)	Stable
3	|x(n)|	Stable
4	x(n) u(n)	Stable
5	X(n) + n x(n+1)	Unstable

1.4 ANALYSIS OF DISCRETE LINEAR TIME INVARIANT (LTI/LSI) SYSTEM

1.6 A/D CONVERSION

BASIC BLOCK DIAGRAM OF A/D CONVERTER



SAMPLING THEOREM

It is the process of converting continuous time signal into a discrete time signal by taking samples of the continuous time signal at discrete time instants.

$$X[n]= Xa(t) \text{ where } t= nTs = n/Fs \qquad ….(1)$$

When sampling at a rate of fs samples/sec, if k is any positive or negative integer, we cannot distinguish between the samples values of fa Hz and a sine wave of (fa+ kfs) Hz. Thus (fa + kfs) wave is alias or image of fa wave.

Thus **Sampling Theorem** states that if the highest frequency in an analog signal is Fmax and the signal is sampled at the rate fs > 2Fmax then x(t) can be exactly recovered from its sample values. This sampling rate is called Nyquist rate of sampling. The imaging or aliasing starts after Fs/2 hence folding frequency is fs/2. If the frequency is less than or equal to 1/2 it will be represented properly.

Example:

Case 1: X1(t) = cos 2∏ (10) t Fs= 40 Hz i.e t= n/Fs
x1[n]= cos 2∏(n/4)= cos (∏/2)n

Case 2: X1(t) = cos 2∏ (50) t Fs= 40 Hz i.e t= n/Fs
x1[n]= cos 2∏(5n/4)= cos 2∏(1+ ¼)n
= cos (∏/2)n

Thus the frequency 50 Hz, 90 Hz , 130 Hz … are alias of the frequency 10 Hz at the sampling rate of 40 samples/sec

QUANTIZATION

The process of converting a discrete time continuous amplitude signal into a digital signal by expressing each sample value as a finite number of digits is called quantization. The error introduced in representing the continuous values signal by a finite set of discrete value levels is called quantization error or quantization noise.

Example: $x[n] = 5(0.9)^n u(n)$ where 0 <n < ∞ & fs= 1 Hz

N	[n]	X_q [n] Rounding	X_q [n] Truncating	e_q [n]
0	5	5.0	5.0	0
1	4.5	4.5	4.5	0
2	4.05	4.0	4.0	-0.05
3	3.645	3.6	3.6	-0.045
4	3.2805	3.2	3.3	0.0195

Quantization Step/Resolution : The difference between the two quantization levels is called quantization step. It is given by $\Delta = X_{Max} - x_{Min} / L\text{-}1$ where L indicates Number of quantization levels.

CODING/ENCODING

Each quantization level is assigned a unique binary code. In the encoding operation, the quantization sample value is converted to the binary equivalent of that quantization level. If 16 quantization levels are present, 4 bits are required. Thus bits required in the coder is the smallest integer greater than or equal to Log_2 L. i.e b= Log_2 L
Thus Sampling frequency is calculated as fs=Bit rate / b.

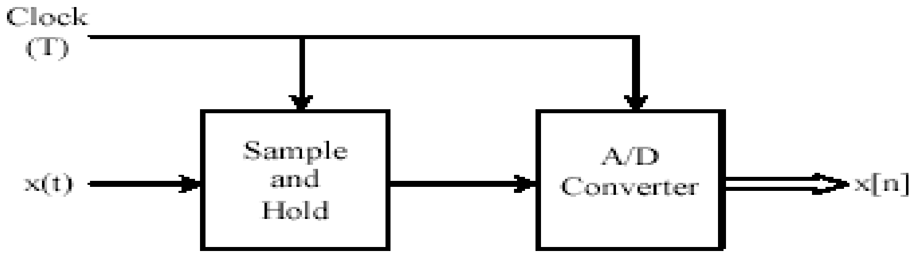
ANTI-ALIASING FILTER

When processing the analog signal using DSP system, it is sampled at some rate depending upon the bandwidth. For example if speech signal is to be processed the frequencies upon 3khz can be used. Hence the sampling rate of 6khz can be used. But the speech signal also contains some frequency components more than 3khz. Hence a sampling rate of 6khz will introduce aliasing. Hence signal should be band limited to avoid aliasing.

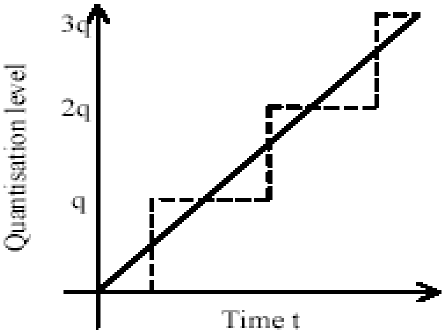
The signal can be band limited by passing it through a filter (LPF) which blocks or attenuates all the frequency components outside the specific bandwidth. Hence called as Anti aliasing filter or pre-filter. (Block Diagram)

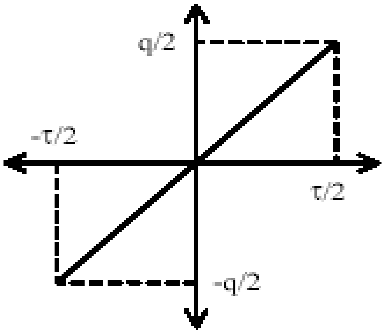
SAMPLE-AND-HOLD CIRCUIT:

The sampling of an analogue continuous-time signal is normally implemented using a device called an analogue-to- digital converter (A/D). The continuous-time signal is first passed through a device called a sample-and-hold (S/H) whose function is to measure the input signal value at the clock instant and hold it fixed for a time interval long enough for the A/D operation to complete. Analogue-to-digital conversion is potentially a slow operation, and a variation of the input voltage during the conversion may disrupt the operation of the converter. The S/H prevents such disruption by keeping the input voltage constant during the conversion. This is schematically illustrated by Figure.



After a continuous-time signal has been through the A/D converter, the quantized output may differ from the input value. The maximum possible output value after the quantization process could be up to half the quantization level q above or q below the ideal output value. This deviation from the ideal output value is called the quantization error. In order to reduce this effect, we increases the number of bits.





Q) Calculate Nyquist Rate for the analog signal x(t)

1) x(t)= 4 cos 50 ∏t + 8 sin 300∏t –cos 100∏t Fn=300 Hz
2) x(t)= 2 cos 2000∏t+ 3 sin 6000∏t + 8 cos 12000∏t Fn=12KHz
3) x(t)= 4 cos 100∏t Fn=100 Hz

Q) The following four analog sinusoidal are sampled with the fs=40Hz. Find out corresponding time signals and comment on them

X1(t)= cos 2∏(10)t
X2(t)= cos 2∏(50)t

$X3(t) = \cos 2\prod(90)t$
$X4(t) = \cos 2\prod(130)t$

Q) Signal $x1(t)=10\cos2\prod(1000)t+ 5 \cos2\prod(5000)t$. Determine Nyquist rate. If the signal is sampled at 4khz will the signal be recovered from its samples.

Q) Signal $x1(t)=3 \cos 600\prod t+ 2\cos800\prod t$. The link is operated at 10000 bits/sec and each input sample is quantized into 1024 different levels. Determine Nyquist rate, sampling frequency, folding frequency & resolution.

DIFFERENCE BETWEEN FIR AND IIR

Sr No	**Finite Impulse Response (FIR)**	**Infinite Impulse Response (IIR)**
1	FIR has an impulse response that is zero outside of some finite time interval.	IIR has an impulse response on infinite time interval.
2	Convolution formula changes to $y(n) = \sum_{n=-M}^{M} x(k)\ h(n-k)$ For causal FIR systems limits changes to 0 to M.	Convolution formula changes to $y(n) = \sum_{n=-\infty}^{\infty} x(k)\ h(n-k)$ For causal IIR systems limits changes to 0 to ∞.
3	The FIR system has limited span which views only most recent M input signal samples forming output called as "Windowing".	The IIR system has unlimited span.
4	FIR has limited or finite memory requirements.	IIR System requires infinite memory.
5	Realization of FIR system is generally based on Convolution Sum Method.	Realization of IIR system is generally based on Difference Method.

Discrete time systems has one more type of classification.

1. Recursive Systems
2. Non-Recursive Systems

Sr No	**Recursive Systems**	**Non-Recursive systems**
1	In Recursive systems, the output depends upon past, present, future value of inputs as well as past output.	In Non-Recursive systems, the output depends only upon past, present or future values of inputs.
2	Recursive Systems has feedback from output to input.	No Feedback.
3	Examples $y(n) = x(n) + y(n-2)$	$Y(n) = x(n) + x(n-1)$

1.5 ANALYSIS OF LTI SYSTEM

1.5.1 Z TRANFORM

INTRODUCTION TO Z TRANSFORM

For analysis of continuous time LTI system Laplace transform is used. And for analysis of discrete time LTI system z transform is used. Z transform is mathematical tool used for conversion of time domain into frequency domain (z domain) and is a function of the complex valued variable Z. The z transform of a discrete time signal x(n) denoted by X(z) and given as

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)\, z^{-n} \qquad \text{z-Transform.......(1)}$$

Z transform is an infinite power series because summation index varies from -∞ to ∞. But it is useful for values of z for which sum is finite. The values of z for which f (z) is finite and lie within the region called as "region of convergence (ROC).

ADVANTAGES OF Z TRANSFORM

1. The DFT can be determined by evaluating z transform.
2. Z transform is widely used for analysis and synthesis of digital filter.
3. Z transform is used for linear filtering. z transform is also used for finding Linear convolution, cross-correlation and auto-correlations of sequences.
4. In z transform user can characterize LTI system (stable/unstable, causal/anti-causal) and its response to various signals by placements of pole and zero plot.

ADVANTAGES OF ROC(REGION OF CONVERGENCE)

1. ROC is going to decide whether system is stable or unstable.
2. ROC decides the type of sequences causal or anti-causal.
3. ROC also decides finite or infinite duration sequences.

Z TRANSFORM PLOT

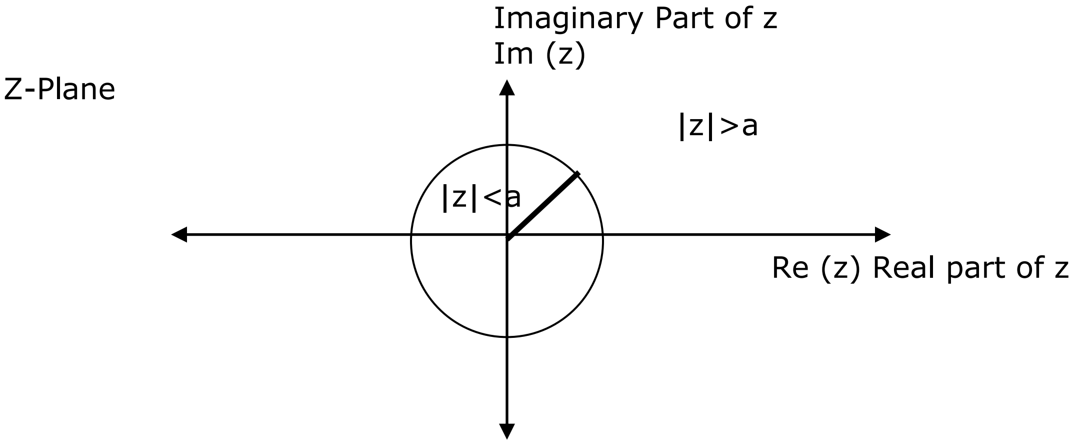


Fig show the plot of z transforms. The z transform has real and imaginary parts. Thus a plot of imaginary part versus real part is called complex z-plane. The radius of circle is 1 called as unit circle. This complex z plane is used to show ROC, poles and zeros. Complex variable z is also expressed in polar form as $Z= re^{j\omega}$ where r is radius of circle is given by |z| and ω is the frequency of the sequence in radians and given by $\angle z$.

Sr No	**Time Domain Sequence**	**Property**	**z Transform**	**ROC**
1	δ(n) (Unit sample)		1	complete z plane
2	δ(n-k)	Time shifting	z^{-k}	except z=0
3	δ(n+k)	Time shifting	z^{k}	except z=∞
4	u(n) (Unit step)		$1/1- z^{-1} = z/z-1$	$\|z\| > 1$
5	u(-n)	Time reversal	$1/1- z$	$\|z\| < 1$
6	-u(-n-1)	Time reversal	$z/z- 1$	$\|z\| < 1$
7	n u(n) (Unit ramp)	Differentiation	$z^{-1} / (1- z^{-1})^2$	$\|z\| > 1$
8	a^n u(n)	Scaling	$1/1- (az^{-1})$	$\|z\| > \|a\|$
9	$-a^n$ u(-n-1)(Left side exponential sequence)		$1/1- (az^{-1})$	$\|z\| < \|a\|$
10	n a^n u(n)	Differentiation	$a z^{-1} / (1- az^{-}1)^2$	$\|z\| > \|a\|$
11	-n a^n u(-n-1)	Differentiation	$a z^{-1} / (1- az^{-}1)^2$	$\|z\| < \|a\|$
12	a^n for 0 < n < N-1		$1- (a z^{-1})^N/ 1- az^{-}1$	$\|az^{-}1\| < \infty$ except z=0
13	1 for 0<n<N-1 or u(n) – u(n-N)	Linearity Shifting	$1- z^{-N}/ 1- z^{-}1$	$\|z\| > 1$
14	cos(ω_0n) u(n)		$\frac{1- z^{-1}\cos\omega_0}{1- 2z^{-1}\cos\omega_0+z^{-2}}$	$\|z\| > 1$
15	sin(ω_0n) u(n)		$\frac{z^{-1}\sin\omega_0}{1- 2z^{-1}\cos\omega_0+z^{-2}}$	$\|z\| > 1$
16	a^n cos(ω_0n) u(n)	Time scaling	$\frac{1- (z/a)^{-1}\cos\omega_0}{1- 2(z/a)^{-1}\cos\omega_0+(z/a)^{-2}}$	$\|z\| > \|a\|$
17	a^n sin(ω_0n) u(n)	Time scaling	$\frac{(z/a)^{-1}\sin\omega_0}{1- 2(z/a)^{-1}\cos\omega_0+(z/a)^{-2}}$	$\|z\| > \|a\|$

Q) Determine z transform of following signals. Also draw ROC.
i) x(n) = {$\underline{1}$,2,3,4,5}
ii) x(n)={1,2,3,$\underline{4}$,5,0,7}
Q) Determine z transform and ROC for x(n) = $(-1/3)^n$ u(n) –$(1/2)^n$ u(-n-1).
Q) Determine z transform and ROC for x(n) = [3.(4^n)–4(2^n)] u(n).
Q) Determine z transform and ROC for x(n) = $(1/2)^n$ u(-n).
Q) Determine z transform and ROC for x(n) = $(1/2)^n$ {u(n) – u(n-10)}.
Q) Find linear convolution using z transform. X(n)={1,2,3} & h(n)={1,2}

PROPERTIES OF Z TRANSFORM (ZT)

1) Linearity

The linearity property states that if

$$x1(n) \xleftrightarrow{z} X1(z) \text{ And}$$

$$x2(n) \xleftrightarrow{z} X2(z) \text{ Then}$$

Then

$$a1\ x1(n) + a2\ x2(n) \overset{z}{\longleftrightarrow} a1\ X1(z) + a2\ X2(z)$$

z Transform of linear combination of two or more signals is equal to the same linear combination of z transform of individual signals.

2) Time shifting

The Time shifting property states that if

$$x(n) \overset{z}{\longleftrightarrow} X(z) \text{ And}$$

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

Thus shifting the sequence circularly by ‘k’ samples is equivalent to multiplying its z transform by z^{-k}

3) Scaling in z domain

This property states that if

$$x(n) \overset{z}{\longleftrightarrow} X(z) \text{ And}$$

Then $$a^{n}\ x(n) \overset{z}{\longleftrightarrow} x(z/a)$$

Thus scaling in z transform is equivalent to multiplying by a^{n} in time domain.

4) Time reversal Property

The Time reversal property states that if

$$x(n) \overset{z}{\longleftrightarrow} X(z) \text{ And}$$

Then $$x(-n) \overset{z}{\longleftrightarrow} x(z^{-1})$$

It means that if the sequence is folded it is equivalent to replacing z by z^{-1} in z domain.

5) Differentiation in z domain

The Differentiation property states that if

$$x(n) \overset{z}{\longleftrightarrow} X(z) \text{ And}$$

Then $$n\ x(n) \overset{z}{\longleftrightarrow} -z\ d/dz\ (X(z))$$

6) Convolution Theorem

The Circular property states that if

$$x1(n) \overset{z}{\longleftrightarrow} X1(z) \text{ And}$$

$$x2(n) \overset{z}{\longleftrightarrow} X2(z) \text{ Then}$$

Then $$x1(n)\ *\ x2(n) \underset{N}{\overset{z}{\longleftrightarrow}} X1(z)\ X2(z)$$

Convolution of two sequences in time domain corresponds to multiplication of its Z transform sequence in frequency domain.

7) Correlation Property

The Correlation of two sequences states that if

$$x1(n) \overset{z}{\longleftrightarrow} X1(z) \text{ And}$$

$$x2(n) \overset{z}{\longleftrightarrow} X2(z) \text{ Then}$$

then $$\sum_{n=-\infty}^{\infty} x1\,(l)\; x2(-l) \overset{z}{\longleftrightarrow} X1(z)\, x2(z^{-1})$$

8) Initial value Theorem

Initial value theorem states that if

$$x(n) \overset{z}{\longleftrightarrow} X(z) \text{ And}$$

then

$$x(0) = \lim_{z \to \infty} X(Z)$$

9) Final value Theorem

Final value theorem states that if

$$x(n) \overset{z}{\longleftrightarrow} X(z) \text{ And}$$

then

$$\lim_{z \to \infty} x(n) = \lim_{z \to 1} (z-1)\, X(z)$$

RELATIONSHIP BETWEEN FOURIER TRANSFORM AND Z TRANSFORM.

There is a close relationship between Z transform and Fourier transform. If we replace the complex variable z by $e^{-j\omega}$, then z transform is reduced to Fourier transform.

Z transform of sequence x(n) is given by

$$X(z) = \sum_{n=-\infty}^{\infty} x\,(n)\, z^{-n} \qquad \textbf{(Definition of z-Transform)}$$

Fourier transform of sequence x(n) is given by

$$X(\omega) = \sum_{n=-\infty}^{\infty} x\,(n)\, e^{-j\omega n} \qquad \textbf{(Definition of Fourier Transform)}$$

Complex variable z is expressed in polar form as $Z= re^{j\omega}$ where r= |z| and ω is $\angle z$. Thus we can be written as

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

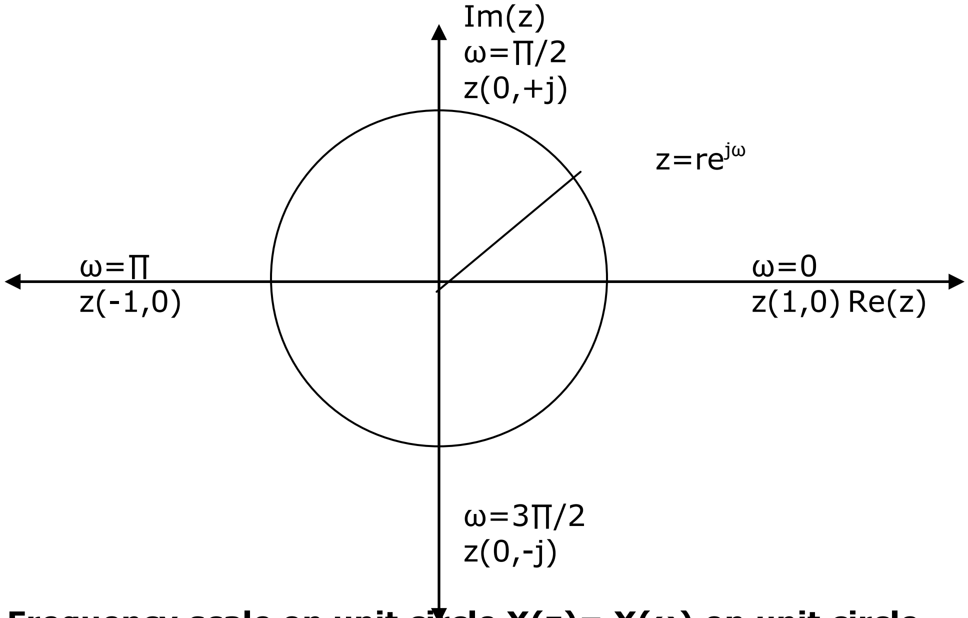
$$X(z)\Big|_{z=e^{jw}} = \sum_{n=-\infty}^{\infty} x\,(n)\, e^{-j\omega n}$$

$$X(z)\Big|_{z=e^{jw}} = x(\omega) \qquad \text{at } |z| = \text{unit circle.}$$

Thus, X(z) can be interpreted as Fourier Transform of signal sequence ($x(n)\ r^{-n}$). Here r^{-n} grows with n if r<1 and decays with n if r>1. X(z) converges for |r|= 1. hence Fourier transform may be viewed as Z transform of the sequence evaluated on unit circle. Thus The relationship between DFT and Z transform is given by

$$X(z)\Big|_{z=e^{j2\Pi kn}} = x(k)$$

The frequency ω=0 is along the positive Re(z) axis and the frequency ∏/2 is along the positive Im(z) axis. Frequency ∏ is along the negative Re(z) axis and 3∏/2 is along the negative Im(z) axis.



Frequency scale on unit circle X(z)= X(ω) on unit circle

INVERSE Z TRANSFORM (IZT)

The signal can be converted from time domain into z domain with the help of z transform (ZT). Similar way the signal can be converted from z domain to time domain with the help of inverse z transform(IZT). The inverse z transform can be obtained by using two different methods.

1) Partial fraction expansion Method (PFE) / Application of residue theorem
2) Power series expansion Method (PSE)

1. PARTIAL FRACTION EXPANSION METHOD

In this method X(z) is first expanded into sum of simple partial fraction.

$$X(z) = \frac{a_0 z^m + a_1 z^{m-1} + \ldots\ldots. + a_m}{b_0 z^n + b_1 zn^{n-1} + \ldots\ldots. + b_n} \qquad \text{for } m \le n$$

First find the roots of the denominator polynomial

$$X(z) = \frac{a_0 z^m + a_1 z^{m-1} + \ldots\ldots. + a_m}{(z - p_1)\,(z - p_2)\ldots\ldots\,(z - p_n)}$$

The above equation can be written in partial fraction expansion form and find the coefficient A_K and take IZT.

SOLVE USING PARTIAL FRACTION EXPANSION METHOD (PFE)

Sr No	Function (ZT)	Time domain sequence	Comment
1	$\frac{1}{1- a\, z^{-1}}$	$a^n\, u(n)$ for $\lvert z\rvert > a$	causal sequence
		$-a^n\, u(-n-1)$ for $\lvert z\rvert < a$	anti-causal sequence
2	$\frac{1}{1+z^{-1}}$	$(-1)^n\, u(n)$ for $\lvert z\rvert > 1$	causal sequence
		$-(-1)^n\, u(-n-1)$ for $\lvert z\rvert < a$	anti-causal sequence
3	$\frac{3-4z^{-1}}{1- 3.5\, z^{-1}+1.5z^{-2}}$	$-2(3)^n u(-n-1) + (0.5)^n u(n)$ for $0.5<\lvert z\rvert<3$	stable system
		$2(3)^n u(n) + (0.5)^n u(n)$ for $\lvert z\rvert>3$	causal system
		$-2(3)^n u(-n-1) - (0.5)^n u(-n-1)$ for $\lvert z\rvert<0.5$	anti-causal system
4	$\frac{1}{1- 1.5\, z^{-1}+0.5z^{-2}}$	$-2(1)^n u(-n-1) + (0.5)^n u(n)$ for $0.5<\lvert z\rvert<1$	stable system
		$2(1)^n u(n) + (0.5)^n u(n)$ for $\lvert z\rvert>1$	causal system
		$-2(1)^n u(-n-1) - (0.5)^n u(-n-1)$ for $\lvert z\rvert<0.5$	anti-causal system
5	$\frac{1+2\, z^{-1}+ z^{-2}}{1- 3/2\, z^{-1}+0.5z^{-2}}$	$2\delta(n)+8(1)^n u(n)- 9(0.5)^n u(n)$ for $\lvert z\rvert>1$	causal system
6	$\frac{1+ z^{-1}}{1- z^{-1} + 0.5z^{-2}}$	$(1/2-j3/2)\, (1/2+j1/2)^n u(n)+$ $(1/2+j3/2)\, (1/2+j1/2)^n u(n)$	causal system
7	$\frac{1 -(0.5)\, z^{-1}}{1-3/4\, z^{-1}+1/8\, z^{-2}}$	$4(-1/2)^n u(n) - 3\, (-1/4)^n u(n)$ for $\lvert z\rvert>1/2$	causal system
8	$\frac{1- 1/2\, z^{-1}}{1- 1/4\, z^{-2}}$	$(-1/2)^n u(n)$ for $\lvert z\rvert>1/2$	causal system
9	$\frac{z + 1}{3z^2 - 4z + 1}$	$\delta(n)+ u(n) - 2(1/3)^n u(n)$ for $\lvert z\rvert>1$	causal system
10	$\frac{5z}{(z-1)\, (z-2)}$	$5(2^n-1)$ for $\lvert z\rvert>2$	causal system
11	$\frac{z^3}{(z-1)\, (z-1/2)^2}$	$4-(n+3)(1/2)^n$ for $\lvert z\rvert>1$	causal system

2. RESIDUE THEOREM METHOD

In this method, first find $G(z)= z^{n-1}\, X(Z)$ and find the residue of $G(z)$ at various poles of $X(z)$.

SOLVE USING “RESIDUE THEOREM“ METHOD

Sr No	Function (ZT)	Time domain Sequence
1	$\frac{z}{z-a}$	For causal sequence $(a)^n u(n)$
2	$\frac{z}{(z-1)(z-2)}$	$(2^n -1)\ u(n)$
3	$\frac{z^2+z}{(z-1)^2}$	$(2n+1)\ u(n)$
4	$\frac{z^3}{(z-1)(z-0.5)^2}$	$4 - (n+3)(0.5)^n u(n)$

3. POWER-SERIES EXPANSION METHOD

The z transform of a discrete time signal x(n) is given as

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)\, z^{-n} \qquad (1)$$

Expanding the above terms we have

$$x(z) = \ldots..+x(-2)Z^2+ x(-1)Z+ x(0)+ x(1)\, Z^{-1} + x(2)\, Z^2 +\ldots.. \qquad (2)$$

This is the expansion of z transform in power series form. Thus sequence x(n) is given as $x(n) =\{ \ldots.. ,x(-2),x(-1),x(0),x(1),x(2),\ldots\ldots\ldots..\}$.

Power series can be obtained directly or by long division method.

SOLVE USING “POWER SERIES EXPANSION“ METHOD

Sr No	Function (ZT)	Time domain Sequence
1	$\frac{z}{z-a}$	For causal sequence $a^n u(n)$ For Anti-causal sequence $-a^n u(-n-1)$
2	$\frac{1}{1- 1.5 z^{-1}+0.5z^{-2}}$	$\{\underline{1},3/2,7/4,15,8,\ldots\ldots.\}$ For $\lvert z\rvert > 1$ $\{\ldots.14,6,2,0,\underline{0}\}$ For $\lvert z\rvert < 0.5$
3	$\frac{z^2+z}{z^3 -3z^2+3z -1}$	$\{0,1,4,9,\ldots..\}$ For $\lvert z\rvert > 3$
4	$z^2(1-0.5z^{-1})(1+z^{-1}\ \ -z^{-1})$	$X(n) =\{1,-0.5,\underline{-1},0.5\}$
5	$log(1+az^{-1})$	$(-1)^{n+1}a^n/n$ for $n\geq1$ and $\lvert z\rvert>\lvert a\rvert$

4. RECURSIVE ALGORITHM

The long division method can be recast in recursive form.

$$X(z) = \frac{a_0 + a_1 z^{-1}+ a_2 z^{-2}}{b_0 + b_1 z^{-1}+ b_2 z^{-2}}$$

Their IZT is give as

$$x(n) = 1/b0\ [\, a_n - \sum_{i=1}^{n} x(n-i)\, b_i] \qquad \text{for } n=1,2,\ldots\ldots\ldots\ldots.$$

Thus

$X(0) = a_0/b_0$

$X(1) = 1/b_0\ [\, a_1- x(0)\, b_1]$

$X(2) = 1/b_0\ [\, a_1- x(1)\, b_1 - x(0)\, b_2]$ ……………

SOLVE USING “RECURSIVE ALGORITHM“ METHOD

Sr No	Function (ZT)	Time domain Sequence
1	$\dfrac{1+2z^{-1}+z^{-2}}{1-z^{-1}+0.3561z^{2}}$	X(n) = {1,3,3.6439,….}
2	$\dfrac{1+z^{-1}}{1-5/6\ z^{-1}+\ 1/6\ z^{-2}}$	X(n) = {1,11/6,49/36,….}
3	$\dfrac{z^{4}+z^{2}}{z^{2}-3/4z+\ 1/8}$	X(n) = { 23/16,63/64,………}

Example 1:

Let us consider the z-transform of $a^n u(n)$*:*

$$\begin{aligned} ZT\left\{a^n u(n)\right\} &= \sum_{n=0}^{\infty} a^n z^{-n} \\ &= \sum_{n=0}^{\infty} \left(az^{-1}\right)^n \\ &= \frac{1}{1-az^{-1}} \quad \textit{if } \left|az^{-1}\right| < 1, \\ &= \frac{z}{z-a} \end{aligned}$$

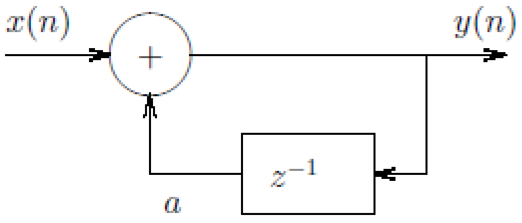
The region of convergence (ROC) *is the set of all values of* z *for which the z-transform converges. In this example, it is* $|z| > |a|$.

Example 2:Find the magnitude and phase plot of

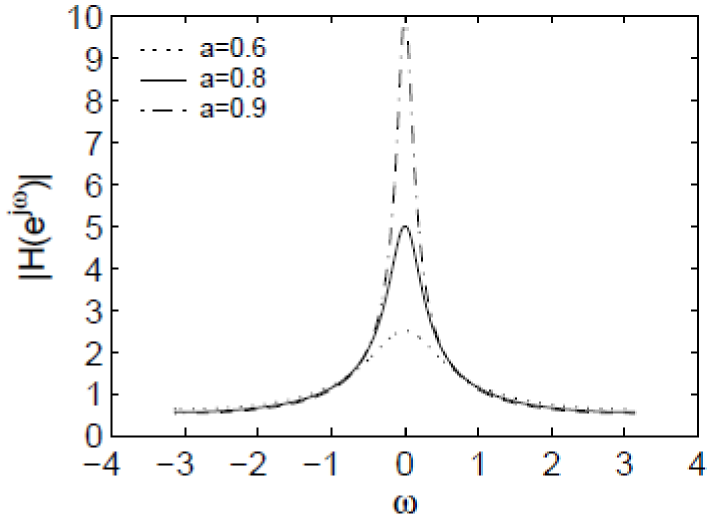
$$\begin{aligned} \frac{Y(z)}{X(z)} &= \frac{1}{1-az^{-1}} \\ Y(z) - az^{-1}Y(z) &= X(z) \\ Y(z) &= az^{-1}Y(z) + X(z) \end{aligned}$$

Inverting the z-transform, we have:

$$y(n) = ay(n-1) + x(n).$$



$$
\begin{aligned}
H\left(e^{j\omega}\right) &= \frac{1}{1-ae^{-j\omega}} \\
\left|H\left(e^{j\omega}\right)\right| &= \frac{1}{\sqrt{(1-ae^{-j\omega})(1-ae^{j\omega})}} \\
&= \frac{1}{\sqrt{1-2a\cos\omega+a^2}} \\
&= \frac{1}{\sqrt{(1-a)^2+2a(1-\cos\omega)}}
\end{aligned}
$$



Example 3:

Let $x(n) = 2^n u(n)$. *Then,*

$$X(z) = \sum_{n=0}^{\infty} 2^n z^{-n} = \sum_{n=0}^{\infty} \left(\frac{2}{z}\right)^n .$$

If $|z| \leq 2$, *then* $\left|\frac{2}{z}\right| \geq 1$. *This means that every term in the series has an absolute value greater than or equal to 1. If it is greater than 1, then every successive terms grows larger. If it is equal to 1, then we are just adding 1 an infinite number of times. In both cases, the series diverges.*

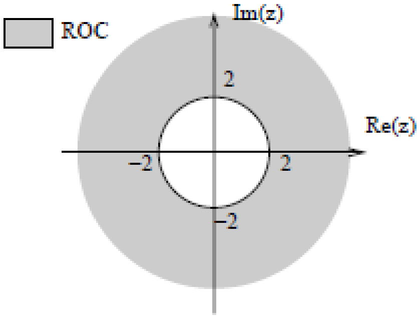
On the other hand, if $|z| > 2$, *then the geometric series converges because* $\left|\frac{2}{z}\right| < 1$, *and we have*

1

$$X(z) = \frac{1}{1-\frac{2}{z}} \ .$$

Thus, the z-transform of $x(n) = 2^n u(n)$ *is*

$$X(z) = \begin{cases} \frac{1}{1-\frac{2}{z}} & |z| > 2 \\ \\ undefined & |z| \leq 2 \end{cases}$$



Example 4:

Let $y(n) = -2^n u(-n-1)$. *Then*

$$\begin{aligned} Y(z) &= \sum_{n=-1}^{-\infty} -2^n z^{-n} \\ &= -\sum_{m=0}^{\infty} 2^{-m-1} z^{m+1} \qquad where\ m = -n-1 \\ &= -\frac{z}{2}\sum_{m=0}^{\infty}\left(\frac{z}{2}\right)^m . \end{aligned}$$

1. *If* $|z| \geq 2$, *the series diverges.*

2. *if* $|z| < 2$, *the series converges, and*

$$\begin{aligned} Y(z) &= -\frac{z}{2}\frac{1}{1-\frac{z}{2}} \\ &= -\frac{z}{2-z} \\ &= \frac{1}{1-\frac{2}{z}} . \end{aligned}$$

Putting it all together,

$$Y(z) = \begin{cases} undefined & |z| \geq 2 \\ \\ \frac{1}{1-\frac{2}{z}} & |z| < 2 \end{cases}$$

Example 5:Find the inverse Z Transform

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

$$\begin{aligned} [X(z)(1-z^{-1})]_{z=1} = \left[\frac{1-4z^{-1}}{1-2z^{-1}}\right]_{z=1} &= \left[A_1 + \frac{A_2(1-z^{-1})}{1-2z^{-1}}\right]_{z=1} \\ \Rightarrow 3 &= A_1. \end{aligned}$$

$$\begin{aligned} [X(z)(1-2z^{-1})]_{z=2} = \left[\frac{1-4z^{-1}}{1-1z^{-1}}\right]_{z=2} &= \left[A_1 + \frac{1-4z^{-1})}{1-z^{-1}}\right]_{z=2} \\ \Rightarrow -2 &= A_2. \end{aligned}$$

Thus, we have

$$X(z) = \frac{3}{1-z^{-1}} - \frac{2}{1-2z^{-1}} \quad .$$

We now consider the three possible ROC's that this z-transform can have.

Case 1. *ROC* $|z| > 2$. *Using (1.51),*

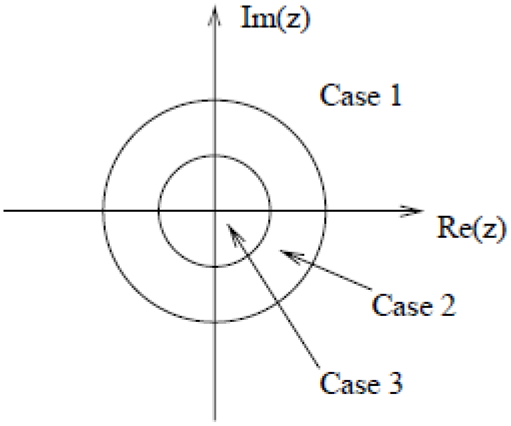
$$\begin{aligned} x(n) &= 3 \cdot 1^n u(n) - 2 \cdot 2^n u(n) \\ &= (3 - 2^{n+1})u(n). \end{aligned}$$

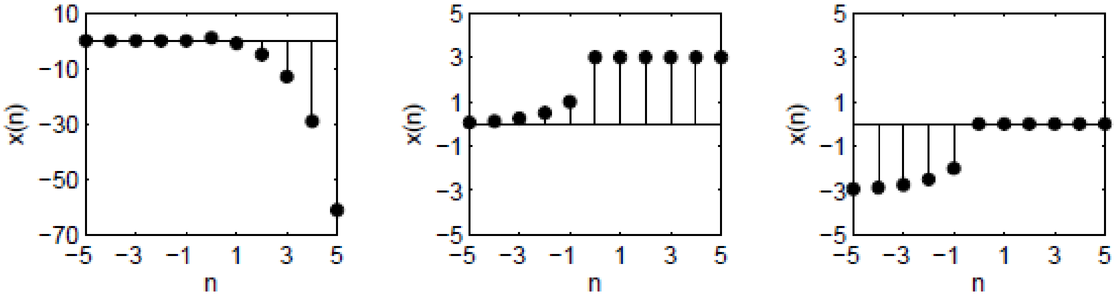
Case 2. *ROC* $1 < |z| < 2$. *Using (1.51) and (1.52),*

$$x(n) = 3 \cdot 1^n u(n) + 2 \cdot 2^n u(-n-1).$$

Case 3. *ROC* $|z| < 1$. *Using (1.52),*

$$x(n) = (-3 + 2^{n+1})u(-n-1).$$





(a) $|z| > 2$ (b) $1 < |z| < 2$ (c) $|z| < 1$

POLE –ZERO PLOT

1. X(z) is a rational function, that is a ratio of two polynomials in z^{-1} or z. The roots of the denominator or the value of z for which X(z) becomes infinite, defines locations of the poles. The roots of the numerator or the value of z for which X(z) becomes zero, defines locations of the zeros.

2. ROC dos not contain any poles of X(z). This is because x(z) becomes infinite at the locations of the poles. Only poles affect the causality and stability of the system.

3. **CASUALTY CRITERIA FOR LSI SYSTEM**
   LSI system is causal if and only if the ROC the system function is exterior to the circle. i. e $|z| > r$. This is the condition for causality of the LSI system in terms of z transform. (The condition for LSI system to be causal is $h(n) = 0$ ….. $n<0$)

4. **STABILITY CRITERIA FOR LSI SYSTEM**
   Bounded input x(n) produces bounded output y(n) in the LSI system only if

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty$$

With this condition satisfied, the system will be stable. The above equation states that the LSI system is stable if its unit sample response is absolutely summable. This is necessary and sufficient condition for the stability of LSI system.

$$H(z) = \sum_{n=-\infty}^{\infty} h(n)\, z^{-n} \qquad \text{Z-Transform.......(1)}$$

Taking magnitude of both the sides

$$|H(z)| = \left| \sum_{n=-\infty}^{\infty} h(n)\, z^{-n} \right| \qquad ......(2)$$

Magnitudes of overall sum is less than the sum of magnitudes of individual sums.

$$|H(z)| \leq \left| \sum_{n=-\infty}^{\infty} h(n)\ z^{-n} \right|$$

$$|H(z)| \leq \sum_{n=-\infty}^{\infty} |h(n)|\ |\ z^{-n}\ | \qquad \text{….(3)}$$

If H(z) is evaluated on the unit circle $|\ z^{-n}|=|z|=1$.
Hence LSI system is stable if and only if the ROC the system function includes the unit circle. i.e $r < 1$. This is the condition for stability of the LSI system in terms of z transform. Thus

For stable system $|z| < 1$
For unstable system $|z| > 1$
Marginally stable system $|z| = 1$

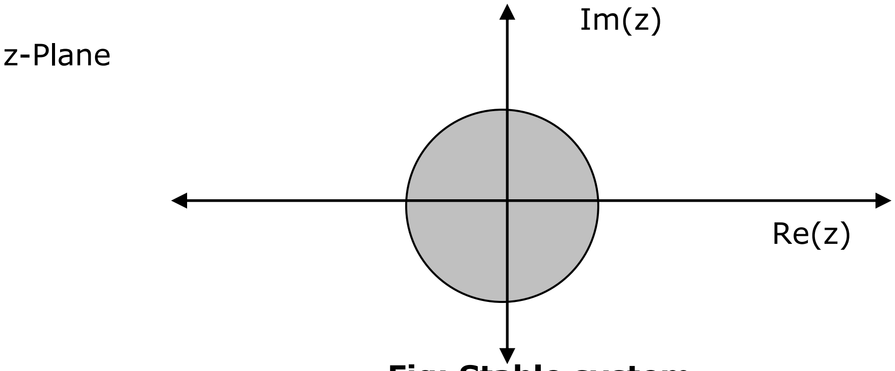


Fig: Stable system

Poles inside unit circle gives stable system. Poles outside unit circle gives unstable system. Poles on unit circle give marginally stable system.

6. A causal and stable system must have a system function that converges for $|z| > r < 1$.

STANDARD INVERSE Z TRANSFORMS

Sr No	Function (ZT)	Causal Sequence $\|z\| > \|a\|$	Anti-causal sequence $\|z\| <\|a\|$
1	$\frac{z}{z-a}$	$(a)^n\ u(n)$	$-(a)^n\ u(-n-1)$
2	$\frac{z}{z-1}$	u(n)	u(-n-1)
3	$\frac{z^2}{(z-a)^2}$	$(n+1)a^n$	$-(n+1)a^n$
4	$\frac{z^k}{(z-a)^k}$	$1/(k-1)!\ (n+1)\ (n+2)\ldots\ldots a^n$	$-1/(k-1)!\ (n+1)\ (n+2)\ldots\ldots\ldots a^n$
5	1	$\delta(n)$	$\delta(n)$
6	Z^k	$\delta(n+k)$	$\delta(n+k)$
7	Z^{-k}	$\delta(n-k)$	$\delta(n-k)$

ONE SIDED Z TRANSFORM

Sr No	z Transform (Bilateral)	One sided z Transform (Unilateral)
1	z transform is an infinite power series because summation index varies from ∞ to -∞. Thus Z transform are given by $X(z) = \sum_{n=-\infty}^{\infty} x(n)\, z^{-n}$	One sided z transform summation index varies from 0 to ∞. Thus One sided z transform are given by $X(z) = \sum_{n=0}^{\infty} x(n)\, z^{-n}$
2	z transform is applicable for relaxed systems (having zero initial condition).	One sided z transform is applicable for those systems which are described by differential equations with non zero initial conditions.
3	z transform is also applicable for non-causal systems.	One sided z transform is applicable for causal systems only.
4	ROC of x(z) is exterior or interior to circle hence need to specify with z transform of signals.	ROC of x(z) is always exterior to circle hence need not to be specified.

Properties of one sided z transform are same as that of two sided z transform except shifting property.

1) Time delay

$$x(n) \xleftrightarrow{z+} X^{+}(z) \quad \text{And}$$

Then $$x(n-k) \xleftrightarrow{z+} z^{-k}\left[X^{+}(z) + \sum_{n=1}^{k} x(-n)\, z^{n}\right] \qquad k>0$$

2) Time advance

$$x(n) \xleftrightarrow{z+} X^{+}(z) \quad \text{And}$$

Then $$x(n+k) \xleftrightarrow{z+} z^{k}\left[X^{+}(z) - \sum_{n=0}^{k-1} x(n)\, z^{-n}\right] \qquad k>0$$

Examples:

Q) Determine one sided z transform for following signals

1) $x(n)=\{\underline{1},2,3,4,5\}$ 2) $x(n)=\{1,2,\underline{3},4,5\}$

SOLUTION OF DIFFERENTIAL EQUATION

One sided Z transform is very efficient tool for the solution of difference equations with nonzero initial condition. System function of LSI system can be obtained from its difference equation.

$$Z\{x(n-1)\} = \sum_{n=0}^{\infty} x(n-1)\ z^{-n} \qquad \textbf{(One sided Z transform)}$$

$$= x(-1) + x(0)\ z^{-1} + x(1)\ z^{-2} + x(2)\ z^{-3} + \ldots\ldots\ldots\ldots\ldots$$

$$= x(-1) + z^{-1}\ [x(0)\ z^{-1} + x(1)\ z^{-2} + x(2)\ z^{-3} + \ldots\ldots\ldots\ldots\ldots]$$

$$\mathbf{Z\{\ x(n-1)\ \} = z^{-1}\ X(z) + x(-1)}$$
$$\mathbf{Z\{\ x(n-2)\ \} = z^{-2}\ X(z) + z^{-1}\ x(-1) + x(-2)}$$

Similarly

$$\mathbf{Z\{\ x(n+1)\ \} = z\ X(z) - z\ x(0)}$$
$$\mathbf{Z\{\ x(n+2)\ \} = z^{2}\ X(z) - z^{1}\ x(0) + x(1)}$$

1. Difference equations are used to find out the relation between input and output sequences. It is also used to relate system function H(z) and Z transform.

2. The transfer function H(ω) can be obtained from system function H(z) by putting $z=e^{j\omega}$. Magnitude and phase response plot can be obtained by putting various values of ω.

<u>First order Difference Equation</u>

$$\mathbf{y(n) = x(n) + a\ y(n-1)}$$

where y(n) = Output Response of the recursive system
x(n) = Input signal
a= Scaling factor
y(n-1) = Unit delay to output.

Now we will start at n=0

n=0 $y(0) = x(0) + a\ y(-1)$ ....(1)

n=1 $y(1) = x(1) + a\ y(0)$ ....(2)

$= x(1) + a\ [\ x(0) + a\ y(-1)\ ]$

$= a^{2}\ y(-1) + a\ x(0) + x(1)$ ....(3)

hence

$$\mathbf{y(n) = a^{n+1}\ y(-1) + \sum_{k=0}^{n} a^{k}\ x\ (n-k)} \qquad \mathbf{n \geq 0}$$

1) The first part (A) is response depending upon initial condition.
2) The second Part (B) is the response of the system to an input signal.

Zero state response (Forced response) : Consider initial condition are zero. (System is relaxed at time n=0) i.e y(-1) =0

Zero Input response (Natural response) : No input is forced as system is in non-relaxed initial condition. i.e y(-1) != 0
Total response is the sum of zero state response and zero input response.

Q) Determine zero input response for y(n) – 3y(n-1) – 4y(n-2)=0; (Initial Conditions are y(-1)=5 & y(-2)= 10) Answer: y(n)= 7 $(-1)^n$ + 48 $(4)^n$

Q) A difference equation of the system is given below

Y(n)= 0.5 y(n-1) + x(n)

Determine a) System function

b) Pole zero plot

c) Unit sample response

Q) A difference equation of the system is given below

Y(n)= 0.7 y(n-1) – 0.12 y(n-2) + x(n-1) + x(n-2)

a) System Function

b) Pole zero plot

c) Response of system to the input x(n) = nu(n)

d) Is the system stable? Comment on the result.

Q) A difference equation of the system is given below

Y(n)= 0.5 x(n) + 0.5 x(n-1)

Determine a) System function

b) Pole zero plot

c) Unit sample response

d) Transfer function

e) Magnitude and phase plot

Q) A difference equation of the system is given below

a. Y(n)= 0.5 y(n-1) + x(n) + x(n-1)

b. Y(n)= x(n) + 3x(n-1) + 3x(n-2) + x(n-3)

a) System Function

b) Pole zero plot

c) Unit sample response

d) Find values of y(n) for n=0,1,2,3,4,5 for x(n)= δ(n) for no initial condition.

Q) Solve second order difference equation

2x(n-2) – 3x(n-1) + x(n) = 3^{n-2} with x(-2)=-4/9 and x(-1)=-1/3.

Q) Solve second order difference equation

x(n+2) + 3x(n+1) + 2x(n) with x(0)=0 and x(1)=1.

Q) Find the response of the system by using Z transform

x(n+2) - 5x(n+1) + 6x(n)= u(n) with x(0)=0 and x(1)=1.

1.6 CONVOLUTION

1.6.1 LINEAR CONVOLUTION SUM METHOD

1. This method is powerful analysis tool for studying LSI Systems.
2. In this method we decompose input signal into sum of elementary signal. Now the elementary input signals are taken into account and individually given to the system. Now using linearity property whatever output response we get for decomposed input signal, we simply add it & this will provide us total response of the system to any given input signal.
3. Convolution involves folding, shifting, multiplication and summation operations.
4. If there are M number of samples in x(n) and N number of samples in h(n) then the maximum number of samples in y(n) is equals to M+n-1.

Linear Convolution states that

$$y(n) = x(n) * h(n)$$

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

Example 1: h(n) = { 1 , $\underset{\uparrow}{\underline{2}}$, 1, -1 } & x(n) = { $\underline{1}$, 2, $\underset{\uparrow}{3}$, 1 } Find y(n)

METHOD 1: GRAPHICAL REPRESENTATION

Step 1) Find the value of n = n_x+ n_h = -1 (Starting Index of x(n)+ starting index of h(n))

Step 2) y(n)= { y(-1) , y(0) , y(1), y(2), ….} It goes up to length(xn)+ length(yn) -1.

i.e n=-1 y(-1) = x(k) * h(-1-k)
n=0 y(0) = x(k) * h(0-k)
n=1 y(1) = x(k) * h(1-k) ….
ANSWER : y(n) ={1, 4, 8, 8, 3, -2, -1 }

$\uparrow$

METHOD 2: MATHEMATICAL FORMULA

Use Convolution formula

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

k= 0 to 3 (start index to end index of x(n))

y(n) = x(0) h(n) + x(1) h(n-1) + x(2) h(n-2) + x(3) h(n-3)

METHOD 3: VECTOR FORM (TABULATION METHOD)

X(n)= {x1,x2,x3} & h(n) ={ h1,h2,h3}

	X1	x2	x3
h1	h1x1	h1x2	h1x3
h2	h2x1	h2x2	h2x3
h3	h3x1	h3x2	h3x3

y(-1) = h1 x1
y(0) = h2 x1 + h1 x2
y(1) = h1 x3 + h2x2 + h3 x1 …………

METHOD 4: SIMPLE MULTIPLICATION FORM

X(n)= {x1,x2,x3} & h(n) ={ h1,h2,h3}

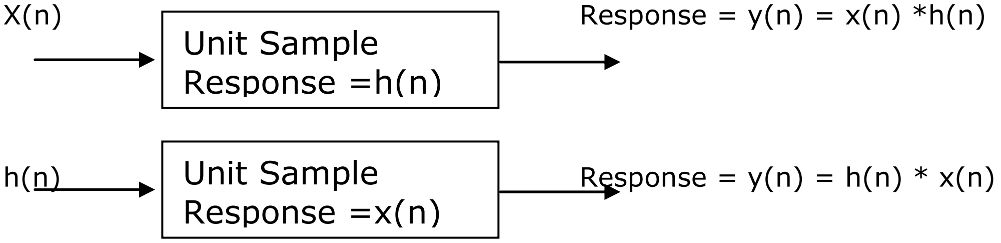
x1 x2 x3
y(n) = ×
y1 y2 y3

1.4.2PROPERTIES OF LINEAR CONVOLUTION

x(n) = Excitation Input signal
y(n) = Output Response
h(n) = Unit sample response

1. Commutative Law: (Commutative Property of Convolution)

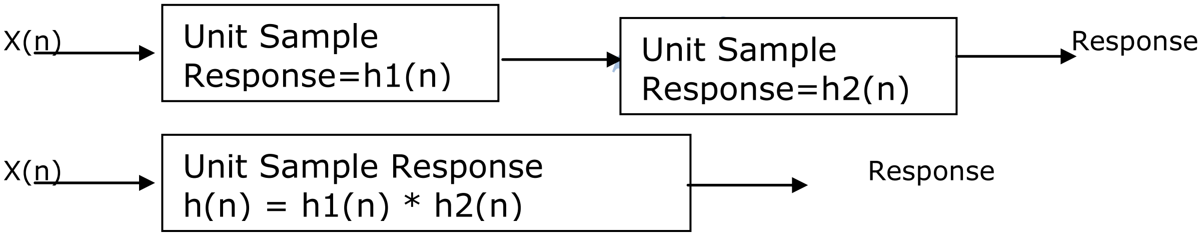
x(n) * h(n) = h(n) * x(n)



X(n)
Unit Sample Response =h(n)
Response = y(n) = x(n) *h(n)
h(n)
Unit Sample Response =x(n)
Response = y(n) = h(n) * x(n)

2. Associate Law: (Associative Property of Convolution)

[x(n) * h1(n)] * h2(n) = x(n) * [h1(n) * h2(n)]



X(n)
Unit Sample Response=h1(n)
Unit Sample Response=h2(n)
Response
X(n)
Unit Sample Response h(n) = h1(n) * h2(n)
Response

3 Distribute Law: (Distributive property of convolution)

x(n) * [h1(n) + h2(n)] = x(n) * h1(n) + x(n) * h2(n)

CAUSALITY OF LSI SYSTEM

The output of causal system depends upon the present and past inputs. The output of the causal system at n= n_0 depends only upon inputs x(n) for n≤ n_0. The linear convolution is given as

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

At n= n_0 ,the output $y(n_0)$ will be

$$y(n_0) = \sum_{k=-\infty}^{\infty} h(k)\ x(n_0-k)$$

Rearranging the above terms...

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

The output of causal system at n= n_0 depends upon the inputs for n< n_0 Hence h(-1)=h(-2)=h(-3)=0

Thus LSI system is causal if and only if

h(n) =0 for n<0

This is the necessary and sufficient condition for causality of the system.
Linear convolution of the causal LSI system is given by

$$y(n) = \sum_{k=0}^{n} x(k)\ h(n-k)$$

STABILITY FOR LSI SYSTEM

A System is said to be stable if every bounded input produces a bounded output. The input x(n) is said to bounded if there exists some finite number M_x such that $|x(n)| \leq M_x < \infty$. The output y(n) is said to bounded if there exists some finite number M_y such that $|y(n)| \leq M_y < \infty$.

Linear convolution is given by

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

Taking the absolute value of both sides

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

The absolute values of total sum is always less than or equal to sum of the absolute values of individually terms. Hence

$$|y(n)| \leq \left| \sum_{k=-\infty}^{\infty} h(k)\ x(n-k) \right|$$

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

The input x(n) is said to bounded if there exists some finite number M_x such that $|x(n)| \leq M_x < \infty$. Hence bounded input x(n) produces bounded output y(n) in the LSI system only if

$$\sum_{k=-\infty}^{\infty} |h(k)| < \infty$$

With this condition satisfied, the system will be stable. The above equation states that the LSI system is stable if its unit sample response is absolutely summable. This is necessary and sufficient condition for the stability of LSI system.

Example 1:

Let us find the response to $x(n) = \begin{cases} \frac{1}{3}, & n = -1, \\ 1, & n = 0, \\ \frac{2}{3}, & n = 1, \\ 0, & \textit{otherwise.} \end{cases}$

The impulse response h of the system is the response to the unit impulse:

$$h(n) = \delta(n) + \frac{1}{2}\delta(n-1) = \begin{cases} 1, & n = 0, \\ \frac{1}{2}, & n = 1, \\ 0, & \textit{otherwise.} \end{cases}$$

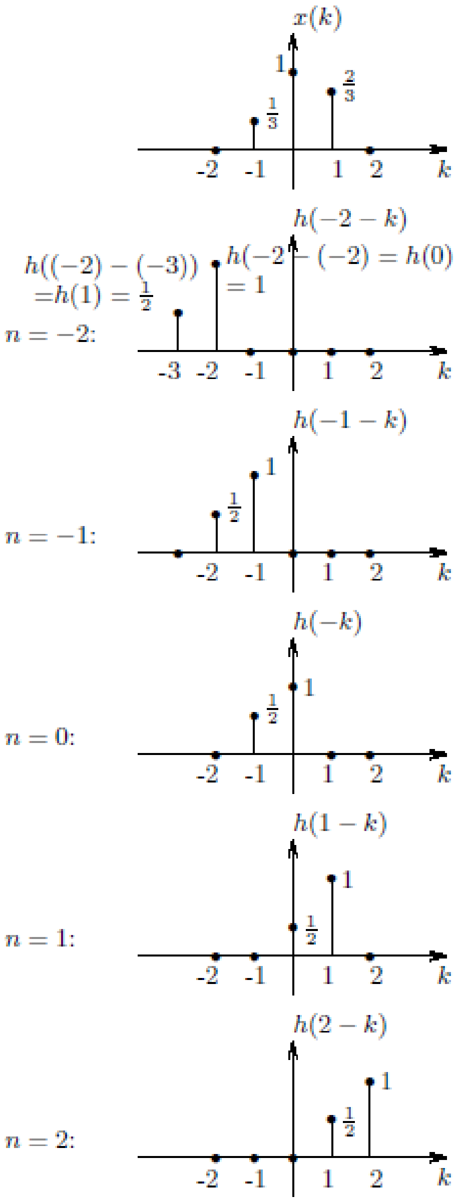
Solution-

(1) flip h;

(2) for a fixed n, shift h by n;

(3) for the same fixed n, multiply $x(k)$ by $h(n-k)$, for each k;

(4) Sum the products over k: $\sum_{k=-\infty}^{\infty} x(k)h(n-k) = y(n)$.



$$y(-2) = \sum_{k=-\infty}^{\infty} x(k)h(-2-k) = 0$$

$$y(-1) = \sum_{k=-\infty}^{\infty} x(k)h(-1-k) = \tfrac{1}{3}\cdot 1 = \tfrac{1}{3}$$

$$y(0) = \tfrac{1}{3}\cdot\tfrac{1}{2} + 1\cdot 1 = \tfrac{7}{6}$$

$$y(1) = 1\cdot\tfrac{1}{2} + \tfrac{2}{3}\cdot 1 = \tfrac{7}{6}$$

$$y(2) = \tfrac{2}{3}\cdot\tfrac{1}{2} = \tfrac{1}{3}$$

$$y(n) = \begin{cases} \frac{1}{3}, & n = -1, 2, \\ \frac{1}{6}, & n = 0, 1, \\ 0, & otherwise. \end{cases}$$

Example 2:

To evaluate the convolution of signals $x(n) = 2^{-|n|}$ *and* $h(n) = u(n)$*,*

we substitute the two expressions into the definition of convolution:

$$\begin{aligned} y(n) &= x * h(n) = \sum_{k=-\infty}^{\infty} 2^{-|k|} u(n-k) \\ &= \sum_{k=-\infty}^{n} 2^{-|k|}. \end{aligned}$$

For $n \leq 0$*, the summation is only over nonpositive values of* k *and therefore* $|k|$ *can be replaced with* $-k$*:*

$$\begin{aligned} y(n) &= \sum_{k=-\infty}^{n} 2^{-(-k)} = \sum_{k=-\infty}^{n} 2^{k} \\ &= \sum_{m=-n}^{\infty} 2^{-m} = \sum_{m=-n}^{\infty} \left(\frac{1}{2}\right)^{m} = \frac{(1/2)^{-n}}{1-1/2} \\ &= 2^{n+1}, \text{ for any integer } n \leq 0, \end{aligned}$$

where we substituted $m = -k$*. When* $n > 0$*, the summation can be broken into two pieces: one for nonpositive values of* k *(i.e. for* k *from* $-\infty$ *to* 0*) and the other for positive values of* k *(i.e. for* k *from* 1 *to* n*):*

$$\begin{aligned} y(n) &= \sum_{k=-\infty}^{0} 2^{-(-k)} + \sum_{k=1}^{n} 2^{-k} \\ &= \sum_{m=0}^{\infty} 2^{-m} + \sum_{k=1}^{n} 2^{-k} \\ &= \frac{1}{1-1/2} + \frac{(1/2)^{1} - (1/2)^{n+1}}{1-1/2} \\ &= 3 - 2^{-n}, \text{ for any integer } n > 0. \end{aligned}$$

Putting together the two cases,

$$y(n) = \begin{cases} 2^{n+1}, & n \leq 0, \\ 3 - 2^{-n}, & n > 0. \end{cases} \qquad \blacksquare$$

SELF-STUDY: Exercise No. 1

Q1) Show that the discrete time signal is periodic only if its frequency is expressed as the ratio of two integers.

Q2) Show that the frequency range for discrete time sinusoidal signal is -∏ to ∏ radians/sample or -½ cycles/sample to ½ cycles/sample.

Q3) Prove δ (n)= u(n)= u(n-1).

Q4) Prove $u(n)= \sum_{k=-\infty}^{n} \delta(k)$

Q5) Prove $u(n)= \sum_{k=0}^{\infty} \delta(n-k)$

Q6) Prove that every discrete sinusoidal signal can be expressed in terms of weighted unit impulse.

Q7) Prove the Linear Convolution theorem.

1.7 CORRELATION:

It is frequently necessary to establish similarity between one set of data and another. It means we would like to correlate two processes or data. Correlation is closely related to convolution, because the correlation is essentially convolution of two data sequences in which one of the sequences has been reversed.

Applications are in

1) Images processing for robotic vision or remote sensing by satellite in which data from different image is compared

2) In radar and sonar systems for range and position finding in which transmitted and reflected waveforms are compared.

3) Correlation is also used in detection and identifying of signals in noise.

4) Computation of average power in waveforms.

5) Identification of binary codeword in pulse code modulation system.

1.7.1 DIFFERENCE BETWEEN LINEAR CONVOLUTION AND CORRELATION

Sr No	Linear Convolution	Correlation
1	In case of convolution two signal sequences input signal and impulse response given by the same system is calculated	In case of Correlation, two signal sequences are just compared.
2	Our main aim is to calculate the response given by the system.	Our main aim is to measure the degree to which two signals are similar and thus to extract some information that depends to a large extent on the application
3	Linear Convolution is given by the equation y(n) = x(n) * h(n) & calculated as $y(n) = \sum_{k=-\infty}^{\infty} x(k)\ h(n-k)$	Received signal sequence is given as Y(n) = ɑ x(n-D) + ω(n) Where ɑ= Attenuation Factor D= Delay ω(n) = Noise signal
4	Linear convolution is commutative	Not commutative.

1.7.2 TYPES OF CORRELATION

Under Correlation there are two classes.

1) **CROSS CORRELATION**: When the correlation of two different sequences x(n) and y(n) is performed it is called as Cross correlation. Cross-correlation of x(n) and y(n) is $r_{xy}(l)$ which can be mathematically expressed as

$$\mathbf{r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n)\ y(n-l)}$$

OR

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

2) **AUTO CORRELATION**: In Auto-correlation we correlate signal x(n) with itself, which can be mathematically expressed as

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

OR

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

1.7.3 PROPERTIES OF CORRELATION

1) The cross-correlation is not commutative.

$$r_{xy}(l) = r_{yx}(-l)$$

2) The cross-correlation is equivalent to convolution of one sequence with folded version of another sequence.

$$r_{xy}(l) = x(l) * y(-l).$$

3) The autocorrelation sequence is an even function.

$$r_{xx}(l) = r_{xx}(-l)$$

Examples:

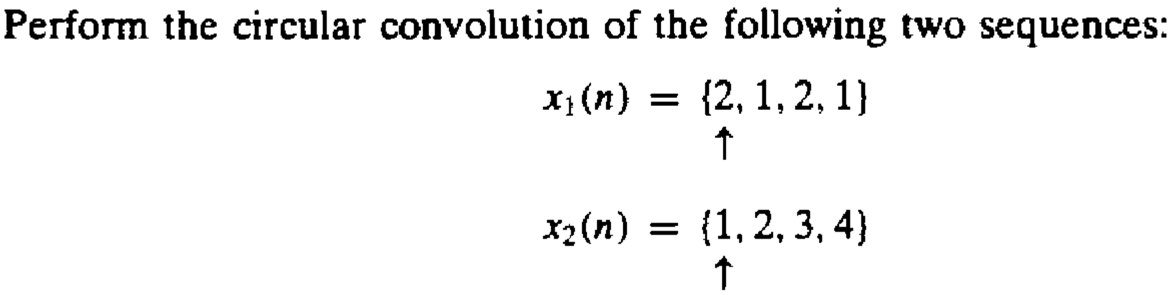
Q) Determine cross-correlation sequence

$x(n)=\{2, -1, 3, 7, \underline{1}, 2, -3\}$ & $y(n)=\{1, -1, 2, -2, \underline{4}, 1, -2, 5\}$

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

Q) Determine autocorrelation sequence

$x(n)=\{\underline{1}, 2, 1, 1\}$ Answer: $r_{xx}(l) = \{1, 3, 5, \underline{7}, 5, 3, 1\}$



Perform the circular convolution of the following two sequences:

$$x_1(n) = \{\underset{\uparrow}{2}, 1, 2, 1\}$$

$$x_2(n) = \{\underset{\uparrow}{1}, 2, 3, 4\}$$

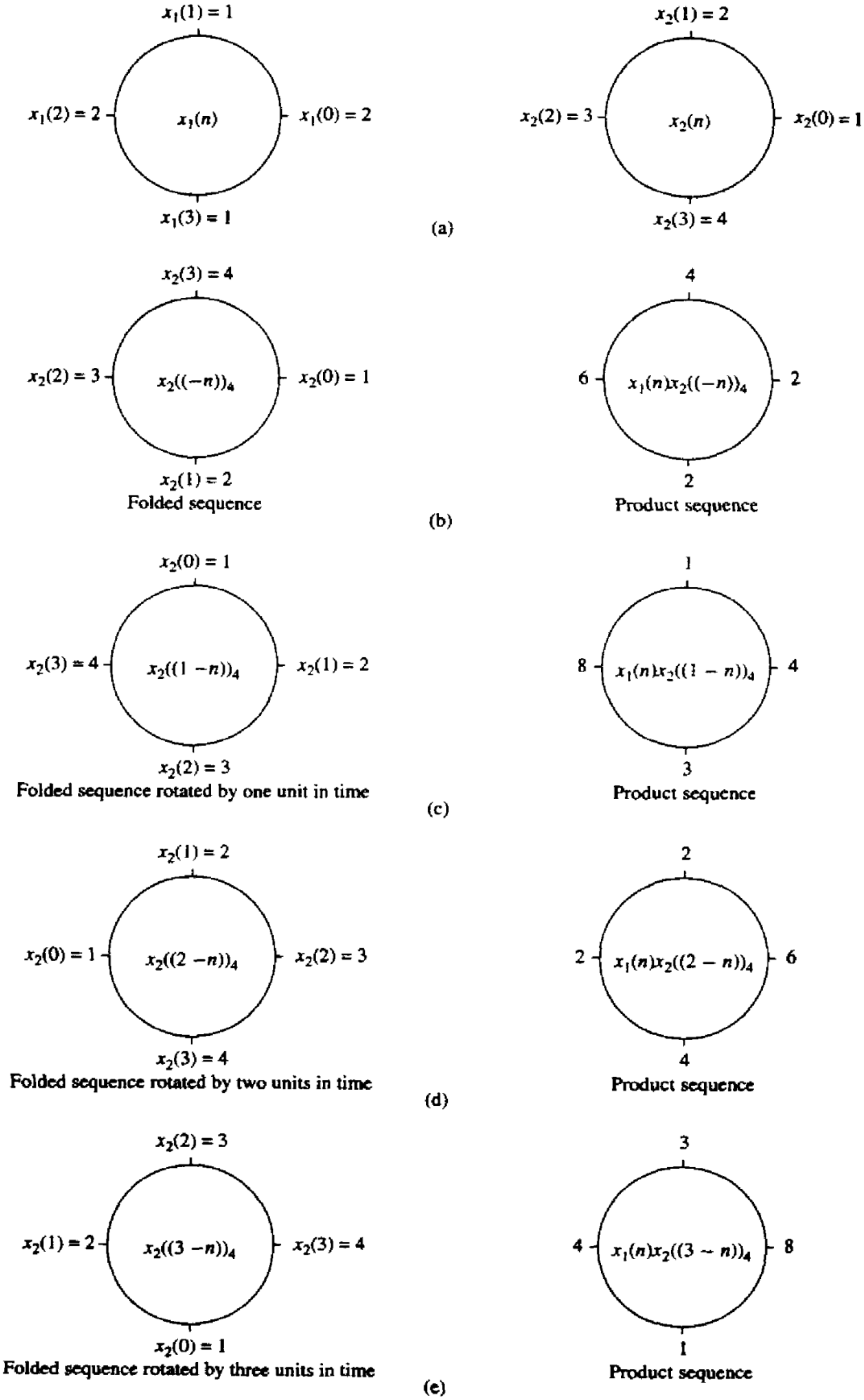


Figure 5.8 Circular convolution of two sequences.

UNIT - 2
FREQUENCY TRANSFORMATIONS

2.1 INTRODUCTION

Any signal can be decomposed in terms of sinusoidal (or complex exponential) components. Thus the analysis of signals can be done by transforming time domain signals into frequency domain and vice-versa. This transformation between time and frequency domain is performed with the help of Fourier Transform(FT) But still it is not convenient for computation by DSP processors hence Discrete Fourier Transform(DFT) is used.

Time domain analysis provides some information like amplitude at sampling instant but does not convey frequency content & power, energy spectrum hence frequency domain analysis is used.

For Discrete time signals x(n) , Fourier Transform is denoted as x(ω) & given by

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j\omega n} \qquad \text{FT..........(1)}$$

DFT is denoted by x(k) and given by (ω= 2 ∏ k/N)

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\Pi kn/N} \qquad \text{DFT.......(2)}$$

IDFT is given as

$$x(n) = 1/N \sum_{k=0}^{N-1} X(k)\, e^{j2\Pi kn/N} \qquad \text{IDFT......(3)}$$

2.2 DIFFERENCE BETWEEN FT & DFT

Sr No	Fourier Transform (FT)	Discrete Fourier Transform (DFT)
1	FT x(ω) is the continuous function of x(n).	DFT x(k) is calculated only at discrete values of ω. Thus DFT is discrete in nature.
2	The range of ω is from - ∏ to ∏ or 0 to 2∏.	Sampling is done at N equally spaced points over period 0 to 2∏. Thus DFT is sampled version of FT.
3	FT is given by equation (1)	DFT is given by equation (2)
4	FT equations are applicable to most of infinite sequences.	DFT equations are applicable to causal, finite duration sequences
5	In DSP processors & computers applications of FT are limited because x(ω) is continuous function of ω.	In DSP processors and computers DFT's are mostly used. **APPLICATION** a) Spectrum Analysis b) Filter Design

Q) Prove that FT $x(\omega)$ is periodic with period 2Π.
Q) Determine FT of $x(n)= a^{n}\, u(n)$ for $-1< a < 1$.
Q) Determine FT of $x(n)= A$ for $0 \le n \le L-1$.
Q) Determine FT of $x(n)= u(n)$
Q) Determine FT of $x(n)= \delta(n)$
Q) Determine FT of $x(n)= e^{-at}\, u(t)$

2.3 CALCULATION OF DFT & IDFT

For calculation of DFT & IDFT two different methods can be used. First method is using mathematical equation & second method is 4 or 8 point DFT. If x(n) is the sequence of N samples then consider $W_N= e^{-j2\,\Pi\,/\,N}$ (twiddle factor)

Four POINT DFT (4-DFT)

Sr No	$W_N=W_4=e^{-j\,\Pi/2}$	Angle	Real	Imaginary	Total
1	$W_4^{\,0}$	0	1	0	1
2	$W_4^{\,1}$	$-\,\Pi/2$	0	-j	-j
3	$W_4^{\,2}$	$-\,\Pi$	-1	0	-1
4	$W_4^{\,3}$	$-\,3\,\Pi\,/2$	0	J	J

$$[\mathbf{W}_N] = \begin{array}{c} \\ \mathbf{k=0} \\ \mathbf{k=1} \\ \mathbf{k=2} \\ \mathbf{k=3} \end{array} \begin{array}{cccc} \mathbf{n=0} & \mathbf{n=1} & \mathbf{n=2} & \mathbf{n=3} \\ W_4^{\,0} & W_4^{\,0} & W_4^{\,0} & W_4^{\,0} \\ W_4^{\,0} & W_4^{\,1} & W_4^{\,2} & W_4^{\,3} \\ W_4^{\,0} & W_4^{\,2} & W_4^{\,4} & W_4^{\,6} \\ W_4^{\,0} & W_4^{\,3} & W_4^{\,6} & W_4^{\,9} \end{array}$$

Thus 4 point DFT is given as $X_N = [W_N\,]\,X_N$

$$[\mathbf{W_N}] = \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{-j} & \mathbf{-1} & \mathbf{j} \\ \mathbf{1} & \mathbf{-1} & \mathbf{1} & \mathbf{-1} \\ \mathbf{1} & \mathbf{j} & \mathbf{-1} & \mathbf{-j} \end{pmatrix}$$

EIGHT POINT DFT (8-DFT)

Sr No	$W_N = W_8= e^{-j\,\Pi/4}$	Angle	Magnitude	Imaginary	Total
1	$W_8^{\,0}$	0	1	----	1
2	$W_8^{\,1}$	$-\,\Pi/4$	$1/\sqrt{2}$	$-j\ 1/\sqrt{2}$	$1/\sqrt{2}\ -j\ 1/\sqrt{2}$
3	$W_8^{\,2}$	$-\,\Pi/2$	0	-j	-j
4	$W_8^{\,3}$	$-\,3\,\Pi\,/4$	$-1/\sqrt{2}$	$-j\ 1/\sqrt{2}$	$-\ 1/\sqrt{2}\ -j\ 1/\sqrt{2}$
5	$W_8^{\,4}$	$-\,\Pi$	-1	----	-1
6	$W_8^{\,5}$	$-\,5\Pi\,/\,4$	$-1/\sqrt{2}$	$+j\ 1/\sqrt{2}$	$-\ 1/\sqrt{2}\ +\ j\ 1/\sqrt{2}$
7	$W_8^{\,6}$	$-\,7\Pi\,/\,4$	0	J	J
8	$W_8^{\,7}$	$-\,2\Pi$	$1/\sqrt{2}$	$+j\ 1/\sqrt{2}$	$1/\sqrt{2}\ +\ j\ 1/\sqrt{2}$

Remember that $W_8^{\,0} = W_8^{\,8} = W_8^{\,16} = W_8^{\,24} = W_8^{\,32} = W_8^{\,40}$ (Periodic Property)
Magnitude and phase of x(k) can be obtained as,
$|x(k)| = \text{sqrt}\ (X_r(k)^2 + X_I(k)^2)$
Angle $x(k) = \tan^{-1} (X_I(k) / X_R(k))$

Examples:

Q) Compute DFT of x(n) = {0,1,2,3} Ans: x4=[6, -2+2j, -2, -2-2j]
Q) Compute DFT of x(n) = {1,0,0,1} Ans: x4=[2, 1+j, 0, 1-j]
Q) Compute DFT of x(n) = {1,0,1,0} Ans: x4=[2, 0, 2, 0]
Q) Compute IDFT of x(k) = {2, 1+j, 0, 1-j } Ans: x4=[1,0,0,1]

2.4 DIFFERENCE BETWEEN DFT & IDFT

Sr No	DFT (Analysis transform)	IDFT (Synthesis transform)
1	DFT is finite duration discrete frequency sequence that is obtained by sampling one period of FT.	IDFT is inverse DFT which is used to calculate time domain representation (Discrete time sequence) form of x(k).
2	DFT equations are applicable to causal finite duration sequences.	IDFT is used basically to determine sample response of a filter for which we know only transfer function.
3	Mathematical Equation to calculate DFT is given by $\mathbf{X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi kn/N}}$	Mathematical Equation to calculate IDFT is given by $\mathbf{x(n) = 1/N \sum_{n=0}^{N-1} X(k) e^{j2\pi kn/N}}$
4	Thus DFT is given by $\mathbf{X(k)= [W_N][xn]}$	In DFT and IDFT difference is of factor 1/N & sign of exponent of twiddle factor. Thus $\mathbf{x(n)= 1/N\ [W_N]^{-1}[X_K]}$

2.5 PROPERTIES OF DFT

$$x(n) \underset{N}{\overset{DFT}{\longleftrightarrow}} x(k)$$

1. Periodicity

Let x(n) and x(k) be the DFT pair then if

$x(n+N) = x(n)$ for all n then
$X(k+N) = X(k)$ for all k

Thus periodic sequence xp(n) can be given as

$$xp(n) = \sum_{l=-\infty}^{\infty} x(n-lN)$$

2. Linearity

The linearity property states that if

$$x1(n) \overset{DFT}{\longleftrightarrow} X1(k) \text{ And}$$

N

$$x2(n) \xleftrightarrow[N]{DFT} X2(k) \text{ Then}$$

Then

$$a1\,x1(n) + a2\,x2(n) \xleftrightarrow[N]{DFT} a1\,X1(k) + a2\,X2(k)$$

DFT of linear combination of two or more signals is equal to the same linear combination of DFT of individual signals.

3. Circular Symmetries of a sequence

A) A sequence is said to be circularly even if it is symmetric about the point zero on the circle. Thus X(N-n) = x(n)

B) A sequence is said to be circularly odd if it is anti symmetric about the point zero on the circle. Thus X(N-n) = - x(n)

C) A circularly folded sequence is represented as $x((-n))_N$ and given by $x((-n))_N = x(N-n)$.

D) Anticlockwise direction gives delayed sequence and clockwise direction gives advance sequence. Thus delayed or advances sequence x\`(n) is related to x(n) by the circular shift.

4. Symmetry Property of a sequence

A) Symmetry property for real valued x(n) i.e $x_I(n)=0$

This property states that if x(n) is real then $X(N-k) = X^*(k)=X(-k)$

B) Real and even sequence x(n) i.e $x_I(n)=0$ & $X_I(K)=0$

This property states that if the sequence is real and even x(n)= x(N-n) then DFT becomes

$$X(k) = \sum_{n=0}^{N-1} x(n) \cos (2\Pi kn/N)$$

C) Real and odd sequence x(n) i.e $x_I(n)=0$ & $X_R(K)=0$

This property states that if the sequence is real and odd x(n)=-x(N-n) then DFT becomes

$$X(k) = -j \sum_{n=0}^{N-1} x(n) \sin (2\Pi kn/N)$$

D) Pure Imaginary x(n) i.e $x_R(n)=0$

This property states that if the sequence is purely imaginary $x(n)=j\ X_I(n)$ then DFT becomes

$$X_R(k) = \sum_{n=0}^{N-1} x_I(n) \sin (2\Pi kn/N)$$

$$X_I(k) = \sum_{n=0}^{N-1} x_I(n) \cos (2\prod kn/N)$$

5. Circular Convolution

The Circular Convolution property states that if

$$x1(n) \overset{DFT}{\underset{N}{\longleftrightarrow}} X1(k) \quad \text{And}$$

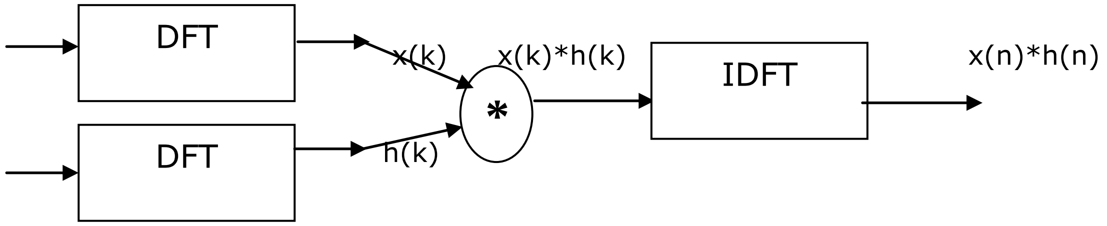
$$x2(n) \overset{DFT}{\underset{N}{\longleftrightarrow}} X2(k) \quad \text{Then}$$

$$\text{Then } x1(n) \; \textcircled{N} \; x2(n) \overset{DFT}{\underset{N}{\longleftrightarrow}} x1(k) \;\; x2(k)$$

It means that circular convolution of x1(n) & x2(n) is equal to multiplication of their DFT's.
Thus circular convolution of two periodic discrete signal with period N is given by

$$\mathbf{y(m) = \sum_{n=0}^{N-1} x1\,(n)\; x2\,(m-n)_N} \qquad \qquad \mathbf{..........(4)}$$

Multiplication of two sequences in time domain is called as Linear convolution while Multiplication of two sequences in frequency domain is called as circular convolution. Results of both are totally different but are related with each other.



There are two different methods are used to calculate circular convolution
1) Graphical representation form
2) Matrix approach

DIFFERENCE BETWEEN LINEAR CONVOLUTION & CIRCULAR CONVOLUTION

Sr No	Linear Convolution	Circular Convolution
1	In case of convolution two signal sequences input signal x(n) and impulse response h(n) given by the same system, output y(n) is calculated	Multiplication of two DFT's is called as circular convolution.
2	Multiplication of two sequences in time domain is called as Linear convolution	Multiplication of two sequences in frequency domain is called as circular convolution.
3	Linear Convolution is given by the equation y(n) = x(n) * h(n) & calculated as $y(n) = \sum^{\infty} x\,(k)\;\; h(n - k\,)$	Circular Convolution is calculated as $y(m) = \sum_{n=0}^{N-1} x1\,(n)\; x2\,(m-n)_N$

	k= -∞	
4	Linear Convolution of two signals returns N-1 elements where N is sum of elements in both sequences.	Circular convolution returns same number of elements that of two signals.

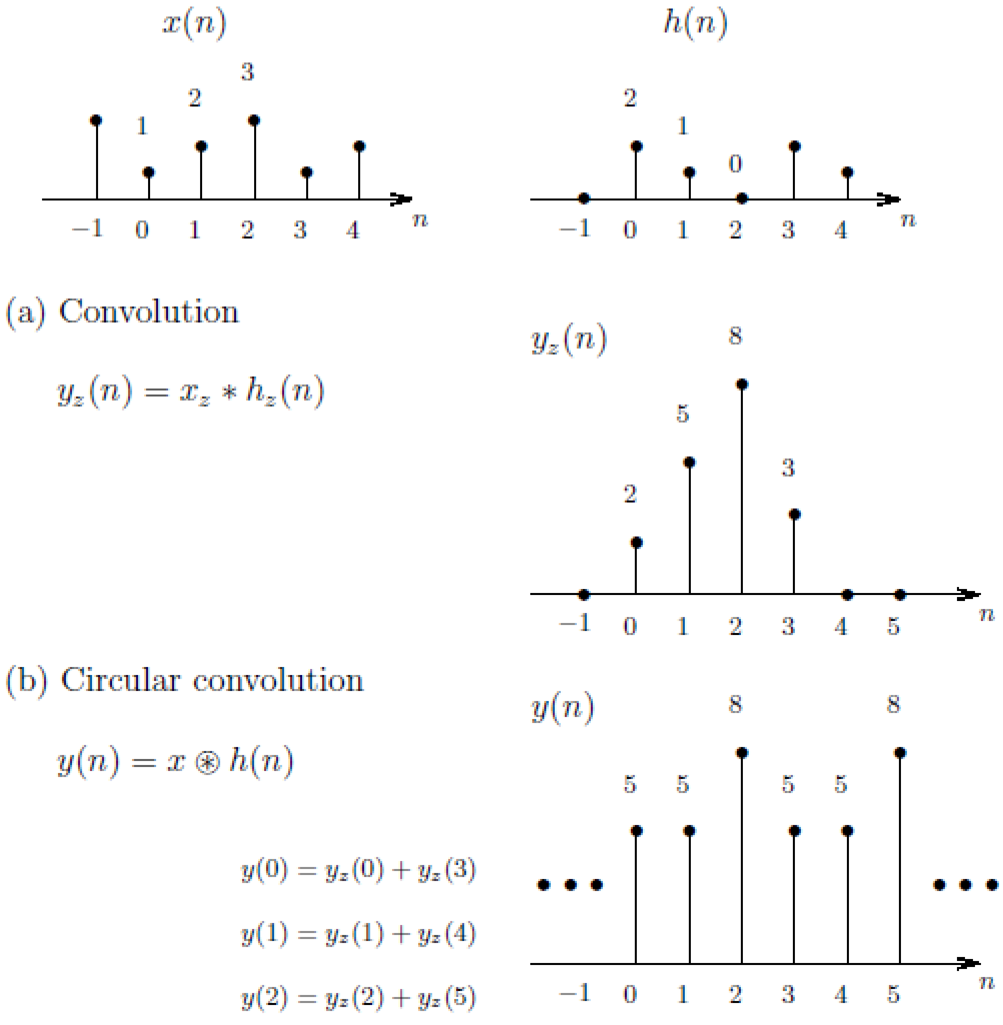
Q) The two sequences x1(n)={2,1,2,1} & x2(n)={1,2,3,4}. Find out the sequence x3(m) which is equal to circular convolution of two sequences. Ans: X3(m)={14,16,14,16}

Q) x1(n)={1,1,1,1,-1,-1,-1,-1} & x2(n)={0,1,2,3,4,3,2,1}. Find out the sequence x3(m) which is equal to circular convolution of two sequences. Ans: X3(m)={-4,-8,-8,-4,4,8,8,4}

Q) Perform Linear Convolution of x(n)={1,2} & h(n)={2,1} using DFT & IDFT.

Q) Perform Linear Convolution of x(n)={1,2,2,1} & h(n)={1,2,3} using 8 Pt DFT & IDFT.

DIFFERENCE BETWEEN LINEAR CONVOLUTION & CIRCULAR CONVOLUTION



(a) Convolution

$y_z(n) = x_z * h_z(n)$

(b) Circular convolution

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

$y(0) = y_z(0) + y_z(3)$

$y(1) = y_z(1) + y_z(4)$

$y(2) = y_z(2) + y_z(5)$

6. Multiplication

The Multiplication property states that if

$$X1(n) \xleftrightarrow[N]{DFT} x1(k) \text{ And}$$

$$X2(n) \xleftrightarrow[N]{DFT} x2(k) \text{ Then}$$

$$\text{Then} \quad x1(n)\ x2(n) \xleftrightarrow[N]{DFT} 1/N \ \ x1(k) \text{ \textcircled{N} } x2(k)$$

It means that multiplication of two sequences in time domain results in circular convolution of their DFT’s in frequency domain.

7. Time reversal of a sequence

The Time reversal property states that if

$$X(n) \xleftrightarrow[N]{DFT} x(k) \text{ And}$$

$$\text{Then } x((-n))_N = x(N-n) \xleftrightarrow[N]{DFT} x((-k))_N = x(N-k)$$

It means that the sequence is circularly folded its DFT is also circularly folded.

8. Circular Time shift

The Circular Time shift states that if

$$X(n) \xleftrightarrow[N]{DFT} x(k) \text{ And}$$

$$\text{Then} \quad x((n-l))_N \xleftrightarrow[N]{DFT} x(k)\ e^{-j2\ \Pi\ k\ l\ /\ N}$$

Thus shifting the sequence circularly by ‘l’ samples is equivalent to multiplying its DFT by $e^{-j2\ \Pi\ k\ l\ /\ N}$

9. Circular frequency shift

The Circular frequency shift states that if

$$X(n) \xleftrightarrow[N]{DFT} x(k) \text{ And}$$

$$\text{Then} \quad x(n)\ e^{j2\ \Pi\ l\ n\ /\ N} \xleftrightarrow[N]{DFT} x((n-l))_N$$

Thus shifting the frequency components of DFT circularly is equivalent to multiplying its time domain sequence by $e^{-j2\ \Pi\ k\ l\ /\ N}$

10. Complex conjugate property

The Complex conjugate property states that if

$$X(n) \overset{DFT}{\underset{N}{\longleftrightarrow}} x(k) \text{ then}$$

$$x^{*}(n) \overset{DFT}{\underset{N}{\longleftrightarrow}} x^{*}((-k))_{N} = x^{*}(N-k) \text{ And}$$

$$x^{*}((-n))_{N} = x^{*}(N-k) \overset{DFT}{\underset{N}{\longleftrightarrow}} x^{*}(k)$$

11. Circular Correlation

The Complex correlation property states

$$r_{xy}(l) \overset{DFT}{\underset{N}{\longleftrightarrow}} R_{xy}(k)= x(k)\ Y^{*}(k)$$

Here $r_{xy}(l)$ is circular cross correlation which is given as

$$r_{xy}(l) = \sum_{n=0}^{N-1} x(n)\ y^{*}((n - l))_{N}$$

This means multiplication of DFT of one sequence and conjugate DFT of another sequence is equivalent to circular cross-correlation of these sequences in time domain.

12. Parseval's Theorem

The Parseval's theorem states

$$\sum_{n=0}^{N-1} X(n)\ y^{*}(n) = 1/N \sum_{n=0}^{N-1} x(k)\ y^{*}(k)$$

This equation give energy of finite duration sequence in terms of its frequency components.

2.6 APPLICATION OF DFT

1. DFT FOR LINEAR FILTERING

Consider that input sequence x(n) of Length L & impulse response of same system is h(n) having M samples. Thus y(n) output of the system contains N samples where N=L+M-1. If DFT of y(n) also contains N samples then only it uniquely represents y(n) in time domain. Multiplication of two DFT's is equivalent to circular convolution of corresponding time domain sequences. But the length of x(n) & h(n) is less than N. Hence these sequences are appended with zeros to make their length N called as "Zero padding". The N point circular convolution and linear convolution provide the same sequence. Thus linear convolution can be obtained by circular convolution. Thus linear filtering is provided by DFT.

When the input data sequence is long then it requires large time to get the output sequence. Hence other techniques are used to filter long data sequences. Instead of finding the output of complete input sequence it is broken into small length sequences. The output due to these small length sequences are computed fast. The outputs due to these small length sequences are fitted one after another to get the final output response.

METHOD 1: OVERLAP SAVE METHOD OF LINEAR FILTERING

Step 1> In this method L samples of the current segment and M-1 samples of the previous segment forms the input data block. Thus data block will be

X1(n) ={0,0,0,0,0,………………… ,x(0),x(1),…………….x(L-1)}
X2(n) ={x(L-M+1), …………….x(L-1),x(L),x(L+1),,,,,,,,,,,,,x(2L-1)}
X3(n) ={x(2L-M+1), …………….x(2L-1),x(2L),x(2L+2),,,,,,,,,,,,,x(3L-1)}

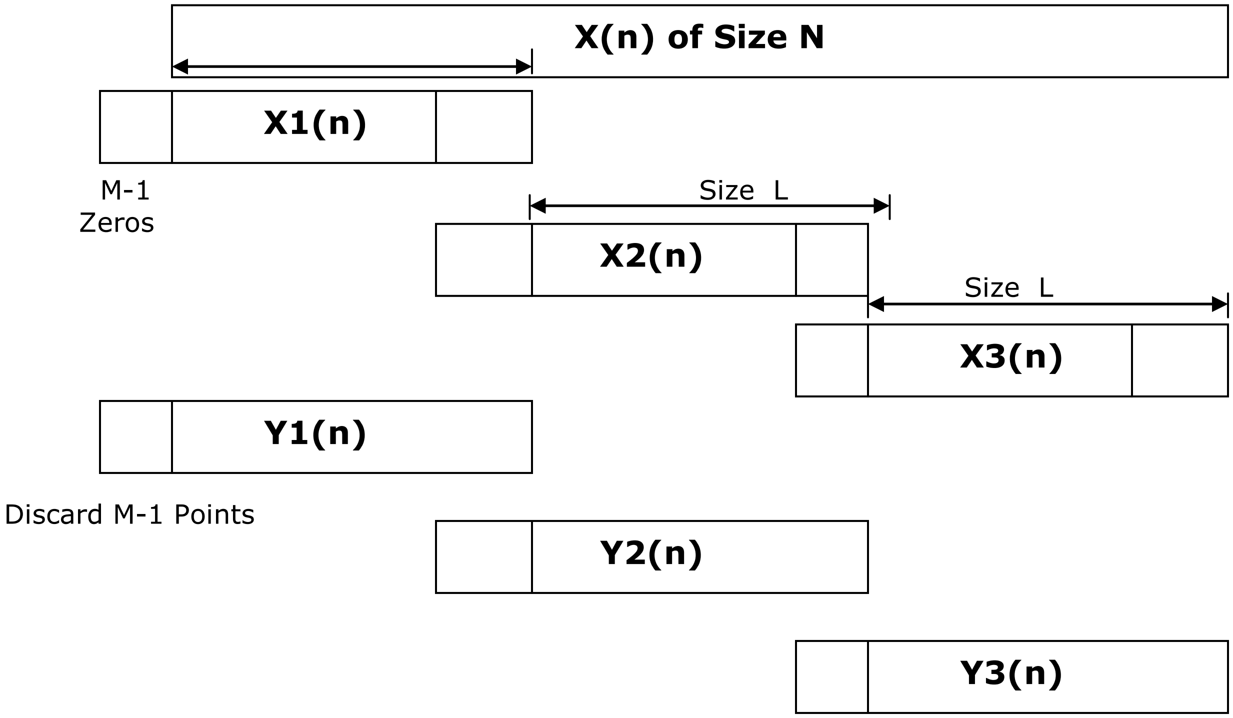
Step2> Unit sample response h(n) contains M samples hence its length is made N by padding zeros. Thus h(n) also contains N samples.

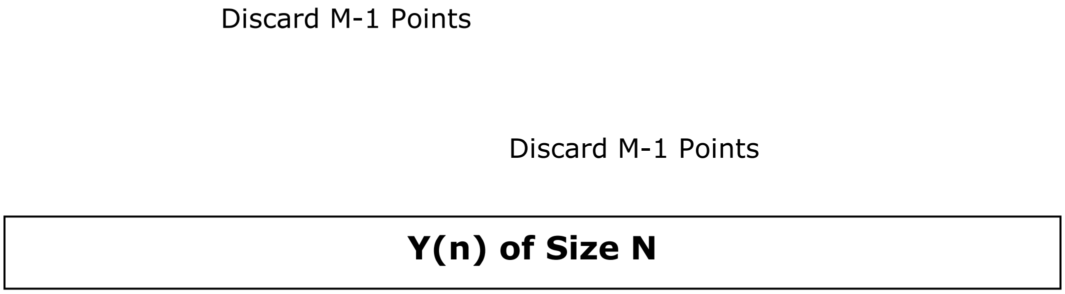
h(n)={ h(0), h(1), …………….h(M-1), 0,0,0,……………………(L-1 zeros)}

Step3> The N point DFT of h(n) is H(k) & DFT of mth data block be $x_m(K)$ then corresponding DFT of output be $Y`_m(k)$

$$Y`_m(k)= H(k)\, x_m(K)$$

Step 4> The sequence $y_m(n)$ can be obtained by taking N point IDFT of $Y`_m(k)$. Initial (M-1) samples in the corresponding data block must be discarded. The last L samples are the correct output samples. Such blocks are fitted one after another to get the final output.





METHOD 2: OVERLAP ADD METHOD OF LINEAR FILTERING

Step 1> In this method L samples of the current segment and M-1 samples of the previous segment forms the input data block. Thus data block will be

X1(n) ={x(0),x(1),…………….x(L-1),0,0,0,……….}
X2(n) ={x(L),x(L+1),x(2L-1),0,0,0,0}
X3(n) ={x(2L),x(2L+2),,,,,,,,,,,,,x(3L-1),0,0,0,0}

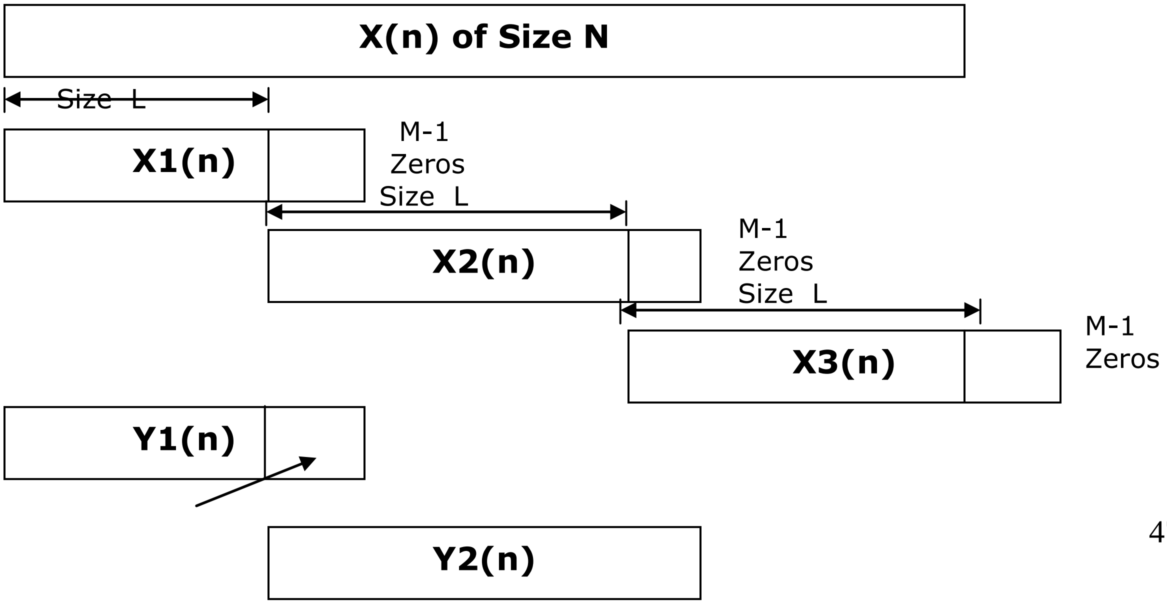
Step2> Unit sample response h(n) contains M samples hence its length is made N by padding zeros. Thus h(n) also contains N samples.

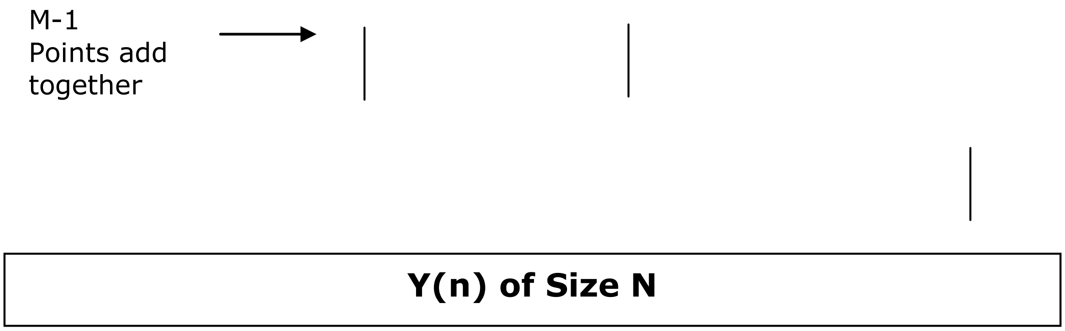
h(n)={ h(0), h(1), …………….h(M-1), 0,0,0,……………………(L-1 zeros)}

Step3> The N point DFT of h(n) is H(k) & DFT of m^{th} data block be $x_m(K)$ then corresponding DFT of output be $Y`_m(k)$

$$Y`_m(k)= H(k)\, x_m(K)$$

Step 4> The sequence $y_m(n)$ can be obtained by taking N point IDFT of $Y`_m(k)$. Initial (M-1) samples are not discarded as there will be no aliasing. The last (M-1) samples of current output block must be added to the first M-1 samples of next output block. Such blocks are fitted one after another to get the final output.





DIFFERENCE BETWEEN OVERLAP SAVE AND OVERLAP ADD METHOD

Sr No	OVERLAP SAVE METHOD	OVERLAP ADD METHOD
1	In this method, L samples of the current segment and (M-1) samples of the previous segment forms the input data block.	In this method L samples from input sequence and padding M-1 zeros forms data block of size N.
2	Initial M-1 samples of output sequence are discarded which occurs due to aliasing effect.	There will be no aliasing in output data blocks.
3	To avoid loss of data due to aliasing last M-1 samples of each data record are saved.	Last M-1 samples of current output block must be added to the first M-1 samples of next output block. Hence called as overlap add method.

2. SPECTRUM ANALYSIS USING DFT

DFT of the signal is used for spectrum analysis. DFT can be computed on digital computer or digital signal processor. The signal to be analyzed is passed through anti-aliasing filter and samples at the rate of Fs≥ 2 Fmax. Hence highest frequency component is Fs/2.

Frequency spectrum can be plotted by taking N number of samples & L samples of waveforms. The total frequency range 2∏ is divided into N points. Spectrum is better if we take large value of N & L But this increases processing time. DFT can be computed quickly using FFT algorithm hence fast processing can be done. Thus most accurate resolution can be obtained by increasing number of samples.

2.7 FAST FOURIER ALGORITHM (FFT)

1. Large number of the applications such as filtering, correlation analysis, spectrum analysis require calculation of DFT. But direct computation of DFT require large number of computations and hence processor remain busy. Hence special algorithms are developed to compute DFT quickly called as Fast Fourier algorithms (FFT).

2. The radix-2 FFT algorithms are based on divide and conquer approach. In this method, the N-point DFT is successively decomposed into smaller DFT's. Because of this decomposition, the number of computations are reduced.

RADIX-2 FFT ALGORITHMS

1. DECIMATION IN TIME (DITFFT)

There are three properties of twiddle factor W_N

1) $W_N^{k+N} = W_N^K$ (Periodicity Property)
2) $W_N^{k+N/2} = -W_N^K$ (Symmetry Property)
3) $W_N^2 = W_{N/2}$.

N point sequence x(n) be splitted into two N/2 point data sequences f1(n) and f2(n). f1(n) contains even numbered samples of x(n) and f2(n) contains odd numbered samples of x(n). This splitted operation is called decimation. Since it is done on time domain sequence it is called "**Decimation in Time**". Thus

f1(m)=x(2m) where n=0,1,…………N/2-1
f2(m)=x(2m+1) where n=0,1,…………N/2-1

N point DFT is given as

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{kn} \quad (1)$$

Since the sequence x(n) is splitted into even numbered and odd numbered samples, thus

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

$$X(k) = F1(k) + W_N^k\, F2(k) \quad (3)$$

$$X(k+N/2) = F1(k) - W_N^k\, F2(k) \text{ (Symmetry property)} \quad (4)$$

Fig 1 shows that 8-point DFT can be computed directly and hence no reduction in computation.

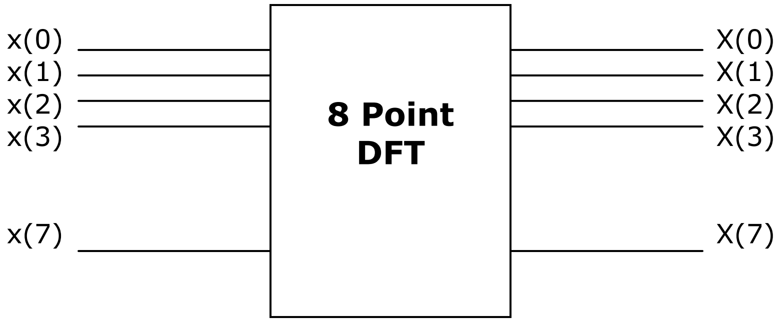


Fig 1. DIRECT COMPUTATION FOR N=8

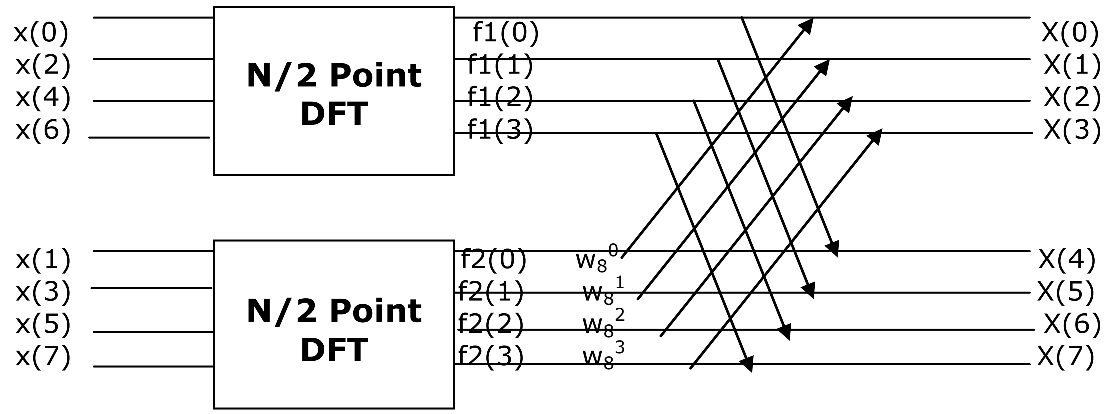


Fig 2. FIRST STAGE FOR FFT COMPUTATION FOR N=8

Fig 3 shows N/2 point DFT base separated in N/4 boxes. In such cases equations become

$g1(k) = P1(k) + W_N^{2k}\ P2(k)$ (5)

$g1(k+N/2) = p1(k) - W_N^{2k}\ P2(k)$ (6)

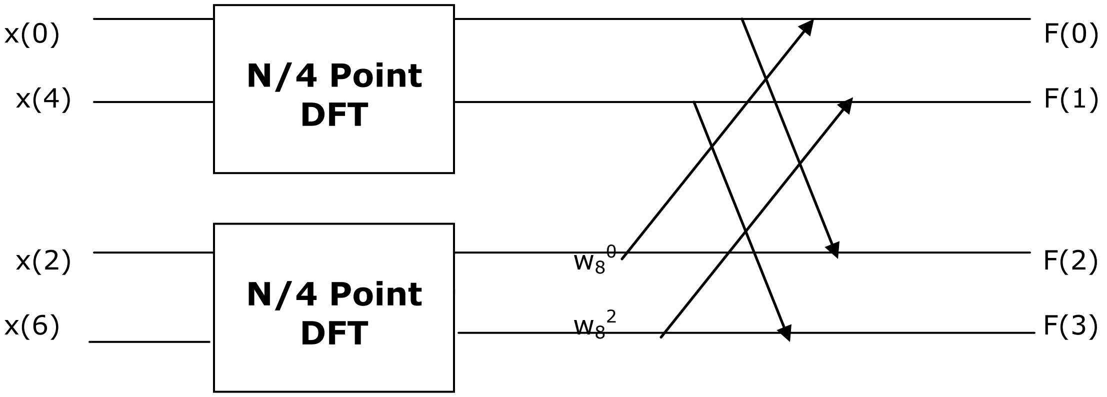


Fig 3. SECOND STAGE FOR FFT COMPUTATION FOR N=8

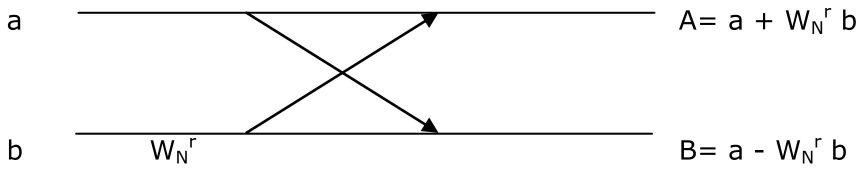


Fig 4. BUTTERFLY COMPUTATION (THIRD STAGE)

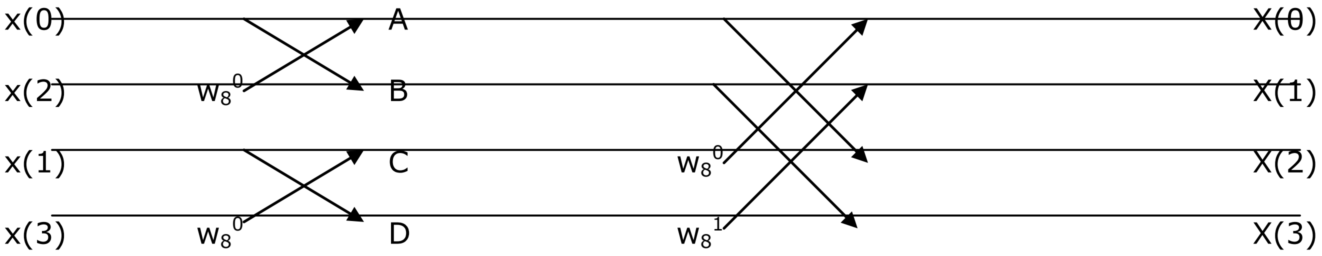


Fig 5. SIGNAL FLOW GRAPH FOR RADIX- DIT FFT N=4

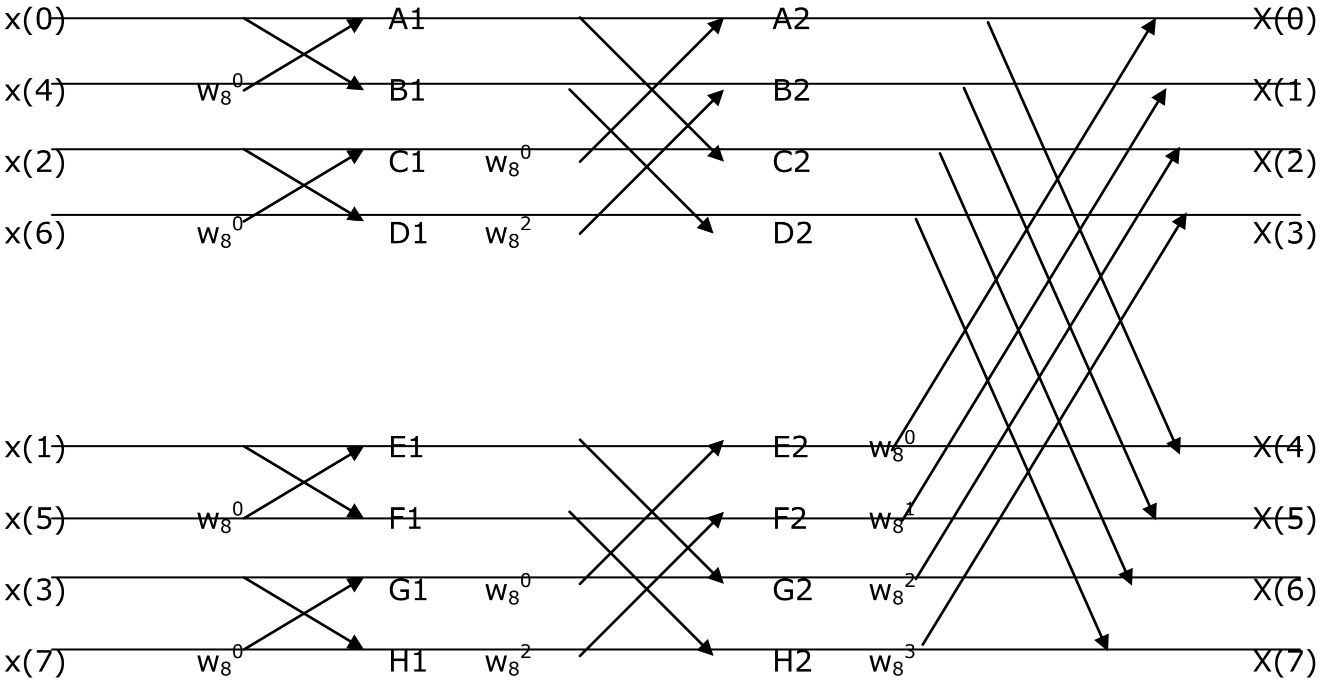
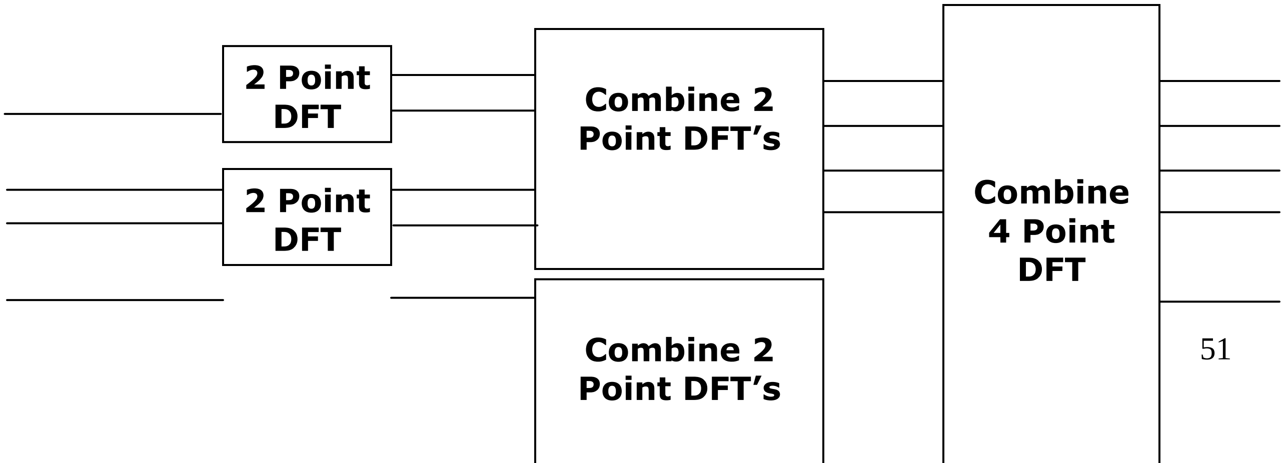


Fig 6. SIGNAL FLOW GRAPH FOR RADIX- DIT FFT N=8



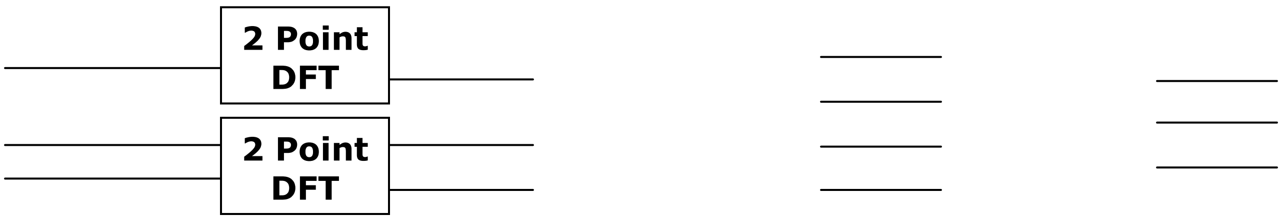


Fig 7. BLOCK DIAGRAM FOR RADIX- DIT FFT N=8

COMPUTATIONAL COMPLEXITY → FFT V/S DIRECT COMPUTATION

For Radix-2 algorithm value of N is given as $N= 2^V$
Hence value of v is calculated as

$$V= \log_{10} N / \log_{10} 2$$
$$= \log_2 N$$

Thus if value of N is 8 then the value of v=3. Thus three stages of decimation. Total number of butterflies will be Nv/2 = 12.
If value of N is 16 then the value of v=4. Thus four stages of decimation. Total number of butterflies will be Nv/2 = 32.

Each butterfly operation takes two addition and one multiplication operations. Direct computation requires N^2 multiplication operation & N^2 – N addition operations.

N	**Direct computation**		**DIT FFT algorithm**		**Improvement in processing speed for multiplication**
	Complex Multiplication N^2	Complex Addition N^2 - N	Complex Multiplication $N/2 \log_2 N$	Complex Addition $N \log_2 N$	
8	64	52	12	24	5.3 times
16	256	240	32	64	8 times
256	65536	65280	1024	2048	64 times

MEMORY REQUIREMENTS AND IN PLACE COMPUTATION

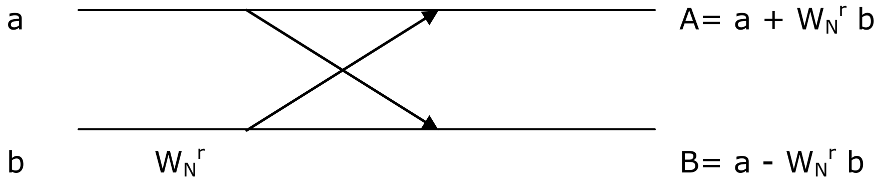


Fig. BUTTERFLY COMPUTATION

From values a and b new values A and B are computed. Once A and B are computed, there is no need to store a and b. Thus same memory locations can be used to store A and B where a and b were stored hence called as In place computation. The advantage of in place computation is that it reduces memory requirement.

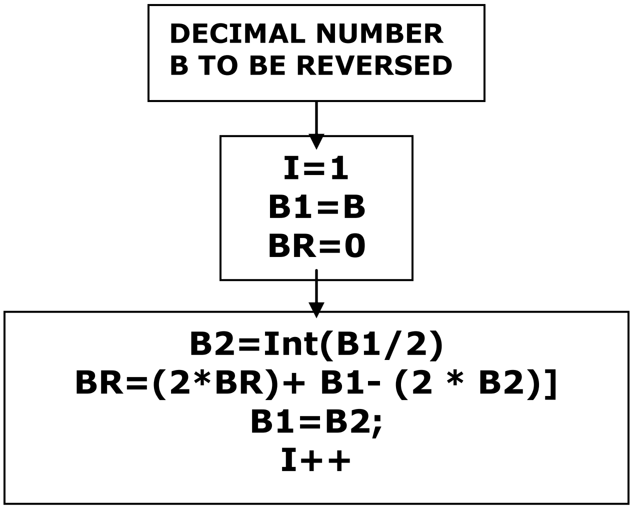
Thus for computation of one butterfly, four memory locations are required for storing two complex numbers A and B. In every stage there are N/2 butterflies hence total 2N memory locations are required. 2N locations are required for each stage. Since stages are computed successively these memory locations can be shared. In every stage N/2 twiddle factors are required hence maximum storage requirements of N point DFT will be (2N + N/2).

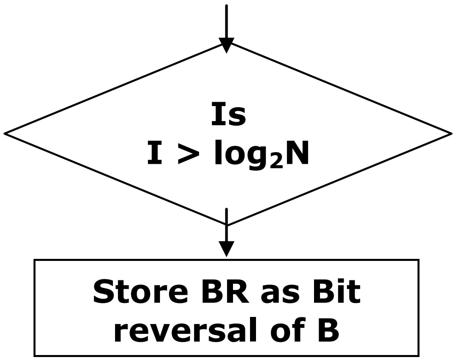
BIT REVERSAL

For 8 point DIT DFT input data sequence is written as x(0), x(4), x(2), x(6), x(1), x(5), x(3), x(7) and the DFT sequence X(k) is in proper order as X(0), X(1), X(2), X(3), X(4), x(5), X(6), x(7). In DIF FFT it is exactly opposite. This can be obtained by bit reversal method.

Decimal	**Memory Address x(n) in binary (Natural Order)**			**Memory Address in bit reversed order**			**New Address in decimal**
0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	4
2	0	1	0	0	1	0	2
3	0	1	1	1	1	0	6
4	1	0	0	0	0	1	1
5	1	0	1	1	0	1	5
6	1	1	0	0	1	1	3
7	1	1	1	1	1	1	7

Table shows first column of memory address in decimal and second column as binary. Third column indicates bit reverse values. As FFT is to be implemented on digital computer simple integer division by 2 method is used for implementing bit reversal algorithms. Flow chart for Bit reversal algorithm is as follows





2. DECIMATION IN FREQUENCY (DIFFFT)

In DIF N Point DFT is splitted into N/2 points DFT's. X(k) is splitted with k even and k odd this is called Decimation in frequency(DIF FFT).

N point DFT is given as

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{kn} \qquad (1)$$

Since the sequence x(n) is splitted N/2 point samples, thus

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

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

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

$$X(k) = \sum_{m=0}^{N/2-1} \left[x(n) + (-1)^k\, x(n + N/2) \quad W_N^{kn} \right] \qquad (3)$$

Let us split X(k) into even and odd numbered samples

$$X(2k) = \sum_{m=0}^{N/2-1} \left[x(n) + (-1)^{2k}\, x(n + N/2)\, W_N^{2kn} \right] \qquad (4)$$

$$X(2k+1) = \sum_{m=0}^{N/2-1} \left[x(n)+(-1)^{(2k+1)} x(n + N/2)W_N^{(2k+1)n} \right] \qquad (5)$$

Equation (4) and (5) are thus simplified as

$$g1(n) = x(n) + x(n + N/2)$$
$$g2(n) = x(n) - x(n + N/2)\, W_N^{n}$$

Fig 1 shows **Butterfly computation** in DIF FFT.

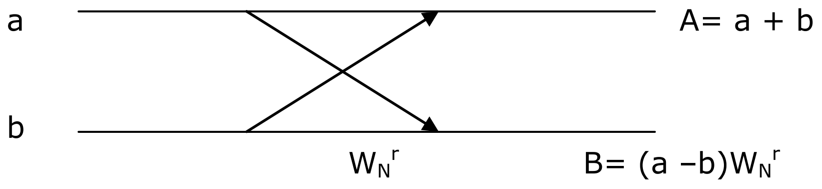


Fig 1. BUTTERFLY COMPUTATION

Fig 2 shows signal flow graph and stages for computation of radix-2 DIF FFT algorithm of N=4

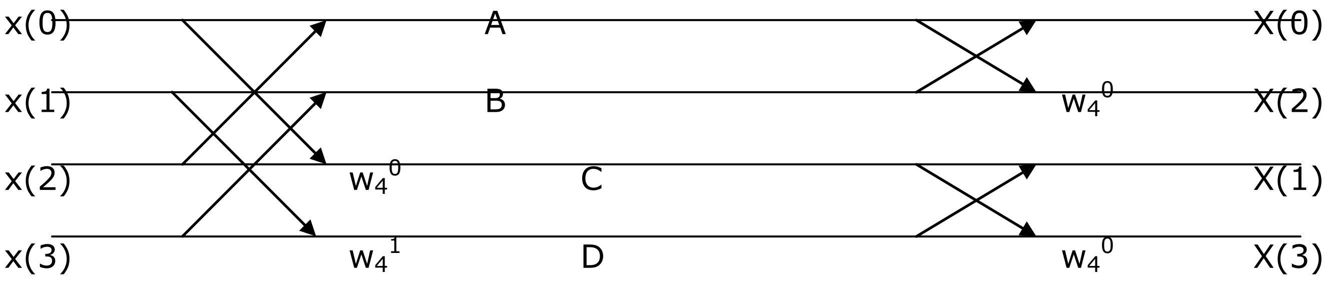
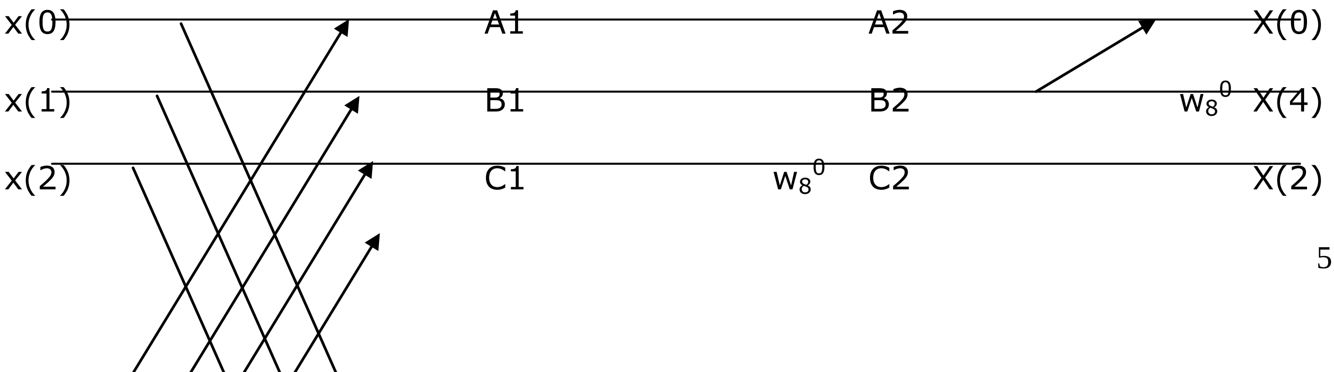


Fig 2. SIGNAL FLOW GRAPH FOR RADIX- DIF FFT N=4

Fig 3 shows signal flow graph and stages for computation of radix-2 DIF FFT algorithm of N=8



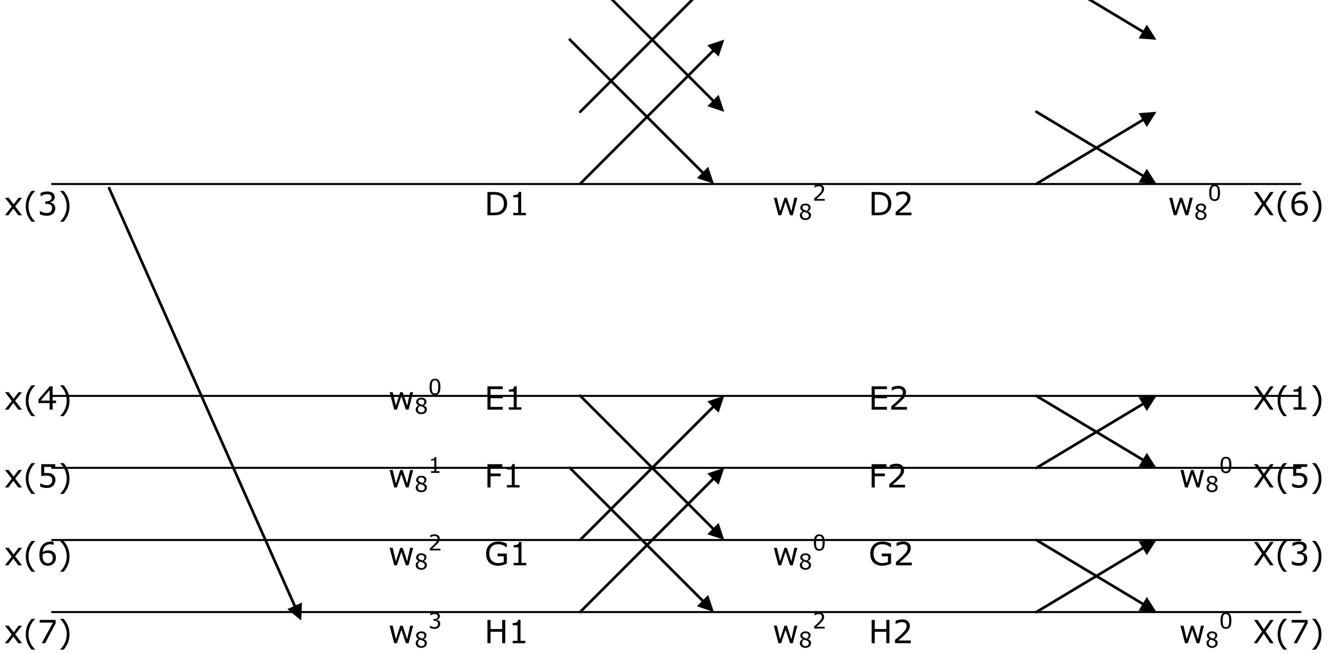


Fig 3. SIGNAL FLOW GRAPH FOR RADIX- DIF FFT N=8

DIFFERENCE BETWEEN DITFFT AND DIFFFT

Sr No	DIT FFT	DIF FFT
1	DITFFT algorithms are based upon decomposition of the input sequence into smaller and smaller sub sequences.	DIFFFT algorithms are based upon decomposition of the output sequence into smaller and smaller sub sequences.
2	In this input sequence x(n) is splitted into even and odd numbered samples	In this output sequence X(k) is considered to be splitted into even and odd numbered samples
3	Splitting operation is done on time domain sequence.	Splitting operation is done on frequency domain sequence.
4	In DIT FFT input sequence is in bit reversed order while the output sequence is in natural order.	In DIFFFT, input sequence is in natural order. And DFT should be read in bit reversed order.

DIFFERENCE BETWEEN DIRECT COMPUTATION & FFT

Sr No	Direct Computation	Radix -2 FFT Algorithms
1	Direct computation requires large number of computations as compared with FFT algorithms.	Radix-2 FFT algorithms requires less number of computations.
2	Processing time is more and more for large number of N hence processor remains busy.	Processing time is less hence these algorithms compute DFT very quickly as compared with direct computation.

3	Direct computation does not requires splitting operation.	Splitting operation is done on time domain basis (DIT) or frequency domain basis (DIF)
4	As the value of N in DFT increases, the efficiency of direct computation decreases.	As the value of N in DFT increases, the efficiency of FFT algorithms increases.
5	In those applications where DFT is to be computed only at selected values of k(frequencies) and when these values are less than $\log_2 N$ then direct computation becomes more efficient than FFT.	Applications 1) Linear filtering 2) Digital filter design

Q) x(n)={1,2,2,1} Find X(k) using DITFFT.
Q) x(n)={1,2,2,1} Find X(k) using DIFFFT.
Q) x(n)={0.3535,0.3535,0.6464,1.0607,0.3535,-1.0607,-1.3535,-0.3535} Find X(k) using DITFFT.
Q) Using radix 2 FFT algorithm, plot flow graph for N=8.

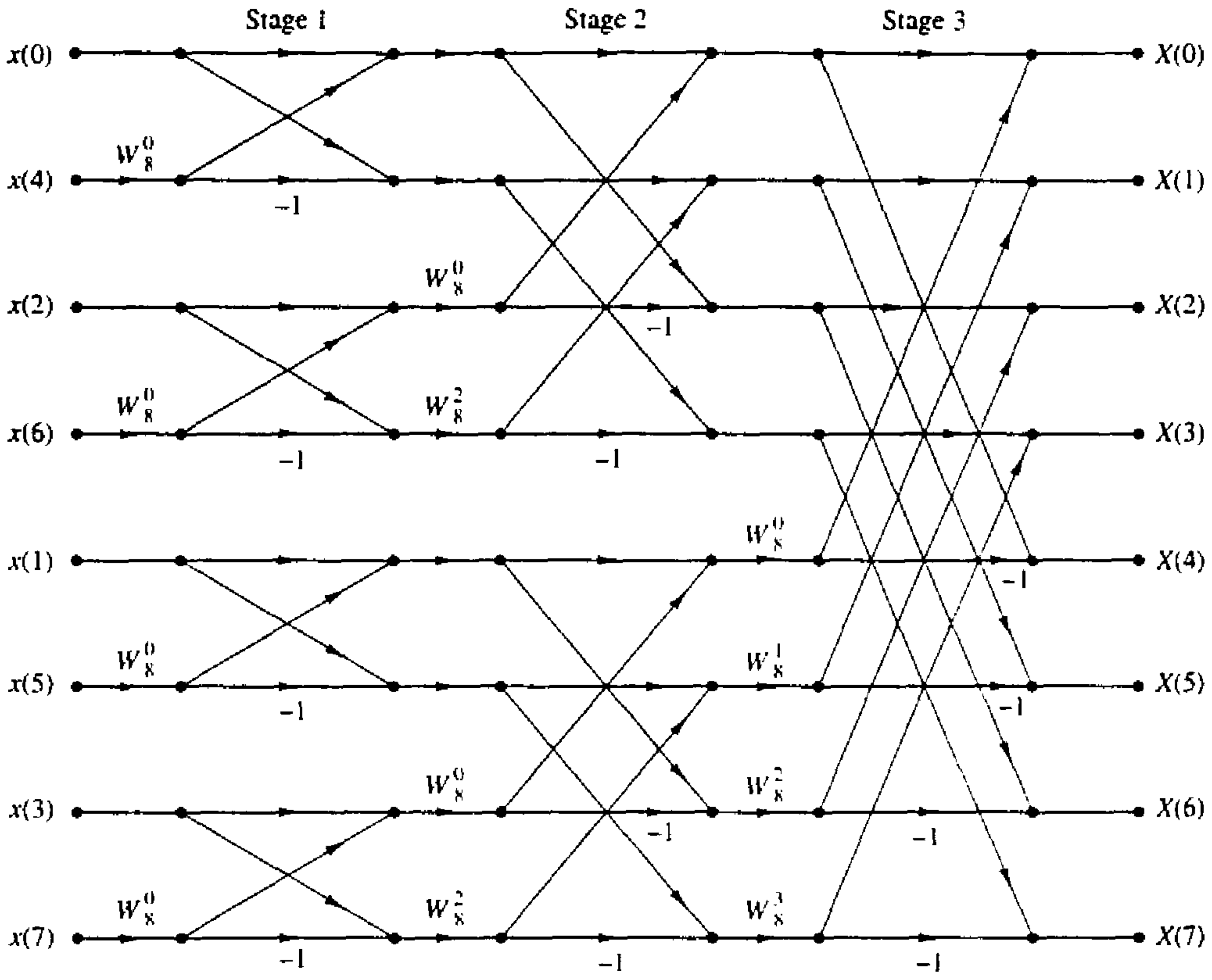


Figure 6.6 Eight-point decimation-in-time FFT algorithm.

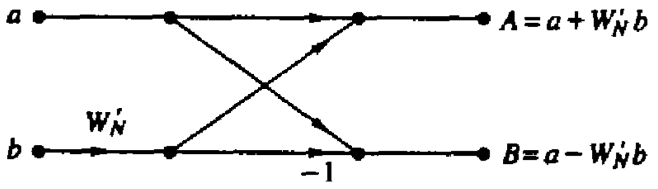
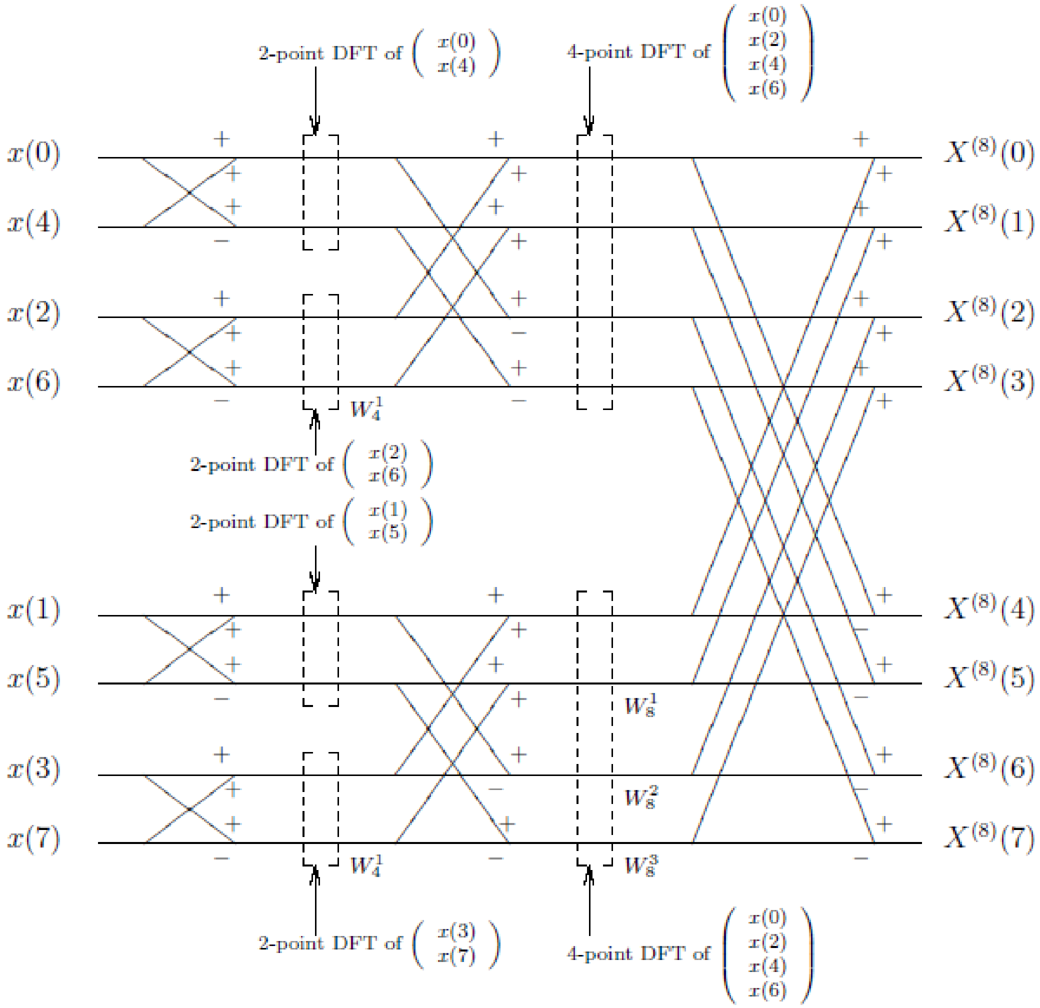


Figure 6.7 Basic butterfly computation in the decimation-in-time FFT algorithm.


2-point DFT of $\begin{pmatrix} x(0) \\ x(4) \end{pmatrix}$
4-point DFT of $\begin{pmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \end{pmatrix}$
$x(0)$
$x(4)$
$x(2)$
$x(6)$
$x(1)$
$x(5)$
$x(3)$
$x(7)$
W_4^1
2-point DFT of $\begin{pmatrix} x(2) \\ x(6) \end{pmatrix}$
2-point DFT of $\begin{pmatrix} x(1) \\ x(5) \end{pmatrix}$
W_8^1
W_8^2
W_4^1
W_8^3
2-point DFT of $\begin{pmatrix} x(3) \\ x(7) \end{pmatrix}$
4-point DFT of $\begin{pmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \end{pmatrix}$
$X^{(8)}(0)$
$X^{(8)}(1)$
$X^{(8)}(2)$
$X^{(8)}(3)$
$X^{(8)}(4)$
$X^{(8)}(5)$
$X^{(8)}(6)$
$X^{(8)}(7)$

$$
\begin{aligned}
W_2 &= e^{-j\frac{2\pi}{2}} = -1, \\
W_4 &= e^{-j\frac{2\pi}{4}} = -j, \\
W_8 &= e^{-j\frac{2\pi}{8}}. \quad \blacksquare
\end{aligned}
$$

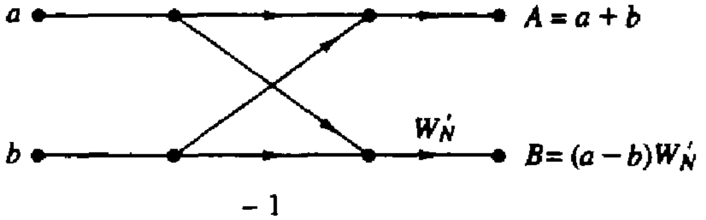


Figure 6.10 Basic butterfly computation in the decimation-in-frequency FFT algorithm.

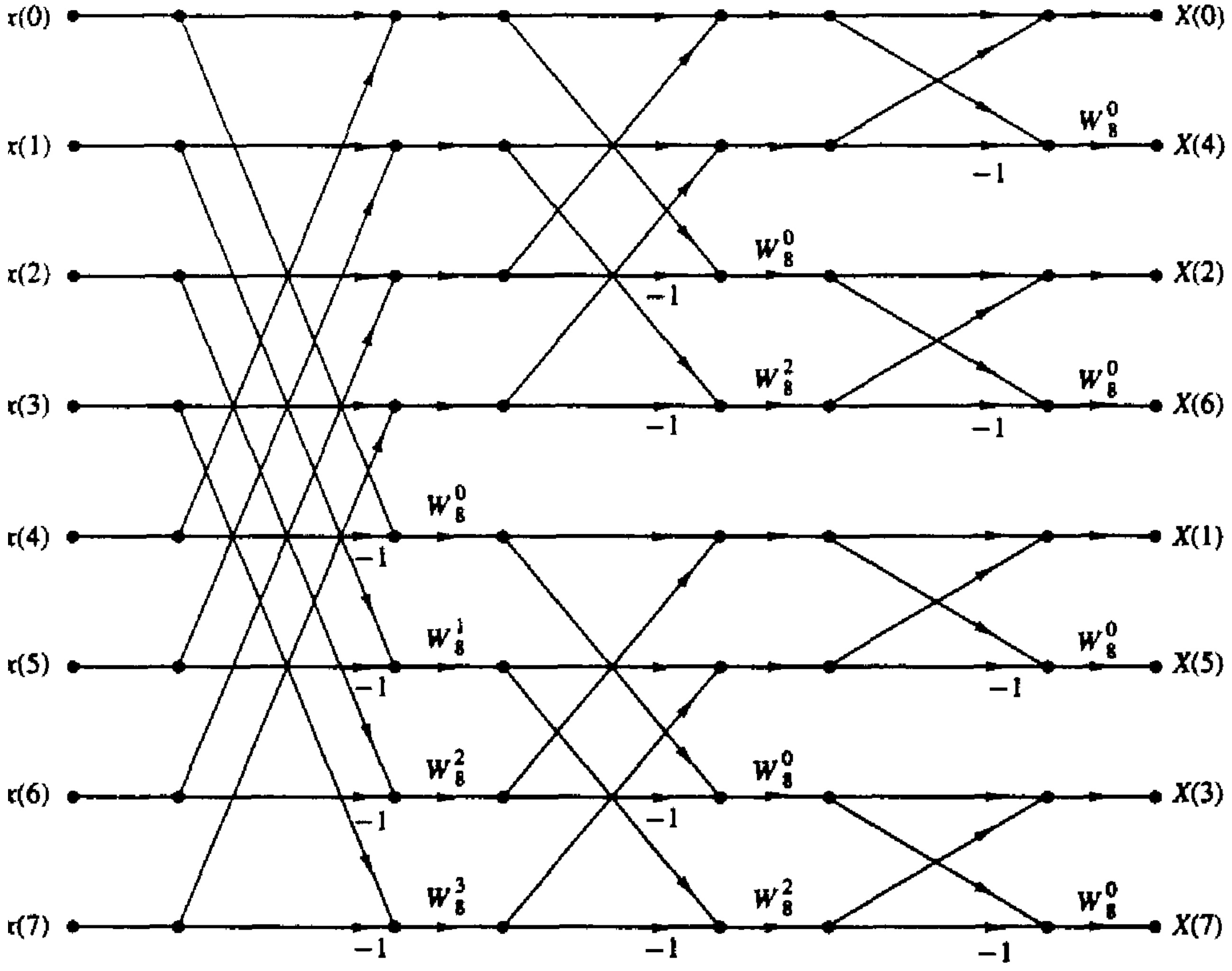


Figure 6.11 N = 8-point decimation-in-frequency FFT algorithmn.

GOERTZEL ALGORITHM

FFT algorithms are used to compute N point DFT for N samples of the sequence x(n). This requires N/2 $\log_2$N number of complex multiplications and N $\log_2$N complex additions. In some applications DFT is to be computed only at selected values of frequencies and selected values are less than $\log_2$N, then direct computations of DFT becomes more efficient than FFT. This direct computations of DFT can be realized through linear filtering of x(n). Such linear filtering for computation of DFT can be implemented using Goertzel algorithm.

By definition N point DFT is given as

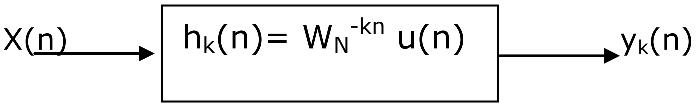
$$X(k) = \sum_{m=0}^{N-1} x(m)\, W_N^{km} \qquad (1)$$

Multiplying both sides by W_N^{-kN} (which is always equal to 1).

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

Thus for LSI system which has input x(n) and having unit sample response

$$h_k(n) = W_N^{-kn}\, u(n)$$



Linear convolution is given by

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

$$y_k(n) = \sum_{m=-\infty}^{\infty} x(m)\, W_N^{-k(n-m)}\ u(n-m) \qquad (3)$$

As x(m) is given for N values

$$y_k(n) = \sum_{m=0}^{N-1} x(m)\, W_N^{-k(n-m)} \qquad (4)$$

The output of LSI system at n=N is given by

$$y_k(n)|_{n=N} = \sum_{m=-\infty}^{\infty} x(m)\, W_N^{-k(N-m)} \qquad (5)$$

Thus comparing equation (2) and (5),

$$X(k) = y_k(n)|_{n=N}$$

Thus DFT can be obtained as the output of LSI system at n=N. Such systems can give X(k) at selected values of k. Thus DFT is computed as linear filtering operations by Goertzel Algorithm.

UNIT III
IIR FILTER DESIGN

3.1 INTRODUCTION

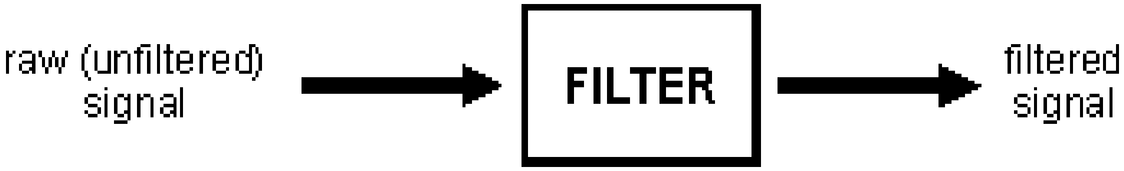
To remove or to reduce strength of unwanted signal like noise and to improve the quality of required signal filtering process is used. To use the channel full bandwidth we mix up two or more signals on transmission side and on receiver side we would like to separate it out in efficient way. Hence filters are used. Thus the digital filters are mostly used in

1. Removal of undesirable noise from the desired signals
2. Equalization of communication channels
3. Signal detection in radar, sonar and communication
4. Performing spectral analysis of signals.

Analog and digital filters

In signal processing, the function of a filter is to remove unwanted parts of the signal, such as random noise, or to extract useful parts of the signal, such as the components lying within a certain frequency range.

The following block diagram illustrates the basic idea.



There are two main kinds of filter, *analog* and *digital*. They are quite different in their physical makeup and in how they work.

An analog filter uses analog electronic circuits made up from components such as resistors, capacitors and op amps to produce the required filtering effect. Such filter circuits are widely used in such applications as noise reduction, video signal enhancement, graphic equalizers in hi-fi systems, and many other areas.

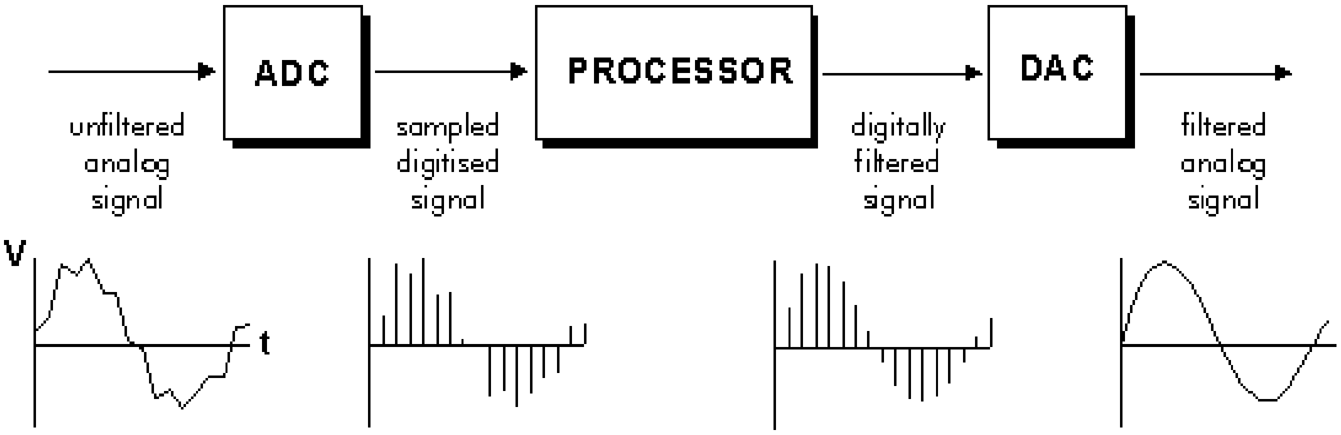
In analog filters the signal being filtered is an electrical voltage or current which is the direct analogue of the physical quantity (e.g. a sound or video signal or transducer output) involved.

A digital filter uses a digital processor to perform numerical calculations on sampled values of the signal. The processor may be a general-purpose computer such as a PC, or a specialized DSP (Digital Signal Processor) chip.

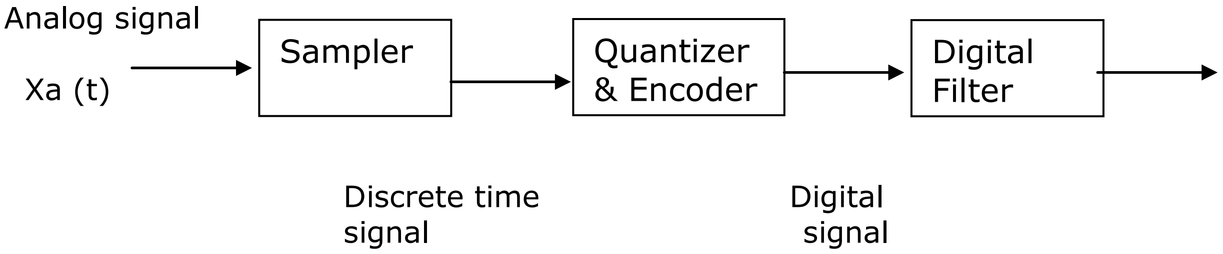
The analog input signal must first be sampled and digitized using an ADC (analog to digital converter). The resulting binary numbers, representing successive sampled values of the input signal, are transferred to the processor, which carries out numerical calculations on them. These calculations typically involve multiplying the input values by constants and adding the products together. If necessary, the results of these calculations, which now represent sampled values of the filtered signal, are output through a DAC (digital to analog converter) to convert the signal back to analog form.

In a digital filter, the signal is represented by a sequence of numbers, rather than a voltage or current.

The following diagram shows the basic setup of such a system.



BASIC BLOCK DIAGRAM OF DIGITAL FILTERS



1. Samplers are used for converting continuous time signal into a discrete time signal by taking samples of the continuous time signal at discrete time instants.

2. The Quantizer are used for converting a discrete time continuous amplitude signal into a digital signal by expressing each sample value as a finite number of digits.

3. In the encoding operation, the quantization sample value is converted to the binary equivalent of that quantization level.

4. The digital filters are the discrete time systems used for filtering of sequences. These digital filters performs the frequency related operations such as low pass, high pass, band pass and band reject etc. These digital Filters are designed with digital hardware and software and are represented by difference equation.

DIFFERENCE BETWEEN ANALOG FILTER AND DIGITAL FILTER

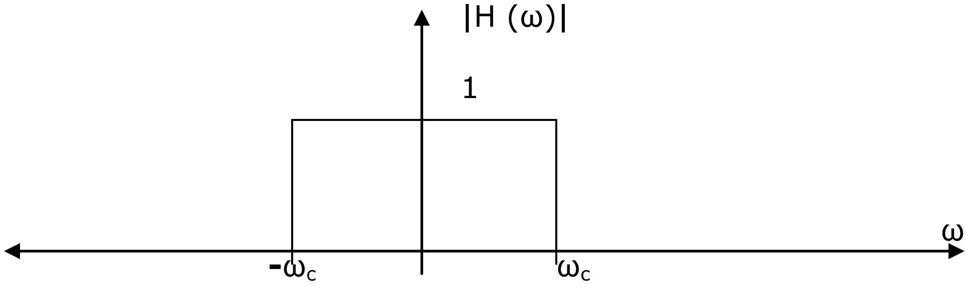
Sr No	Analog Filter	Digital Filter
1	Analog filters are used for filtering analog signals.	Digital filters are used for filtering digital sequences.
2	Analog filters are designed with various components like resistor, inductor and capacitor	Digital Filters are designed with digital hardware like FF, counters shift registers, ALU and software's like C or assembly language.
3	Analog filters less accurate & because of component tolerance of active components & more sensitive to environmental changes.	Digital filters are less sensitive to the environmental changes, noise and disturbances. Thus periodic calibration can be avoided. Also they are extremely stable.
4	Less flexible	These are most flexible as software programs & control programs can be easily modified. Several input signals can be filtered by one digital filter.
5	Filter representation is in terms of system components.	Digital filters are represented by the difference equation.
6	An analog filter can only be changed by redesigning the filter circuit.	A digital filter is programmable, i.e. its operation is determined by a program stored in the processor's memory. This means the digital filter can easily be changed without affecting the circuitry (hardware).

FILTER TYPES AND IDEAL FILTER CHARACTERISTIC

Filters are usually classified according to their frequency-domain characteristic as lowpass, highpass, bandpass and bandstop filters.

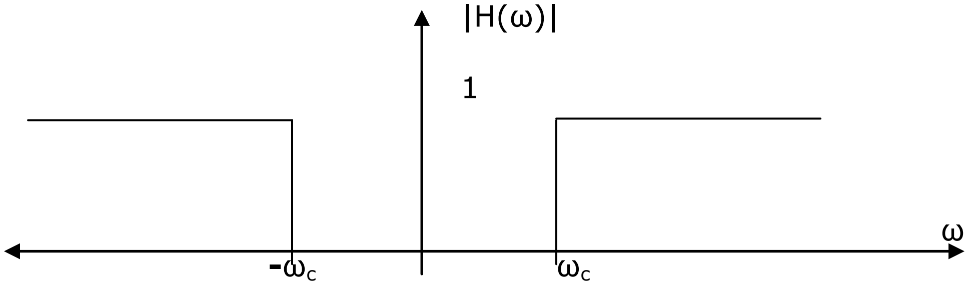
1. Lowpass Filter

A lowpass filter is made up of a passband and a stopband, where the lower frequencies Of the input signal are passed through while the higher frequencies are attenuated.



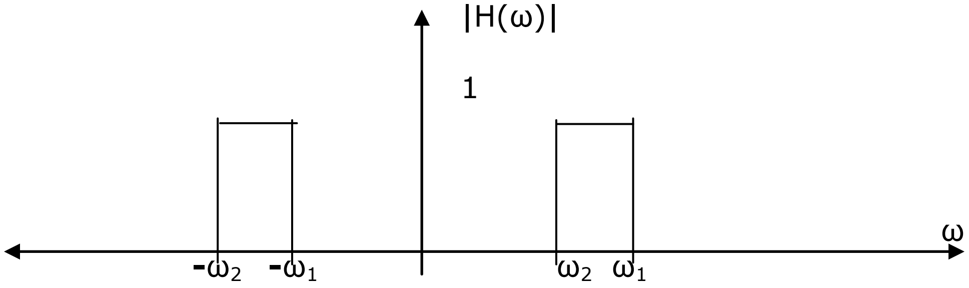
2. Highpass Filter

A highpass filter is made up of a stopband and a passband where the lower frequencies of the input signal are attenuated while the higher frequencies are passed.



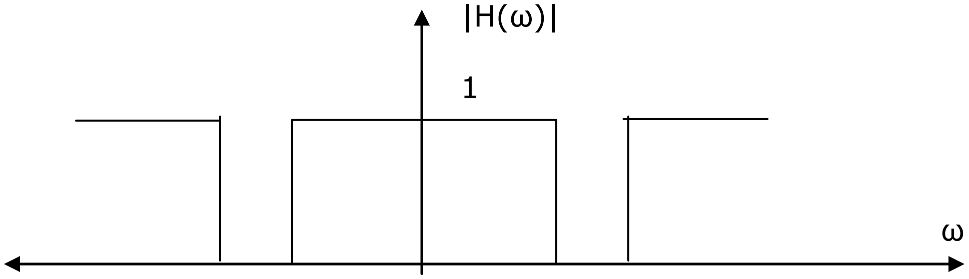
3. Bandpass Filter

A bandpass filter is made up of two stopbands and one passband so that the lower and higher frequencies of the input signal are attenuated while the intervening frequencies are passed.



4. Bandstop Filter

A bandstop filter is made up of two passbands and one stopband so that the lower and higher frequencies of the input signal are passed while the intervening frequencies are attenuated. An idealized bandstop filter frequency response has the following shape.



5. Multipass Filter

A multipass filter begins with a stopband followed by more than one passband. By default, a multipass filter in Digital Filter Designer consists of three passbands and four stopbands. The frequencies of the input signal at the stopbands are attenuated while those at the passbands are passed.

6. Multistop Filter

A multistop filter begins with a passband followed by more than one stopband. By default, a multistop filter in Digital Filter Designer consists of three passbands and two stopbands.

7. All Pass Filter

An all pass filter is defined as a system that has a constant magnitude response for all frequencies.

$$|H(\omega)| = 1 \quad \text{for} \quad 0 \le \omega < \pi$$

The simplest example of an all pass filter is a pure delay system with system function $H(z) = Z^{-k}$. This is a low pass filter that has a linear phase characteristic.

All Pass filters find application as phase equalizers. When placed in cascade with a system that has an undesired phase response, a phase equalizers is designed to compensate for the poor phase characteristic of the system and therefore to produce an overall linear phase response.

IDEAL FILTER CHARACTERISTIC

1. Ideal filters have a constant gain (usually taken as unity gain) passband characteristic and zero gain in their stop band.
2. Ideal filters have a linear phase characteristic within their passband.
3. Ideal filters also have constant magnitude characteristic.
4. Ideal filters are physically unrealizable.

3.2 TYPES OF DIGITAL FILTER

Digital filters are of two types. Finite Impulse Response Digital Filter & Infinite Impulse Response Digital Filter

DIFFERENCE BETWEEN FIR FILTER AND IIR FILTER

Sr No	**FIR Digital Filter**	**IIR Digital Filter**
1	FIR system has finite duration unit sample response. i.e $h(n) = 0$ for $n<0$ and $n \ge M$ Thus the unit sample response exists for the duration from 0 to M-1.	IIR system has infinite duration unit sample response. i. e $h(n) = 0$ for $n<0$ Thus the unit sample response exists for the duration from 0 to ∞.
2	FIR systems are non recursive. Thus output of FIR filter depends upon present and past inputs.	IIR systems are recursive. Thus they use feedback. Thus output of IIR filter depends upon present and past inputs as well as past outputs
3	Difference equation of the LSI system for FIR filters becomes $y(n)=\sum_{k=0}^{M} b_k\, x(n-k)$	Difference equation of the LSI system for IIR filters becomes $y(n)=-\sum_{k=1}^{N} a_k\, y(n-k)+\sum_{k=0}^{M} b_k\, x(n-k)$
4	FIR systems has limited or finite memory requirements.	IIR system requires infinite memory.

5	FIR filters are always stable	Stability cannot be always guaranteed.
6	FIR filters can have an exactly linear phase response so that no phase distortion is introduced in the signal by the filter.	IIR filter is usually more efficient design in terms of computation time and memory requirements. IIR systems usually requires less processing time and storage as compared with FIR.
7	The effect of using finite word length to implement filter, noise and quantization errors are less severe in FIR than in IIR.	Analogue filters can be easily and readily transformed into equivalent IIR digital filter. But same is not possible in FIR because that have no analogue counterpart.
8	All zero filters	Poles as well as zeros are present.
9	FIR filters are generally used if no phase distortion is desired. Example: System described by Y(n) = 0.5 x(n) + 0.5 x(n-1) is FIR filter. h(n)={0.5,0.5}	IIR filters are generally used if sharp cutoff and high throughput is required. Example: System described by Y(n) = y(n-1) + x(n) is IIR filter. h(n)=a^{n} u(n) for n≥0

3. 3 STRUCTURES FOR FIR SYSTEMS

FIR Systems are represented in four different ways

1. Direct Form Structures
2. Cascade Form Structure
3. Frequency-Sampling Structures
4. Lattice structures.

1. DIRECT FORM STRUCTURE OF FIR SYSTEM

The convolution of h(n) and x(n) for FIR systems can be written as

$$y(n)=\sum_{k=0}^{M-1} h(k)\, x(n-k) \qquad (1)$$

The above equation can be expanded as,

$$Y(n)= h(0)\, x(n) + h(1)\, x(n-1) + h(2)\, x(n-2) + \ldots\ldots\ldots\ldots\ldots + h(M-1)\, x(n-M+1) \qquad (2)$$

Implementation of direct form structure of FIR filter is based upon the above equation.

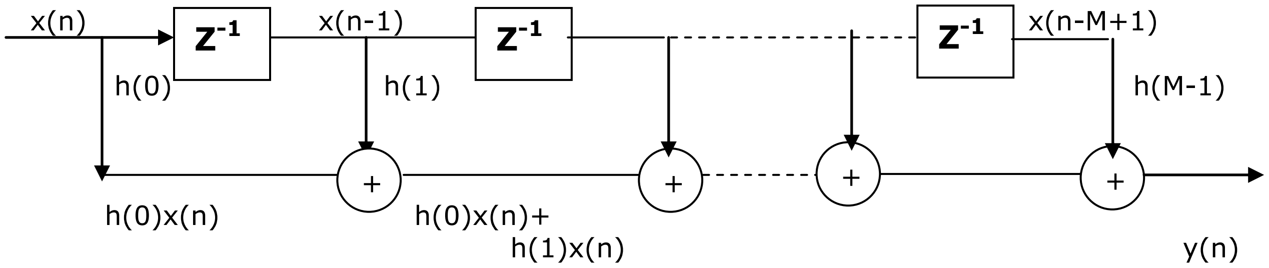


FIG - DIRECT FORM REALIZATION OF FIR SYSTEM

1) There are M-1 unit delay blocks. One unit delay block requires one memory location. Hence direct form structure requires M-1 memory locations.

2) The multiplication of h(k) and x(n-k) is performed for 0 to M-1 terms. Hence M multiplications and M-1 additions are required.

3) Direct form structure is often called as transversal or tapped delay line filter.

2. CASCADE FORM STRUCTURE OF FIR SYSTEM

In cascade form, stages are cascaded (connected) in series. The output of one system is input to another. Thus total K number of stages are cascaded. The total system function 'H' is given by

$$H= H_1(z) . H_2(z)\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. H_k(z) \qquad (1)$$
$$H= Y_1(z)/X_1(z). \; Y_2(z)/X_2(z). \ldots\ldots\ldots\ldots Y_k(z)/X_k(z) \qquad (2)$$

$$H(z)=\prod_{k=1}^{k} H_k(z) \qquad (3)$$

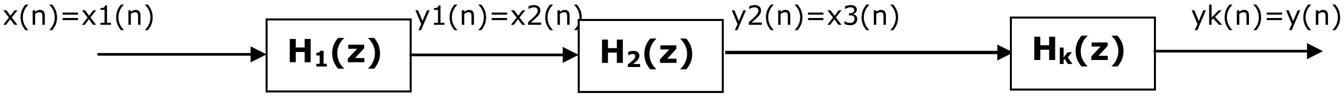


FIG- CASCADE FORM REALIZATION OF FIR SYSTEM

Each H1(z), H2(z)… etc is a second order section and it is realized by the direct form as shown in below figure.

System function for FIR systems

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

Expanding the above terms we have

$$H(z)= H_1(z) . H_2(z)\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. H_k(z)$$

where $H_K(z) = b_{k0} + b_{k1} z^{-1} + b_{k2} z^{-2}$ (2)

Thus Direct form of second order system is shown as

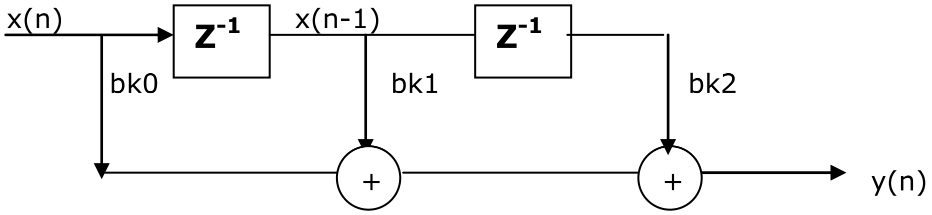


FIG - DIRECT FORM REALIZATION OF FIR SECOND ORDER SYSTEM

3. 4 STRUCTURES FOR IIR SYSTEMS

IIR Systems are represented in four different ways

1. Direct Form Structures Form I and Form II
2. Cascade Form Structure
3. Parallel Form Structure
4. Lattice and Lattice-Ladder structure.

DIRECT FORM STRUCTURE FOR IIR SYSTEMS

IIR systems can be described by a generalized equations as

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

Z transform is given as

$$H(z) = \sum_{K=0}^{M} b_k\, z^{-k} / 1+ \sum_{k=1}^{N} a_k\, z^{-k} \qquad (2)$$

Here $H1(z) = \sum_{K=0}^{M} b_k\, z^{-k}$ And $H2(z) = 1+ \sum_{k=0}^{N} a_k\, z^{-k}$

Overall IIR system can be realized as cascade of two function H1(z) and H2(z). Here H1(z) represents zeros of H(z) and H2(z) represents all poles of H(z).

DIRECT FORM - I

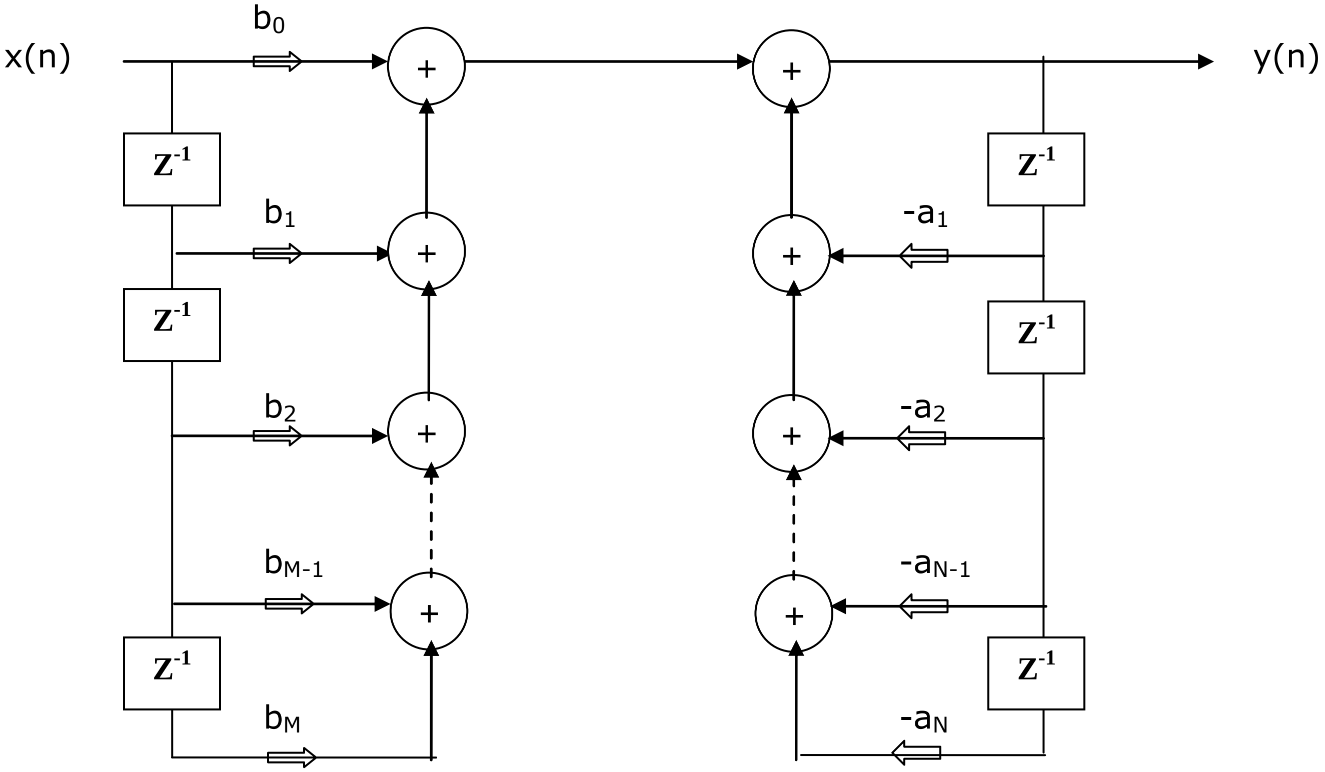


FIG - DIRECT FORM I REALIZATION OF IIR SYSTEM

1. Direct form I realization of H(z) can be obtained by cascading the realization of H1(z) which is all zero system first and then H2(z) which is all pole system.

2. There are M+N-1 unit delay blocks. One unit delay block requires one memory location. Hence direct form structure requires M+N-1 memory locations.

3. Direct Form I realization requires M+N+1 number of multiplications and M+N number of additions and M+N+1 number of memory locations.

DIRECT FORM - II

1. Direct form realization of H(z) can be obtained by cascading the realization of H1(z) which is all pole system and H2(z) which is all zero system.

2. Two delay elements of all pole and all zero system can be merged into single delay element.

3. Direct Form II structure has reduced memory requirement compared to Direct form I structure. Hence it is called canonic form.

4. The direct form II requires same number of multiplications(M+N+1) and additions (M+N) as that of direct form I.

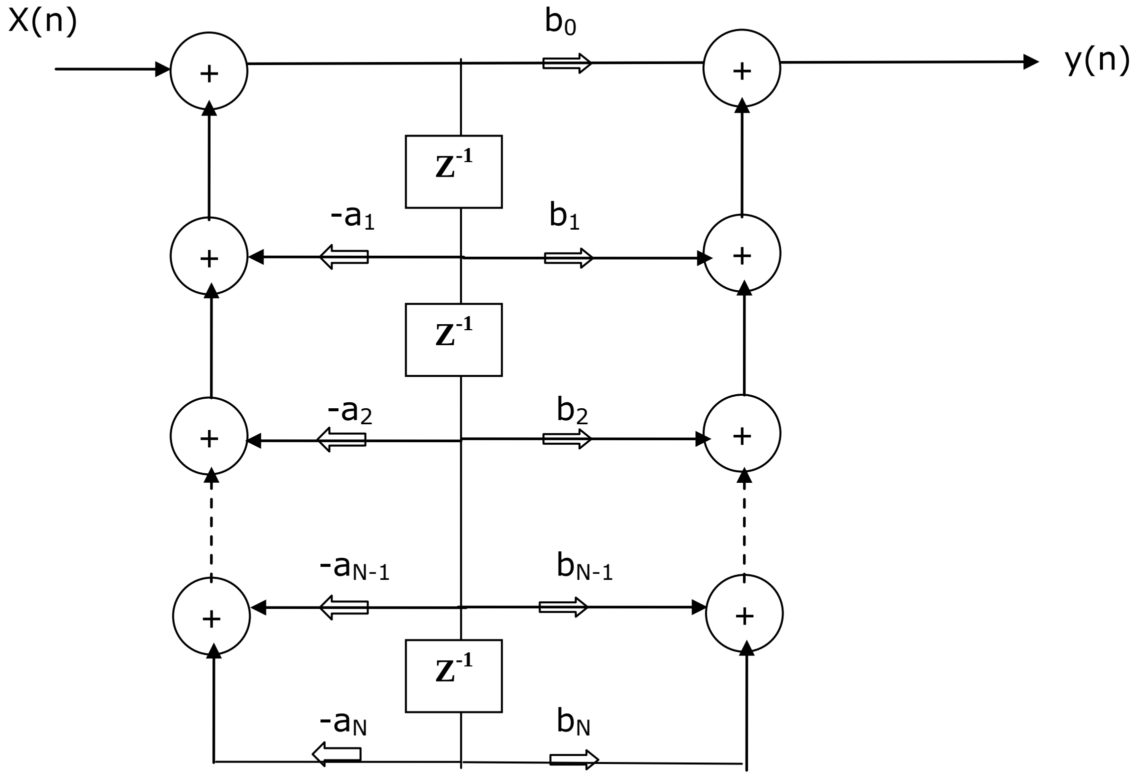


FIG - DIRECT FORM II REALIZATION OF IIR SYSTEM

CASCADE FORM STRUCTURE FOR IIR SYSTEMS

In cascade form, stages are cascaded (connected) in series. The output of one system is input to another. Thus total K number of stages are cascaded. The total system function 'H' is given by

$$H= H_1(z) \cdot H_2(z) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots H_k(z) \qquad (1)$$

$$H= Y_1(z)/X_1(z).\ \ Y_2(z)/X_2(z).\ \ldots\ldots\ldots\ldots\ldots Y_k(z)/X_k(z) \qquad (2)$$

$$H(z)=\prod_{k=1}^{k} H_k(z) \qquad (3)$$

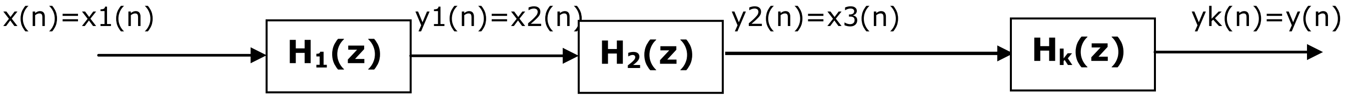


FIG - CASCADE FORM REALIZATION OF IIR SYSTEM

Each H1(z), H2(z)… etc is a second order section and it is realized by the direct form as shown in below figure.

System function for IIR systems

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

Expanding the above terms we have

$$H(z)= H_1(z) \cdot H_2(z) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots H_k(z)$$

$$\text{where } H_K(z) = b_{k0} + b_{k1} z^{-1} + b_{k2} z^{-2} / 1 + a_{k1} z^{-1} + a_{k2} z^{-2} \qquad (2)$$

Thus Direct form of second order IIR system is shown as

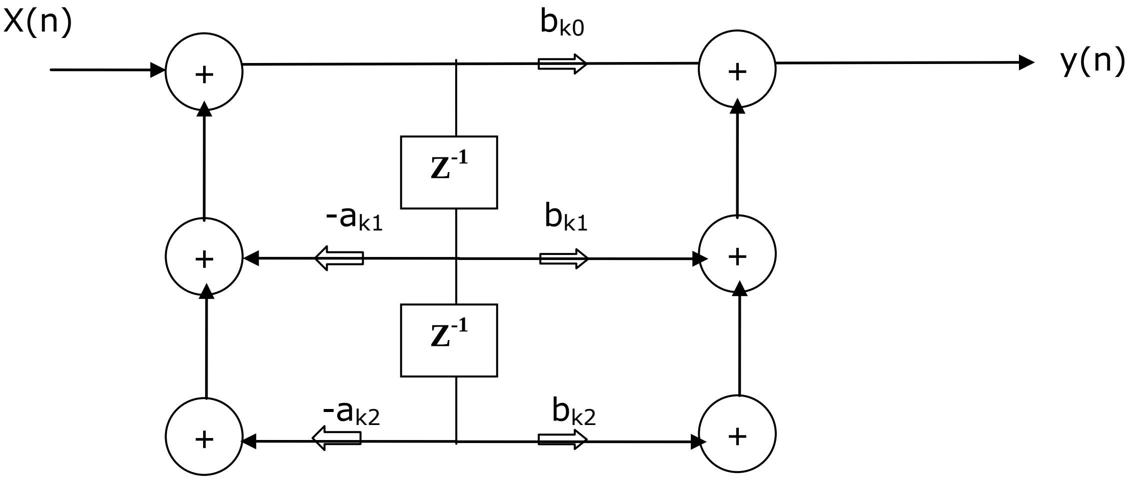


FIG - DIRECT FORM REALIZATION OF IIR SECOND ORDER SYSTEM (CASCADE)

PARALLEL FORM STRUCTURE FOR IIR SYSTEMS

System function for IIR systems is given as

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

$$= b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots\ldots..+ b_M z^{-M} / 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots\ldots + a_N z^{-N} \qquad (2)$$

The above system function can be expanded in partial fraction as follows

$$H(z) = C + H_1(z) + H_2(z)\ldots\ldots\ldots\ldots\ldots\ldots\ldots..+ H_k(z) \qquad (3)$$

Where C is constant and Hk(z) is given as

$$Hk(z) = b_{k0} + b_{k1} z^{-1} / 1 + a_{k1} z^{-1} + a_{k2} z^{-2} \qquad (4)$$

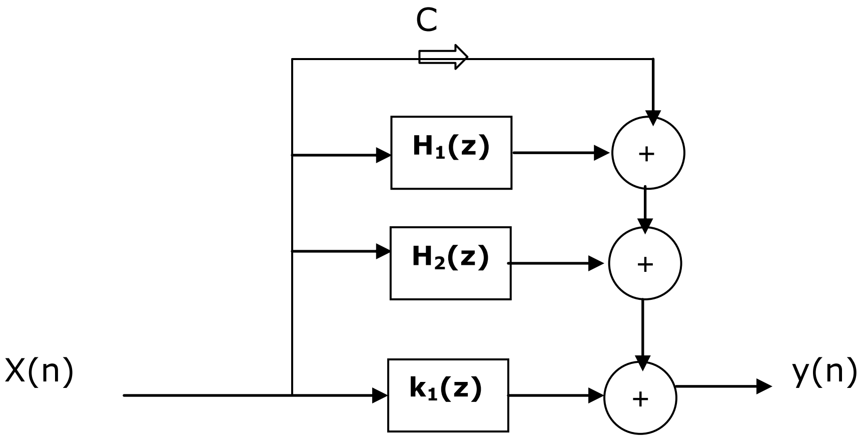


FIG - PARALLEL FORM REALIZATION OF IIR SYSTEM

Thus Direct form of second order IIR system is shown as

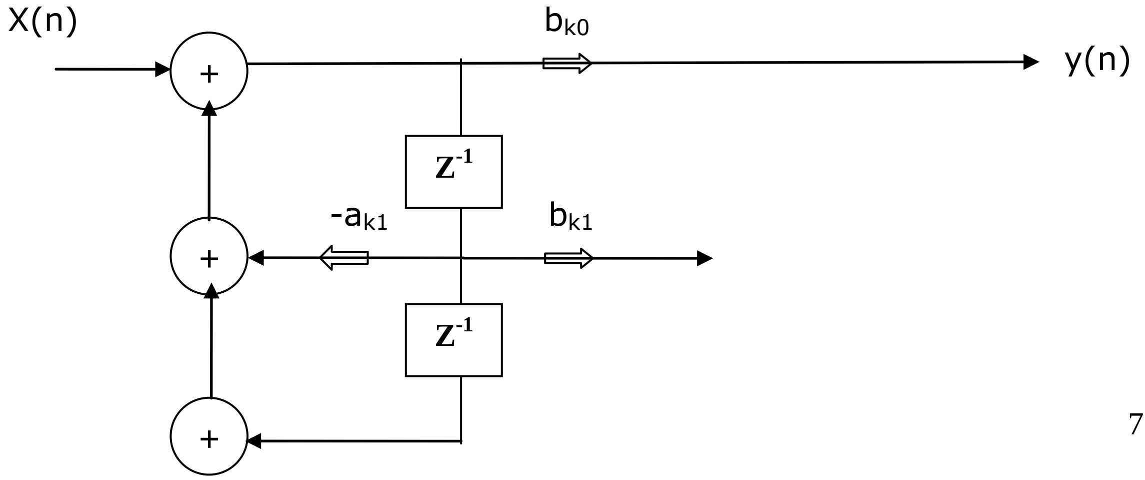


FIG - DIRECT FORM REALIZATION OF IIR SECOND ORDER SYSTEM (PARALLEL)

IIR FILTER DESIGN

1. **IMPULSE INVARIANCE**
2. **BILINEAR TRANSFORMATION**
3. **BUTTERWORTH APPROXIMATION**

4.5 IIR FILTER DESIGN - IMPULSE INVARIANCE METHOD

Impulse Invariance Method is simplest method used for designing IIR Filters.
Important Features of this Method are

1. In impulse variance method, Analog filters are converted into digital filter just by replacing unit sample response of the digital filter by the sampled version of impulse response of analog filter. Sampled signal is obtained by putting t=nT hence

$$h(n) = h_a(nT) \qquad n=0,1,2.\ \ldots\ldots\ldots\ldots$$

   where h(n) is the unit sample response of digital filter and T is sampling interval.
2. But the main disadvantage of this method is that it does not correspond to simple algebraic mapping of S plane to the Z plane. Thus the mapping from analog frequency to digital frequency is many to one. The segments $(2k-1)\Pi/T \le \Omega \le (2k+1)\,\Pi/T$ of $j\Omega$ axis are all mapped on the unit circle $\Pi \le \omega \le \Pi$. This takes place because of sampling.

3. Frequency aliasing is second disadvantage in this method. Because of frequency aliasing, the frequency response of the resulting digital filter will not be identical to the original analog frequency response.

4. Because of these factors, its application is limited to design low frequency filters like LPF or a limited class of band pass filters.

RELATIONSHIP BETWEEN Z PLANE AND S PLANE

Z is represented as $re^{j\omega}$ in polar form and relationship between Z plane and S plane is given as $Z=e^{ST}$ where $s= \sigma + j\,\Omega$.

$$Z= e^{ST} \qquad \text{(Relationship Between Z plane and S plane)}$$
$$Z= e^{(\sigma + j\,\Omega)\,T}$$
$$= e^{\sigma T} \,.\, e^{j\,\Omega\,T}$$

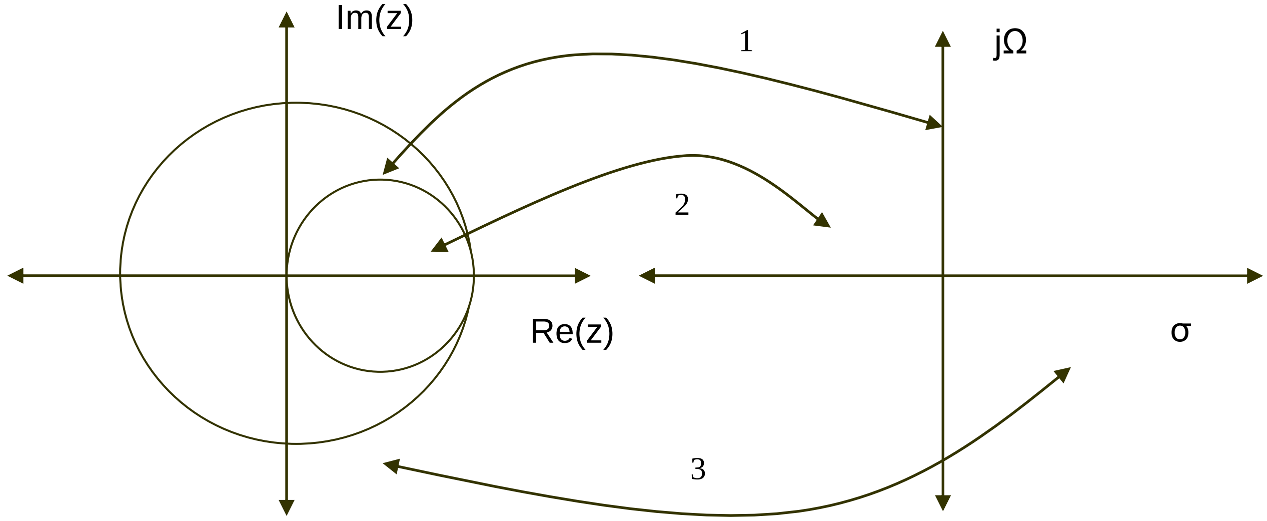
Comparing Z value with the polar form we have.

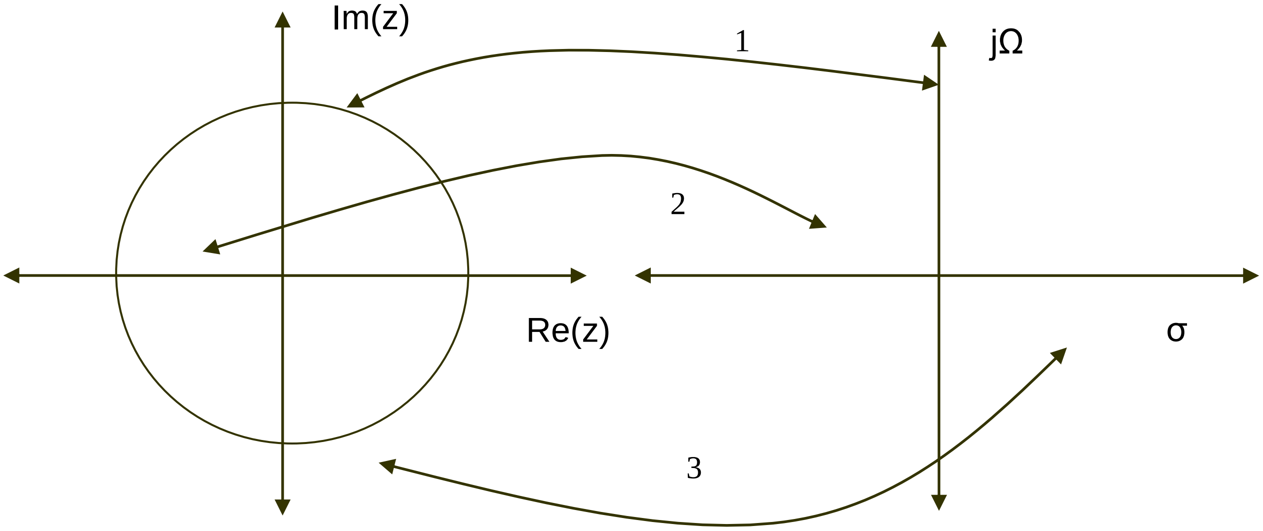
$$\mathbf{r= e^{\sigma T} \text{ and } \omega = \Omega\, T}$$

Here we have three condition
1) If $\sigma = 0$ then $r=1$
2) If $\sigma < 0$ then $0 < r < 1$
3) If $\sigma > 0$ then $r> 1$

Thus

1) Left side of s-plane is mapped inside the unit circle.
2) Right side of s-plane is mapped outside the unit circle.
3) jΩ axis is in s-plane is mapped on the unit circle.





CONVERSION OF ANALOG FILTER INTO DIGITAL FILTER

Let the system function of analog filter is

$$Ha(s)= \sum_{k=1}^{n} C_k / s\text{-}p_k \qquad (1)$$

where pk are the poles of the analog filter and ck are the coefficients of partial fraction expansion. The impulse response of the analog filter ha(t) is obtained by inverse Laplace transform and given as

$$ha(t) = \sum_{k=1}^{n} C_k \; e^{pkt} \qquad (2)$$

The unit sample response of the digital filter is obtained by uniform sampling of ha(t).

$$h(n) = h_a(nT) \qquad n=0,1,2. \ldots\ldots\ldots\ldots$$

$$h(n) =\sum_{k=1}^{n} C_k \; e^{pknT} \qquad (3)$$

System function of digital filter H(z) is obtained by Z transform of h(n).

$$H(z) =\sum_{k=1}^{N} C_k \sum_{n=0}^{\infty} \left[e^{pkT} z^{-1}\right]^{n} \qquad (4)$$

Using the standard relation and comparing equation (1) and (4) system function of digital filter is given as

1 1

$$\frac{}{s - p_k} \leftrightarrow \frac{}{1- e^{pkT} z^{-1}}$$

STANDARD RELATIONS IN IIR DESIGN

Sr No	Analog System Function	Digital System function
1	$\frac{1}{s - a}$	$\frac{1}{1- e^{aT} z^{-1}}$
2	$\frac{s + a}{(s+a)^2 + b^2}$	$\frac{1- e^{-aT} (\cos bT) z^{-1}}{1-2e^{-aT} (\cos bT)z^{-1}+ e^{-2aT}z^{-2}}$
3	$\frac{b}{(s+a)^2 + b^2}$	$\frac{e^{-aT} (\sin bT) z^{-1}}{1-2e^{-aT} (\cos bT)z^{-1}+ e^{-2aT}z^{-2}}$

EXAMPLES - IMPULSE INVARIANCE METHOD

Sr No	Analog System Function	Digital System function
1	$\frac{s + 0.1}{(s+0.1)^2 + 9}$	$\frac{1 - (e^{-0.1T}\cos 3T)z^{-1}}{1-2e^{-0.1T} (\cos 3T)z^{-1}+ e^{-0.2T}z^{-2}}$
2	$\frac{1}{(s+1)(s+2)}$ (for sampling frequency of 5 samples/sec)	$\frac{0.148 z}{z^2 - 1.48 z + 0.548}$
3	$\frac{10}{(s+2)}$ (for sampling time is 0.01 sec)	$\frac{10}{1 - z^{-1}}$

4.6 IIR FILTER DESIGN - BILINEAR TRANSFORMATION METHOD (BZT)

The method of filter design by impulse invariance suffers from aliasing. Hence in order to overcome this drawback Bilinear transformation method is designed. In analogue domain frequency axis is an infinitely long straight line while sampled data z plane it is unit circle radius. The bilinear transformation is the method of squashing the infinite straight analog frequency axis so that it becomes finite.

Important Features of Bilinear Transform Method are

1. Bilinear transformation method (BZT) is a mapping from analog S plane to digital Z plane. This conversion maps analog poles to digital poles and analog zeros to digital zeros. Thus all poles and zeros are mapped.

2. This transformation is basically based on a numerical integration techniques used to simulate an integrator of analog filter.

3. There is one to one correspondence between continuous time and discrete time frequency points. Entire range in Ω is mapped only once into the range -∏≤ω≤∏.

4. Frequency relationship is non-linear. Frequency warping or frequency compression is due to non-linearity. Frequency warping means amplitude response of digital filter is expanded at the lower frequencies and compressed at the higher frequencies in comparison of the analog filter.

5. But the main disadvantage of frequency warping is that it does change the shape of the desired filter frequency response. In particular, it changes the shape of the transition bands.

CONVERSION OF ANALOG FILTER INTO DIGITAL FILTER

Z is represented as $re^{j\omega}$ in polar form and relationship between Z plane and S plane in BZT method is given as

$$S = \frac{2}{T}\,\frac{z - 1}{z + 1}$$

$$S = \frac{2}{T}\,\frac{re^{j\omega} - 1}{re^{j\omega} + 1}$$

$$S = \frac{2}{T}\,\frac{r\,(\cos\omega + j\sin\omega) - 1}{r\,(\cos\omega + j\sin\omega) + 1}$$

$$S = \frac{2}{T}\left[\frac{r^2 - 1}{1 + r^2 + 2r\cos\omega} + \frac{j\,2\,r\sin\omega}{1 + r^2 + 2r\cos\omega}\right]$$

Comparing the above equation with S= σ + j Ω. We have

$$\boldsymbol{\sigma = \frac{2}{T}\,\frac{r^2 - 1}{1 + r^2 + 2r\cos\omega}}$$

$$\boldsymbol{\Omega = \frac{2}{T}\,\frac{2\,r\sin\omega}{1 + r^2 + 2r\cos\omega}}$$

Here we have three condition

1) If $\sigma < 0$ then $0 < r < 1$
2) If $\sigma > 0$ then $r > 1$
3) If $\sigma = 0$ then $r=1$

When r =1

$$\Omega = \underline{2} \qquad \sin\omega$$

$$T \quad \overline{1+\cos\omega}$$

$$\mathbf{\Omega = (2/T)\tan(\omega/2)}$$

$$\omega = 2\tan^{-1}(\Omega T/2)$$

The above equations shows that in BZT frequency relationship is non-linear. The frequency relationship is plotted as

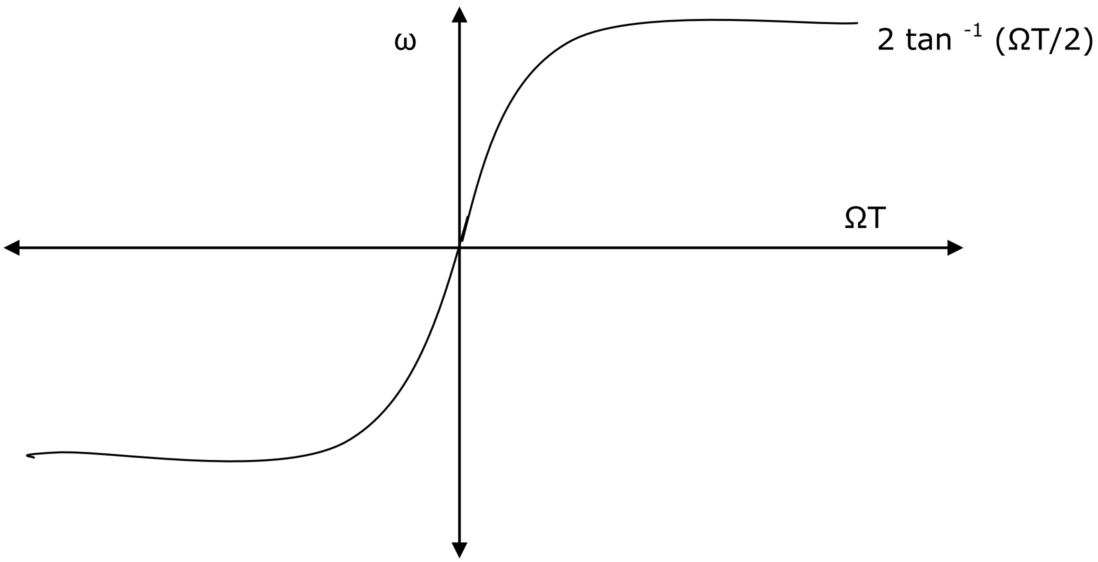


FIG - MAPPING BETWEEN FREQUENCY VARIABLE ω AND Ω IN BZT METHOD.

DIFFERENCE - IMPULSE INVARIANCE Vs BILINEAR TRANSFORMATION

Sr No	Impulse Invariance	Bilinear Transformation
1	In this method IIR filters are designed having a unit sample response h(n) that is sampled version of the impulse response of the analog filter.	This method of IIR filters design is based on the trapezoidal formula for numerical integration.
2	In this method small value of T is selected to minimize the effect of aliasing.	The bilinear transformation is a conformal mapping that transforms the j Ω axis into the unit circle in the z plane only once, thus avoiding aliasing of frequency components.

3	They are generally used for low frequencies like design of IIR LPF and a limited class of bandpass filter	For designing of LPF, HPF and almost all types of Band pass and band stop filters this method is used.
4	Frequency relationship is linear.	Frequency relationship is non-linear. Frequency warping or frequency compression is due to non-linearity.
5	All poles are mapped from the s plane to the z plane by the relationship $Z^k= e^{pkT}$. But the zeros in two domain does not satisfy the same relationship.	All poles and zeros are mapped.

LPF AND HPF ANALOG BUTTERWORTH FILTER TRANSFER FUNCTION

Sr No	Order of the Filter	Low Pass Filter	High Pass Filter
1	1	$1 / s+1$	$s \ / s+1$
2	2	$1 / s^2+ \sqrt{2}\ s + 1$	$s^2 / s^2+ \sqrt{2}\ s + 1$
3	3	$1 / s^3 + 2\ s^2 + 2s +1$	$s^3 / s^3 + 2\ s^2 + 2s +1$

METHOD FOR DESIGNING DIGITAL FILTERS USING BZT

step 1. Find out the value of ω_c^*.

$$\boldsymbol{\omega_c^* = \ (2/T) \tan (\omega_c T_s/2)}$$

step 2. Find out the value of frequency scaled analog transfer function

Normalized analog transfer function is frequency scaled by replacing s by s/ω_p^*.

step 3. Convert into digital filter

Apply BZT. i.e Replace s by the ((z-1)/(z+1)). And find out the desired transfer function of digital function.

Example:

Q) Design first order high pass butterworth filter whose cutoff frequency is 1 kHz at sampling frequency of 10^4 sps. Use BZT Method

Step 1. To find out the cutoff frequency

$$\omega c = 2\Pi f$$
$$= 2000 \text{ rad/sec}$$

Step 2. To find the prewarp frequency

$$\omega c^* = \tan (\omega c\ Ts/2)$$
$$= \tan(\Pi/10)$$

Step 3. Scaling of the transfer function

For First order HPF transfer function $H(s) = s/(s+1)$

Scaled transfer function $H^*(s) = H(s)\ |_{s=s/\omega c^*}$

$H^{*}(s)= s/(s + 0.325)$

Step 4. Find out the digital filter transfer function. Replace s by (z-1)/(z+1)

$$H(z)= \frac{z-1}{1.325z -0.675}$$

Q) Design second order low pass butterworth filter whose cutoff frequency is 1 kHz at sampling frequency of 10^4 sps.

Q) First order low pass butterworth filter whose bandwidth is known to be 1 rad/sec . Use BZT method to design digital filter of 20 Hz bandwidth at sampling frequency 60 sps.

Q) Second order low pass butterworth filter whose bandwidth is known to be 1 rad/sec . Use BZT method to obtain transfer function H(z) of digital filter of 3 DB cutoff frequency of 150 Hz and sampling frequency 1.28 kHz.

Q) The transfer function is given as s^2+1 / s^2+s+1 The function is for Notch filter with frequency 1 rad/sec. Design digital Notch filter with the following specification

(1) Notch Frequency= 60 Hz
(2) Sampling frequency = 960 sps.

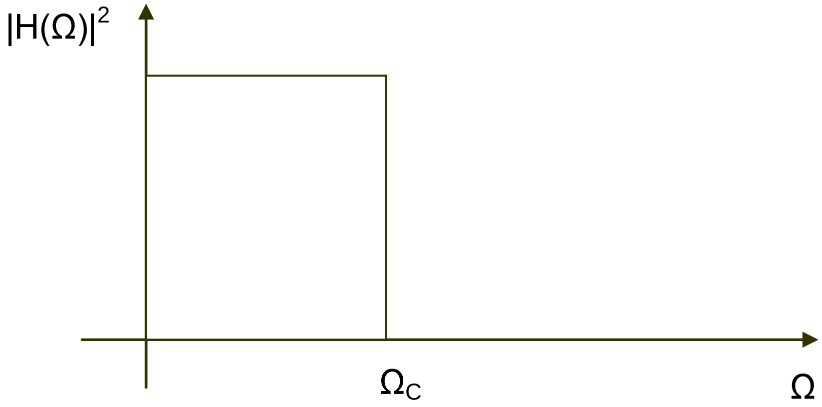
4.7 BUTTERWORTH FILTER APPROXIMATION

The filter passes all frequencies below Ω_c. This is called passband of the filter. Also the filter blocks all the frequencies above $\Omega_{c.}$ This is called stopband of the filter. Ω_c is called cutoff frequency or critical frequency.

No Practical filters can provide the ideal characteristic. Hence approximation of the ideal characteristic are used. Such approximations are standard and used for filter design. Such three approximations are regularly used.

a) Butterworth Filter Approximation
b) Chebyshev Filter Approximation
c) Elliptic Filter Approximation

Butterworth filters are defined by the property that the magnitude response is maximally flat in the passband.



$$|H_a(\Omega)|^2 = \frac{1}{1 + (\Omega/\Omega_c)^{2N}}$$

The squared magnitude function for an analog butterworth filter is of the form.

$$|H_a(\Omega)|^2 = \frac{1}{1 + (\Omega/\Omega_c)^{2N}}$$

N indicates order of the filter and Ω_c is the cutoff frequency (-3DB frequency).
At s = j Ω magnitude of H(s) and H(-s) is same hence

$$Ha(s)\ Ha(-s) = \frac{1}{1 + (-s^2/\Omega_c^2)^N}$$

To find poles of H(s). H(-s) , find the roots of denominator in above equation.

$$\frac{-s^2}{\Omega_c^2} = (-1)^{1/N}$$

As $e^{j(2k+1)\Pi} = -1$ where k = 0,1,2,…….. N-1.

$$\frac{-s^2}{\Omega_c^2} = (e^{j(2k+1)\Pi})^{1/N}$$

$$s^2 = (-1)\ \Omega_c^2\ e^{j(2k+1)\Pi / N}$$

Taking the square root we get poles of s.

$$p_k = \pm \sqrt{-1}\ \Omega_c\ [\ e^{j(2k+1)\Pi / N}\ ]^{1/2}$$

$$Pk = \pm j\ \Omega_c\ e^{j(2k+1)\Pi / 2N}$$

As $e^{j\Pi/2} = j$

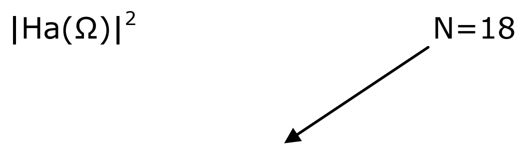
$$Pk = \pm \Omega_c\ e^{j\Pi/2}\ e^{j(2k+1)\Pi / 2N}$$

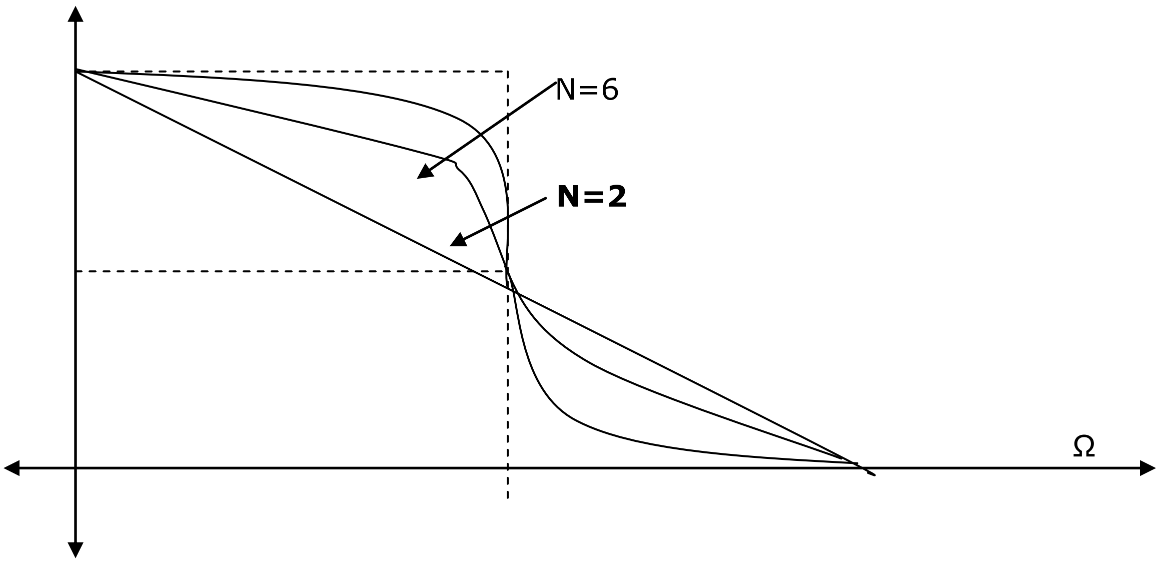
$$\mathbf{P_k = \pm \Omega_c\ e^{j(N+2k+1)\Pi / 2N}} \qquad \textbf{(1)}$$

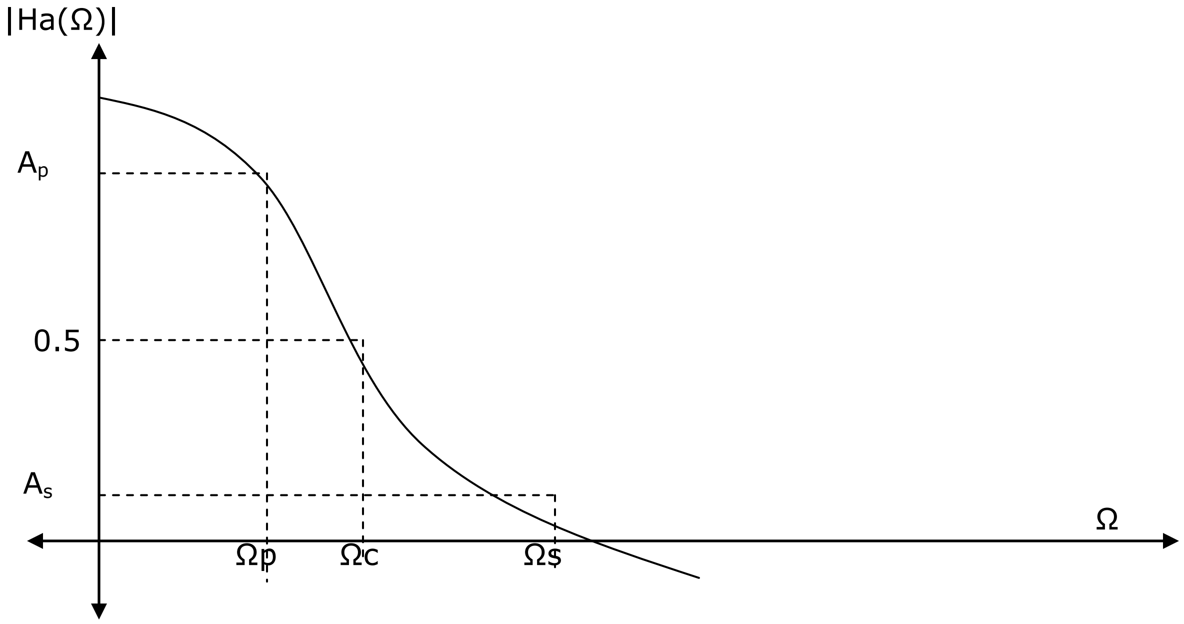
This equation gives the pole position of H(s) and H(-s).

FREQUENCY RESPONSE CHARACTERISTIC

The frequency response characteristic of $|Ha(\Omega)|^2$ is as shown. As the order of the filter N increases, the butterworth filter characteristic is more close to the ideal characteristic. Thus at higher orders like N=16 the butterworth filter characteristic closely approximate ideal filter characteristic. Thus an infinite order filter (N → ∞) is required to get ideal characteristic.







Ap= attenuation in passband.
As= attenuation in stopband.
Ω_p = passband edge frequency
Ω_s = stopband edge frequency

Specification for the filter is
|Ha(Ω)| ≥ Ap for Ω ≤ Ωp and |Ha(Ω)| ≤ As for Ω ≥ Ωs. Hence we have

$$\frac{1}{1 + (\Omega p/\Omega c)^{2N}} \geq Ap^2$$

$$\frac{1}{1 + (\Omega s/\Omega c)^{2N}} \leq As^2$$

To determine the poles and order of analog filter consider equalities.

$$(\Omega p/\Omega c)^{2N} = (1/Ap^2) - 1$$

$$(\Omega s/\Omega c)^{2N} = (1/As^2) - 1$$

$$\left[\frac{\Omega s}{\Omega p}\right]^{2N} = \frac{(1/As^2)-1}{(1/Ap^2)-1}$$

Hence order of the filter (N) is calculated as

$$\mathbf{N= 0.5}\ \frac{\log\left[\dfrac{(1/As^2)-1}{(1/Ap^2)-1}\right]}{\log\ (\Omega s/\ \Omega p)} \qquad \mathbf{(2)}$$

$$\mathbf{N= 0.5}\ \frac{\log((1/As2)\ -1)}{\log\ (\Omega s/\ \Omega c)} \qquad \mathbf{(2A)}$$

And cutoff frequency Ωc is calculated as

$$\Omega c = \frac{\Omega p}{[(1/Ap^2)\ -1]^{1/2N}} \qquad \mathbf{(3)}$$

If As and Ap values are given in DB then

$$As\ (DB) = -\ 20 \log As$$

$$\log As = -As\ /20$$

$$As = 10^{-As/20}$$

$$(As)^{-2} = 10^{As/10}$$

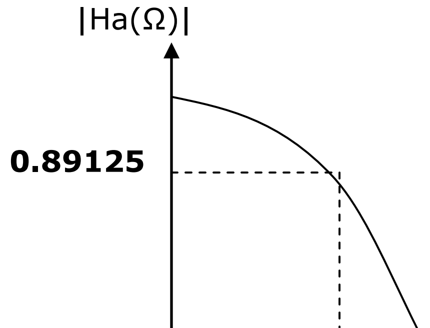
$$(As)^{-2} = 10^{0.1\ As}\ DB$$

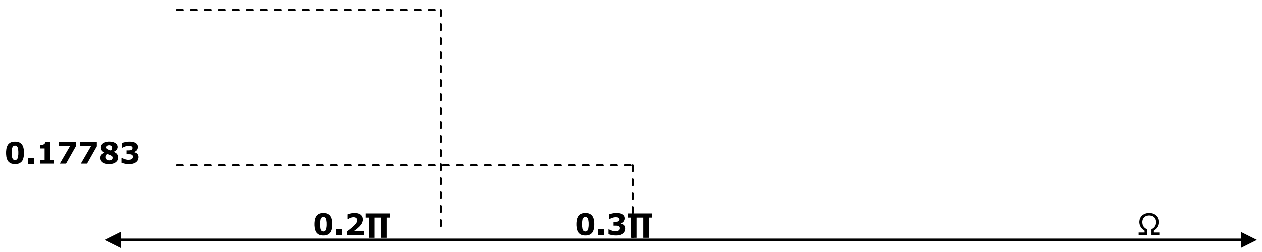
Hence equation (2) is modified as

$$\mathbf{N= 0.5}\ \frac{\log\left[\dfrac{10^{0.1\ As}-1}{10^{0.1\ Ap}-1}\right]}{\log\ (\Omega s/\ \Omega p)} \qquad \mathbf{(4)}$$

Q) Design a digital filter using a butterworth approximation by using impulse invariance.

Example





Filter Type - Low Pass Filter

Ap	- 0.89125
As	- 0.17783
Ωp	- 0.2∏
Ωs	- 0.3∏

Step 1) To convert specification to equivalent analog filter.
(In impulse invariance method frequency relationship is given as ω= Ω T while in Bilinear transformation method frequency relationship is given as Ω= (2/T) tan (ω/2) If Ts is not specified consider as 1)

$|Ha(\Omega)| \geq 0.89125$ for $\Omega \leq 0.2\Pi/T$ and $|Ha(\Omega)| \leq 0.17783$ for $\Omega \geq 0.3\Pi/T$.

Step 2) To determine the order of the filter.

$$N = 0.5 \; \frac{\log \left(\dfrac{(1/As^2)-1}{(1/Ap^2)-1} \right)}{\log\,(\Omega s/\ \Omega p)}$$

N= 5.88

A) Order of the filter should be integer.
B) Always go to nearest highest integer vale of N.

Hence N=6

Step 3) To find out the cutoff frequency (-3DB frequency)

$$\Omega c = \frac{\Omega p}{[(1/Ap^2)\ -1]^{1/2N}}$$

cutoff frequency Ωc = 0.7032

Step 4) To find out the poles of analog filter system function.

$$P_k = \pm\ \Omega_c \ \ e^{\,j(N+2k+1)\,\Pi\ /\ 2N}$$

As N=6 the value of k = 0,1,2,3,4,5.

K	Poles	
0	P0= $\pm$ 0.7032 e $^{j7\Pi/12}$	-0.182 + j 0.679 0.182 - j 0.679
1	P1= $\pm$ 0.7032 e $^{j9\Pi/12}$	-0.497 + j 0.497 0.497 - j 0.497
2	P2= $\pm$ 0.7032 e $^{j11\Pi/12}$	-0.679 + j 0.182 0.679 - j 0.182
3	P3= $\pm$ 0.7032 e $^{j13\Pi/12}$	-0.679 - j 0.182 0.679 + j 0.182
4	P4= $\pm$ 0.7032 e $^{j15\Pi/12}$	-0.497 - j 0.497 0.497 + j 0.497
5	P5= $\pm$ 0.7032 e $^{j17\Pi/12}$	-0.182 - j 0.679 0.182 + j 0.679

For stable filter all poles lying on the left side of s plane is selected. Hence

S1 = -0.182 + j 0.679	$S1^*$ = -0.182 - j 0.679
S2 = -0.497 + j 0.497	$S2^*$ = -0.497 - j 0.497
S3 = -0.679 + j 0.182	$S3^*$ = -0.679 - j 0.182

Step 5) To determine the system function (Analog Filter)

$$\mathbf{Ha(s) = \frac{\Omega c^6}{(s-s1)(s-s1^*)\ (s-s2)(s-s2^*)\ (s-s3)(s-s3^*)}}$$

Hence

$$\mathbf{Ha(s) =} \frac{(0.7032)^6}{(s+0.182-j0.679)(s+0.182+j0.679)\ (s+0.497-j0.497)\ (s+0.497+j0.497)\ (s+0.679-j0.182)(s+0.679-j0.182)}$$

$$\mathbf{Ha(s) =} \frac{0.1209}{[(s+0.182)^2 +(0.679)^2]\ [(s+0.497)^2+(0.497)^2]\ [(s+0.679)^2-(0.182)^2]}$$

$$\mathbf{Ha(s) =} \frac{1.97 \times\ 0.679 \times 0.497 \times\ 0.182}{[(s+0.182)^2 +(0.679)^2]\ [(s+0.497)^2+(0.497)^2]\ [(s+0.679)^2-(0.182)^2]}$$

Step 6) To determine the system function (Digital Filter)
(In Bilinear transformation replace s by the term ((z-1)/(z+1)) and find out the transfer function of digital function)

$$\frac{0.5235\ z^{-1}}{1-1.297z^{-1}+0.695z^{-2}} \qquad \frac{0.29\ z^{-1}}{1-1.07z^{-1}+0.37z^{-2}} \qquad \frac{0.09\ z^{-1}}{1-0.99z^{-1}+0.26z^{-2}}$$

H(z)=1.97 × ____________________×× ____________________× ____________________

Step 7) Represent system function in cascade form or parallel form if asked.

Q) Given for low pass butterworth filter
Ap= -1 db at 0.2∏
As= -15 db at 0.3∏

1) Calculate N and Pole location
2) Design digital filter using BZT method.

Q) Obtain transfer function of a lowpass digital filter meeting specifications
Cutoff 0-60Hz
Stopband > 85Hz
Stopband attenuation > 15 db
Sampling frequency= 256 Hz . use butterworth characteristic.

Q) Design second order low pass butterworth filter whose cutoff frequency is 1 kHz at sampling frequency of 10^4 sps. Use BZT and Butterworth approximation.

3.8 FREQUENCY TRANSFORMATION

When the cutoff frequency Ωc of the low pass filter is equal to 1 then it is called normalized filter. Frequency transformation techniques are used to generate High pass filter, Bandpass and bandstop filter from the lowpass filter system function.

FREQUENCY TRANSFORMATION (ANALOG FILTER)

Sr No	Type of transformation	Transformation (Replace s by)
1	Low Pass	$\frac{s}{\omega_{lp}}$ ω_{lp} - Password edge frequency of another LPF
2	High Pass	$\frac{\omega_{hp}}{s}$ ω_{hp} = Password edge frequency of HPF
3	Band Pass	$\frac{(s^2 + \omega_l \omega_h)}{s(\omega_h - \omega_l)}$ ω_h - higher band edge frequency ω_l - Lower band edge frequency
4	Band Stop	$\frac{s(\omega_h - \omega_l)}{s^2 + \omega_h \omega_l}$ ω_h - higher band edge frequency ω_l - Lower band edge frequency

FREQUENCY TRANSFORMATION (DIGITAL FILTER)

Sr No	Type of transformation	Transformation (Replace z^{-1} by)
1	Low Pass	$\frac{z^{-1} - a}{1 - az^{-1}}$
2	High Pass	$\frac{-(z^{-1} + a)}{1 + az^{-1}}$
3	Band Pass	$\frac{-(z^{-2} - a_1z^{-1} + a_2)}{a_2z^{-2} - a_1z^{-1} + 1}$
4	Band Stop	$\frac{z^{-2} - a_1z^{-1} + a_2}{a_2z^{-2} - a_1z^{-1} + 1}$

Example:

Q) Design high pass butterworth filter whose cutoff frequency is 30 Hz at sampling frequency of 150 Hz. Use BZT and Frequency transformation.

Step 1. To find the prewarp cutoff frequency

$$\omega c^* = \tan(\omega c Ts/2)$$
$$= 0.7265$$

Step 2. LPF to HPF transformation

For First order LPF transfer function H(s) = 1/(s+1)

Scaled transfer function $H^*(s) = H(s)\,|_{s=\omega c^*/s}$

$H^*(s) = s/(s + 0.7265)$

Step 4. Find out the digital filter transfer function. Replace s by (z-1)/(z+1)

$$H(z) = \frac{z-1}{1.7265z - 0.2735}$$

Q) Design second order band pass butterworth filter whose passband of 200 Hz and 300 Hz and sampling frequency is 2000 Hz. Use BZT and Frequency transformation.

Q) Design second order band pass butterworth filter which meet following specification

Lower cutoff frequency = 210 Hz
Upper cutoff frequency = 330 Hz
Sampling Frequency = 960 sps

Use BZT and Frequency transformation.

UNIT 4
FIR FILTER DESIGN

Features of FIR Filter

1. FIR filter always provides linear phase response. This specifies that the signals in the pass band will suffer no dispersion Hence when the user wants no phase distortion, then FIR filters are preferable over IIR. Phase distortion always degrade the system performance. In various applications like speech processing, data transmission over long distance FIR filters are more preferable due to this characteristic.

2. FIR filters are most stable as compared with IIR filters due to its non feedback nature.

3. Quantization Noise can be made negligible in FIR filters. Due to this sharp cutoff FIR filters can be easily designed.

4. Disadvantage of FIR filters is that they need higher ordered for similar magnitude response of IIR filters.

FIR SYSTEM ARE ALWAYS STABLE. Why?

Proof:

Difference equation of FIR filter of length M is given as

$$y(n)=\sum_{k=0}^{M-1} b_k\, x(n-k) \qquad (1)$$

And the coefficient b_k are related to unit sample response as

$H(n) = b_n$ for $0 \leq n \leq M-1$

$= 0$ otherwise.

We can expand this equation as

$$Y(n)= b_0\, x(n) + b_1\, x(n-1) + \ldots\ldots + b_{M-1}\, x(n-M+1) \qquad (2)$$

System is stable only if system produces bounded output for every bounded input. This is stability definition for any system.

Here h(n)={b0, b1, b2, } of the FIR filter are stable. Thus y(n) is bounded if input x(n) is bounded. This means FIR system produces bounded output for every bounded input. Hence FIR systems are always stable.

Symmetric and Anti-symmetric FIR filters

1. Unit sample response of FIR filters is symmetric if it satisfies following condition

$h(n)= h(M-1-n) \qquad n=0,1,2……………M-1$

2. Unit sample response of FIR filters is Anti-symmetric if it satisfies following condition

$h(n)= -h(M-1-n) \qquad n=0,1,2……………M-1$

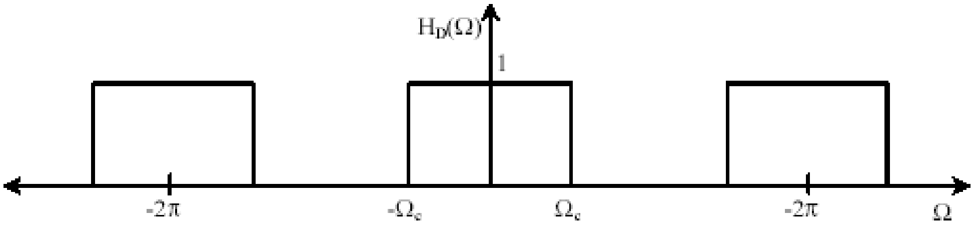
FIR Filter Design Methods

The various method used for FIR Filer design are as follows

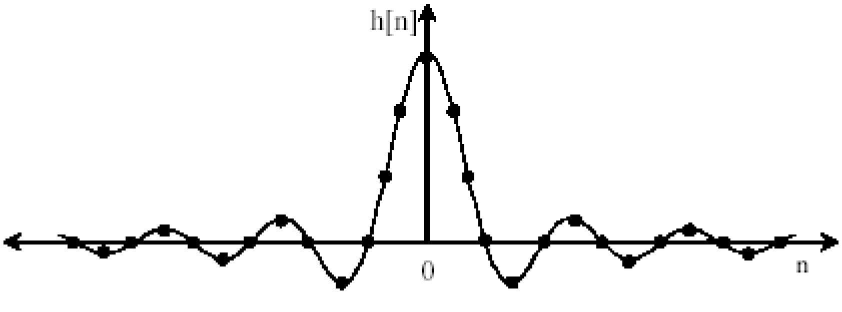
1. Fourier Series method
2. Windowing Method
3. DFT method
4. Frequency sampling Method. (IFT Method)

GIBBS PHENOMENON

Consider the ideal LPF frequency response as shown in Fig 1 with a normalizing angular cut off frequency Ω_c.



Impulse response of an ideal LPF is as shown in Fig 2.

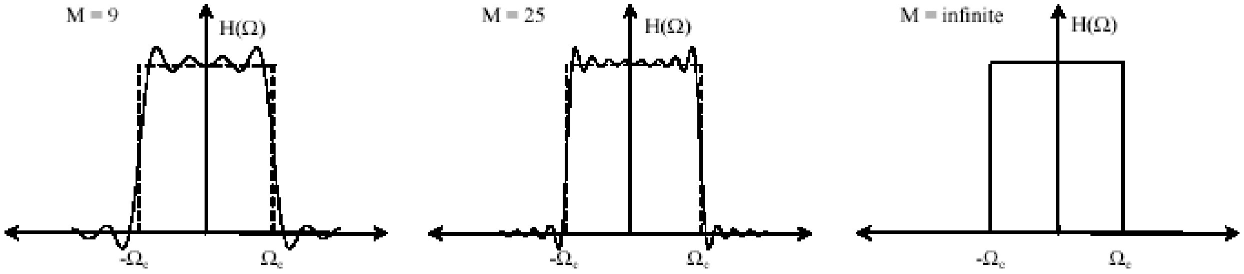


1. In Fourier series method, limits of summation index is -∞ to ∞. But filter must have finite terms. Hence limit of summation index change to -Q to Q where Q is some finite integer. But this type of truncation may result in poor convergence of the series. Abrupt truncation of infinite series is equivalent to multiplying infinite series with rectangular sequence. i.e at the point of discontinuity some oscillation may be observed in resultant series.

2. Consider the example of LPF having desired frequency response $H_d(\omega)$ as shown in figure. The oscillations or ringing takes place near band-edge of the filter.

3. This oscillation or ringing is generated because of side lobes in the frequency response $W(\omega)$ of the window function. This oscillatory behavior is called "Gibbs Phenomenon".

Truncated response and ringing effect is as shown in fig 3.

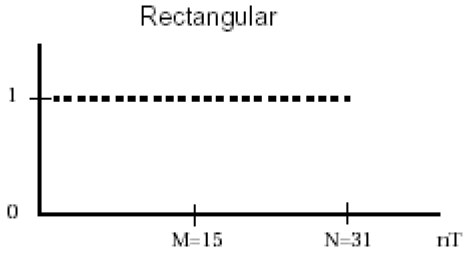


WINDOWING TECHNIQUE W[n]

Windowing is the quickest method for designing an FIR filter. A windowing function simply truncates the ideal impulse response to obtain a causal FIR approximation that is non causal and infinitely long. Smoother window functions provide higher out-of band rejection in the filter response. However this smoothness comes at the cost of wider stopband transitions.

Various windowing method attempts to minimize the width of the main lobe (peak) of the frequency response. In addition, it attempts to minimize the side lobes (ripple) of the frequency response.

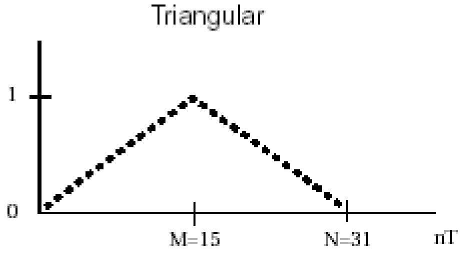
Rectangular Window: Rectangular This is the most basic of windowing methods. It does not require any operations because its values are either 1 or 0. It creates an abrupt discontinuity that results in sharp roll-offs but large ripples.



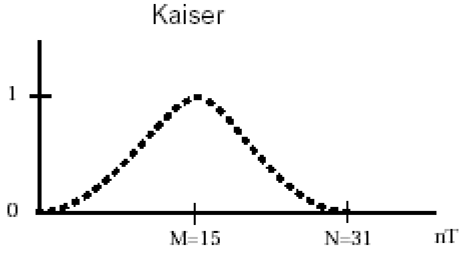
Rectangular window is defined by the following equation.

$$\begin{aligned} &= 1 \qquad \text{for } 0 \le n \le N \\ &= 0 \qquad \text{otherwise} \end{aligned}$$

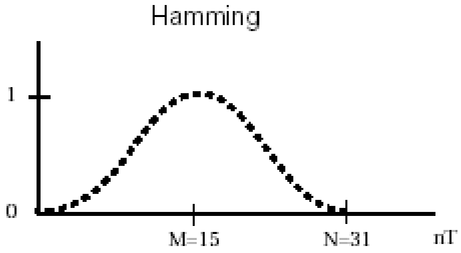
Triangular Window: The computational simplicity of this window, a simple convolution of two rectangle windows, and the lower sidelobes make it a viable alternative to the rectangular window.



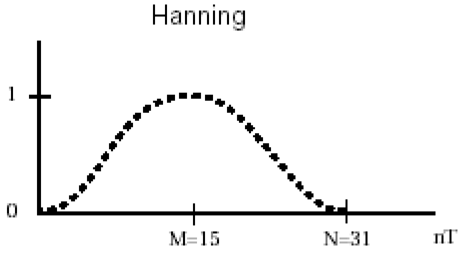
Kaiser Window: This windowing method is designed to generate a sharp central peak. It has reduced side lobes and transition band is also narrow. Thus commonly used in FIR filter design.

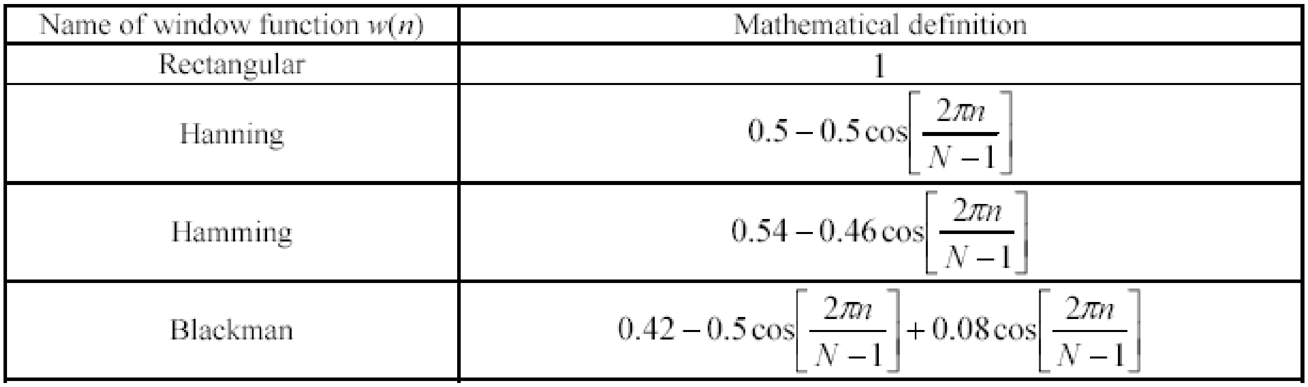


Hamming Window: This windowing method generates a moderately sharp central peak. Its ability to generate a maximally flat response makes it convenient for speech processing filtering.



Hanning Window: This windowing method generates a maximum flat filter design.





Name of window function $w(n)$	Mathematical definition
Rectangular	1
Hanning	$0.5-0.5\cos\left[\frac{2\pi n}{N-1}\right]$
Hamming	$0.54-0.46\cos\left[\frac{2\pi n}{N-1}\right]$
Blackman	$0.42-0.5\cos\left[\frac{2\pi n}{N-1}\right]+0.08\cos\left[\frac{2\pi n}{N-1}\right]$

4.10 DESIGNING FILTER DESIGN FROM POLE ZERO PLACEMENT

Filters can be designed from its pole zero plot. Following two constraints should be imposed while designing the filters.

1. All poles should be placed inside the unit circle on order for the filter to be stable. However zeros can be placed anywhere in the z plane. FIR filters are all zero filters hence they are always stable. IIR filters are stable only when all poles of the filter are inside unit circle.

2. All complex poles and zeros occur in complex conjugate pairs in order for the filter coefficients to be real.

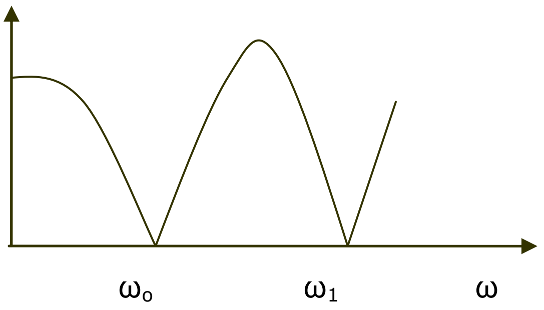
In the design of low pass filters, the poles should be placed near the unit circle at points corresponding to low frequencies (near ω=0)and zeros should be placed near or on unit circle at points corresponding to high frequencies (near ω=∏). The opposite is true for high pass filters.

NOTCH AND COMB FILTERS

A notch filter is a filter that contains one or more deep notches or ideally perfect nulls in its frequency response characteristic. Notch filters are useful in many applications where specific frequency components must be eliminated. Example Instrumentation and recording systems required that the power-line frequency 60Hz and its harmonics be eliminated.

To create nulls in the frequency response of a filter at a frequency ω_0, simply introduce a pair of complex-conjugate zeros on the unit circle at an angle ω_0.

comb filters are similar to notch filters in which the nulls occur periodically across the frequency band similar with periodically spaced teeth. Frequency response characteristic of notch filter |H(ω)| is as shown



DIGITAL RESONATOR

A digital resonator is a special two pole bandpass filter with a pair of complex conjugate poles located near the unit circle. The name resonator refers to the fact that the filter has a larger magnitude response in the vicinity of the pole locations. Digital resonators are useful in many applications, including simple bandpass filtering and speech generations.

IDEAL FILTERS ARE NOT PHYSICALLY REALIZABLE. Why?

Ideal filters are not physically realizable because Ideal filters are anti-causal and as only causal systems are physically realizable.

Proof:

Let take example of ideal lowpass filter.

$$H(\omega) = 1 \text{ for } -\omega_c \le \omega \le \omega_c$$
$$= 0 \text{ elsewhere}$$

The unit sample response of this ideal LPF can be obtained by taking IFT of H(ω).

$$h(n) = \frac{1}{2\Pi} \int_{-\infty}^{\infty} H(\omega)\, e^{j\omega n}\, d\omega \qquad (1)$$

$$h(n) = \frac{1}{2\Pi} \int_{-\omega c}^{\omega c} 1\, e^{j\omega n}\, d\omega \qquad (2)$$

$$h(n) = \frac{1}{2\Pi} \left[\frac{e^{j\omega n}}{jn} \right]_{-\omega c}^{\omega c}$$

$$\frac{1}{2\Pi jn} \left[e^{j\omega cn} - e^{-j\omega cn}\right]$$

Thus $h(n) = \sin \omega_c n / \Pi n$ for $n \neq 0$

Putting n=0 in equation (2) we have

$$h(n) = \frac{1}{2\Pi} \int_{-\omega c}^{\omega c} 1\, d\omega \qquad (3)$$

$$\frac{1}{2\Pi}\quad[\omega]_{-\omega c}^{\omega c}$$

and $h(n) = \omega_c / \Pi$ for n=0

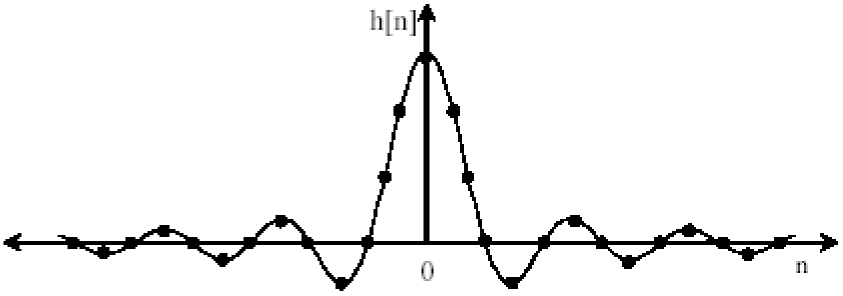
i.e

$$\frac{\sin(\omega cn)}{\Pi n}$$ for n≠0

h(n) =

$$\frac{\omega c}{n}$$ for n=0

Hence impulse response of an ideal LPF is as shown in Fig



LSI system is causal if its unit sample response satisfies following condition.

h(n) = 0 for n<0

In above figure h(n) extends -∞ to ∞. Hence h(n) ≠0 for n<0. This means causality condition is not satisfied by the ideal low pass filter. Hence ideal low pass filter is non causal and it is not physically realizable.

EXAMPLES OF SIMPLE DIGITAL FILTERS:

The following examples illustrate the essential features of digital filters.

1. **UNITY GAIN FILTER:** $y_n = x_n$
   Each output value y_n is exactly the same as the corresponding input value x_n:

2. **SIMPLE GAIN FILTER:** $y_n = Kx_n$ (K = constant) Amplifier or attenuator)
   This simply applies a gain factor *K* to each input value:

3. **PURE DELAY FILTER:** $y_n = x_{n-1}$
   The output value at time $t = nh$ is simply the input at time $t = (n-1)h$, i.e. the signal is delayed by time *h*:

4. **TWO-TERM DIFFERENCE FILTER:** $y_n = x_n - x_{n-1}$
   The output value at $t = nh$ is equal to the difference between the current input x_n and the previous input x_{n-1}:

5. **TWO-TERM AVERAGE FILTER:** $y_n = (x_n + x_{n-1}) / 2$
   The output is the average (arithmetic mean) of the current and previous input:

6. **THREE-TERM AVERAGE FILTER:** $y_n = (x_n + x_{n-1} + x_{n-2}) / 3$
This is similar to the previous example, with the average being taken of the current and two previous inputs.

7. **CENTRAL DIFFERENCE FILTER:** $y_n = (x_n - x_{n-2}) / 2$
This is similar in its effect to example (4). The output is equal to half the change in the input signal over the previous two sampling intervals:

ORDER OF A DIGITAL FILTER

The order of a digital filter can be defined as the *number of previous inputs* (stored in the processor's memory) used to calculate the current output.
This is illustrated by the filters given as examples in the previous section.

Example (1): $y_n = x_n$
This is a *zero order* filter, since the current output y_n depends only on the current input x_n and not on any previous inputs.

Example (2): $y_n = Kx_n$
The order of this filter is again *zero*, since no previous outputs are required to give the current output value.

Example (3): $y_n = x_{n-1}$
This is a *first order* filter, as one previous input (x_{n-1}) is required to calculate y_n. (Note that this filter is classed as first-order because it uses one *previous* input, even though the current input is not used).

Example (4): $y_n = x_n - x_{n-1}$
This is again a *first order* filter, since one previous input value is required to give the current output.

Example (5): $y_n = (x_n + x_{n-1}) / 2$
The order of this filter is again equal to 1 since it uses just one previous input value.

Example (6): $y_n = (x_n + x_{n-1} + x_{n-2}) / 3$
To compute the current output y_n, two previous inputs (x_{n-1} and x_{n-2}) are needed; this is therefore a *second-order* filter.

Example (7): $y_n = (x_n - x_{n-2}) / 2$
The filter order is again 2, since the processor must store two previous inputs in order to compute the current output. This is unaffected by the absence of an explicit x_{n-1} term in the filter expression.

Q) For each of the following filters, state the order of the filter and identify the values of its coefficients:

(a) $yn = 2x_n - x_{n-1}$ A) Order = 1: $a_0 = 2$, $a_1 = -1$
(b) $yn = x_n$-2 B) Order = 2: $a_0 = 0$, $a_1 = 0$, $a_2 = 1$
(c) $yn = x_n - 2x_{n-1} + 2x_{n-2} + x_{n-3}$ C) Order = 3: $a_0 = 1$, $a_1 = -2$, $a_2 = 2$, $a_3 = 1$

Of these, the linear phase property is probably the most important. A filter is said to have a generalised linear phase response if its frequency response can be expressed in the form

$$H(e^{j\omega}) = A(e^{j\omega})e^{-j\alpha\omega+j\beta}$$

where α and β are constants, and $A(e^{j\omega})$ is a real function of ω. If this is the case, then

- If A is positive, then the phase is

  $$\sphericalangle H(e^{j\omega}) = \beta - \alpha\omega.$$

  If A is negative, then

  $$\sphericalangle H(e^{j\omega}) = \pi + \beta - \alpha\omega.$$

In either case, the phase is a linear function of ω.

It is common to restrict the filter to having a real-valued impulse response $h[n]$, since this greatly simplifies the computational complexity in the implementation of the filter.

A FIR system has linear phase if the impulse response satisfies either the even symmetric condition

$$h[n] = h[N-1-n],$$

or the odd symmetric condition

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

The system has different characteristics depending on whether N is even or odd. Furthermore, it can be shown that all linear phase filters must satisfy one of these conditions. Thus there are exactly four types of linear phase filters.

Consider for example the case of an odd number of samples in $h[n]$, and even symmetry. The frequency response for $N = 7$ is

$$\begin{aligned}
H(e^{j\omega}) &= \sum_{n=0}^{6} h[n]e^{-j\omega n} \\
&= h[0] + h[1]e^{-j\omega} + h[2]e^{-j2\omega} + h[3]e^{-j3\omega} + h[4]e^{-j4\omega} \\
&\qquad + h[5]e^{-j5\omega} + h[6]e^{-j6\omega} \\
&= e^{-j3\omega}(h[0]e^{j3\omega} + h[1]e^{j2\omega} + h[2]e^{j\omega} + h[3] + h[4]e^{-j\omega} \\
&\qquad + h[5]e^{-j2\omega} + h[6]e^{-j3\omega}).
\end{aligned}$$

The specified symmetry property means that $h[0] = h[6]$, $h[1] = h[5]$, and $h[2] = h[4]$, so

$$\begin{aligned}
H(e^{j\omega}) &= e^{-j3\omega}(h[0](e^{j3\omega} + e^{-j3\omega}) + h[1](e^{j2\omega} + e^{-j2\omega}) \\
&\qquad + h[2](e^{j\omega} + e^{-j\omega}) + h[3]) \\
&= e^{-j3\omega}(2h[0]\cos(3\omega) + 2h[1]\cos(2\omega) + 2h[2]\cos(\omega)) \\
&= e^{-j3\omega}\sum_{n=0}^{3} a[n]\cos(\omega n),
\end{aligned}$$

where $a[0] = h[3]$, and $a[n] = 2h[3-n]$ for $n = 1, 2, 3$. The resulting filter clearly has a linear phase response for real $h[n]$. It is quite simple to show that in general for odd values of N the frequency response is

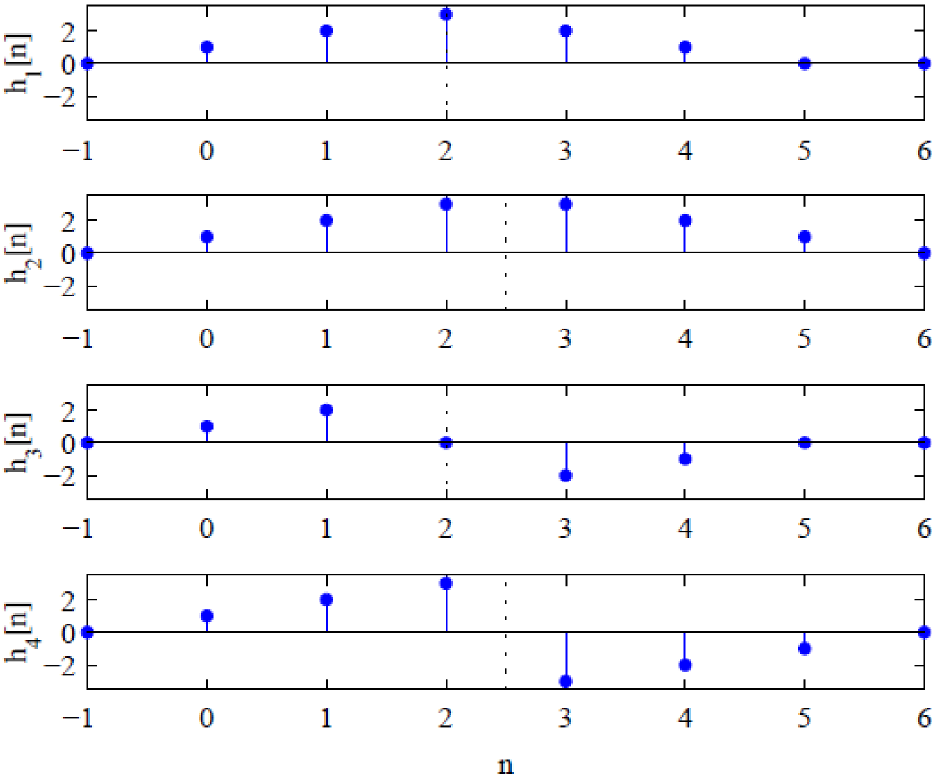
$$H(e^{j\omega}) = e^{-j\omega(N-1)/2} \sum_{n=0}^{(N-1)/2} a[n]\cos(\omega n),$$

for a set of real-valued coefficients $a[0], \ldots, a[(N-1)/2]$. As different values for $a[n]$ are selected, different linear-phase filters are obtained.

The cases of N odd and $h[n]$ antisymmetric are similar to that presented, and the frequency responses are summarised in the following table:

Symmetry	**N**	$H(e^{j\omega})$	**Type**
Even	Odd	$e^{-j\omega(N-1)/2} \sum_{n=0}^{(N-1)/2} a[n] \cos(\omega n)$	1
Even	Even	$e^{-j\omega(N-1)/2} \sum_{n=1}^{N/2} b[n] \cos(\omega(n-1/2))$	2
Odd	Odd	$e^{-j[\omega(N-1)/2-\pi/2]} \sum_{n=0}^{(N-1)/2} a[n] \sin(\omega n)$	3
Odd	Even	$e^{-j[\omega(N-1)/2-\pi/2]} \sum_{n=1}^{N/2} b[n] \sin(\omega(n-1/2))$	4

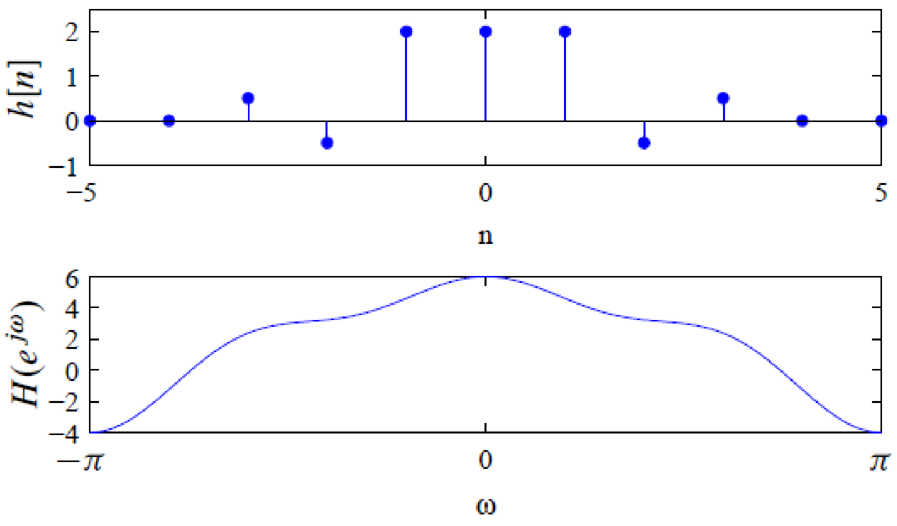
Recall that even symmetry implies $h[n] = h[N-1-n]$ and odd symmetry $h[n] = -h[N-1-n]$. Examples of filters satisfying each of these symmetry conditions are:



The center of symmetry is indicated by the dotted line.

The process of linear-phase filter design involves choosing the $a[n]$ values to obtain a filter with a desired frequency response. This is not always possible, however — the frequency response for a type II filter, for example, has the property that it is *always* zero for $\omega = \pi$, and is therefore not appropriate for a highpass filter. Similarly, filters of type 3 and 4 introduce a 90° phase shift, and have a frequency response that is always zero at $\omega = 0$ which makes them unsuitable for as lowpass filters. Additionally, the type 3 response is always zero at $\omega = \pi$, making it unsuitable as a highpass filter. The type I filter is the most versatile of the four.

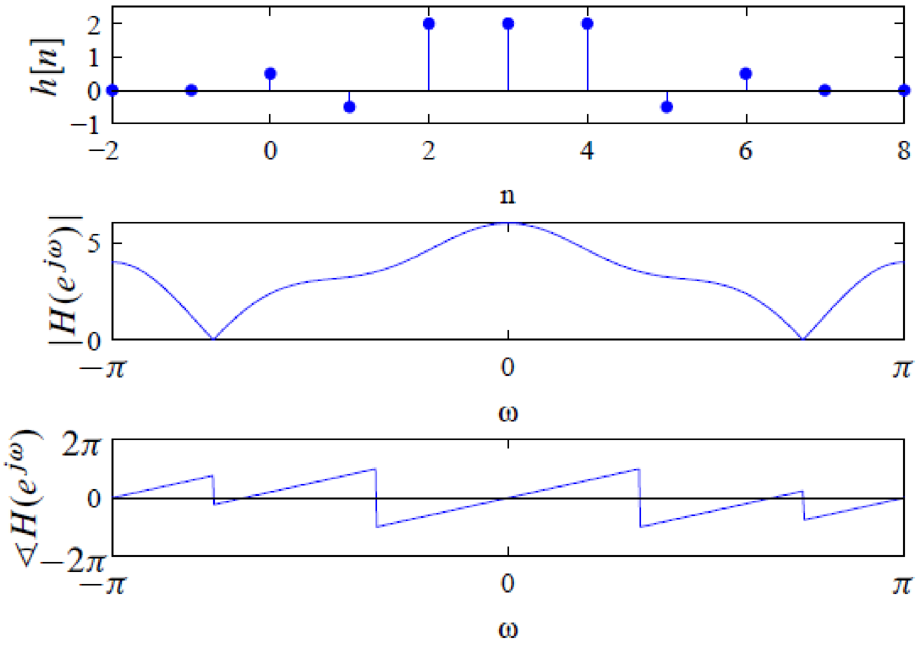
Linear phase filters can be thought of in a different way. Recall that a linear phase characteristic simply corresponds to a time shift or delay. Consider now a real FIR filter with an impulse response that satisfies the even symmetry condition $h[n] = h[-n]$:



Recall from the properties of the Fourier transform this filter has a real-valued frequency response $A(e^{j\omega})$. Delaying this impulse response by $(N-1)/2$ results in a causal filter with frequency response

$$H(e^{j\omega}) = A(e^{j\omega})e^{-j\omega(N-1)/2}.$$

This filter therefore has linear phase.



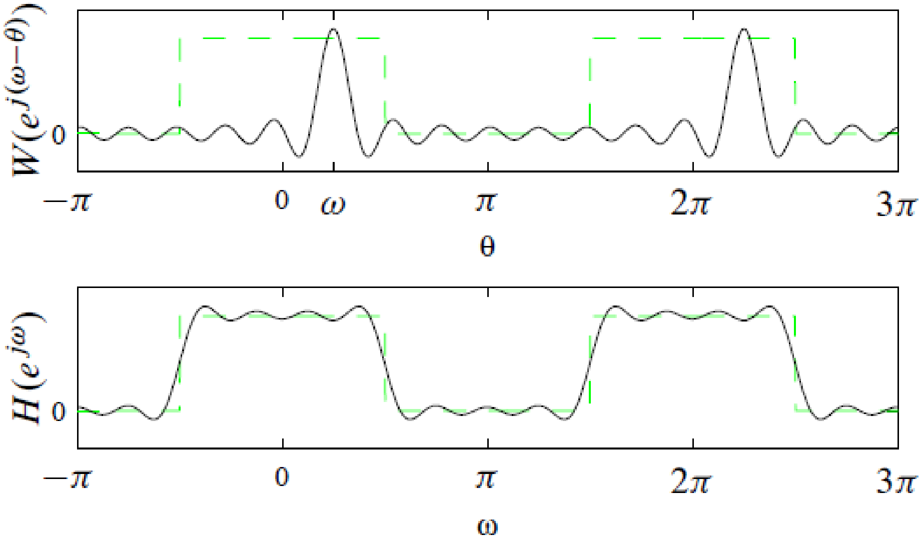
Window method for FIR filter design

Assume that the desired filter response $H_d(e^{j\omega})$ is known. Using the inverse Fourier transform we can determine $h_d[n]$, the desired unit sample response. In the window method, a FIR filter is obtained by multiplying a window $w[n]$ with $h_d[n]$ to obtain a finite duration $h[n]$ of length N. This is required since $h_d[n]$ will in general be an infinite duration sequence, and the corresponding filter will therefore not be realisable. If $h_d[n]$ is even or odd symmetric and $w[n]$ is even symmetric, then $h_d[n]w[n]$ is a linear phase filter.

Two important design criteria are the *length* and *shape* of the window $w[n]$. To see how these factors influence the design, consider the multiplication operation in the frequency domain: since $h[n] = h_d[n]w[n]$,

$$H(e^{j\omega}) = H_d(e^{j\omega}) * W(e^{j\omega}).$$

The following plot demonstrates the convolution operation. In each case the dotted line indicates the desired response $H_d(e^{j\omega})$.



From this, note that

- The *mainlobe* width of $W(e^{j\omega})$ affects the *transition* width of $H(e^{j\omega})$. Increasing the length N of $h[n]$ reduces the mainlobe width and hence the

transition width of the overall response.

- The *sidelobes* of $W(e^{j\omega})$ affect the passband and stopband tolerance of $H(e^{j\omega})$. This can be controlled by changing the shape of the window. Changing N does not affect the sidelobe behaviour.

Some commonly used windows for filter design are

- **Rectangular:**

$$w[n] = \begin{cases} 1 & 0 \leq n \leq N \\ 0 & \text{otherwise} \end{cases}$$

- **Bartlett (triangular):**

$$w[n] = \begin{cases} 2n/N & 0 \leq n \leq N/2 \\ 2 - 2n/N & N/2 < n \leq N \\ 0 & \text{otherwise} \end{cases}$$

- **Hanning:**

$$w[n] = \begin{cases} 0.5 - 0.5\cos(2\pi n/N) & 0 \leq n \leq N \\ 0 & \text{otherwise} \end{cases}$$

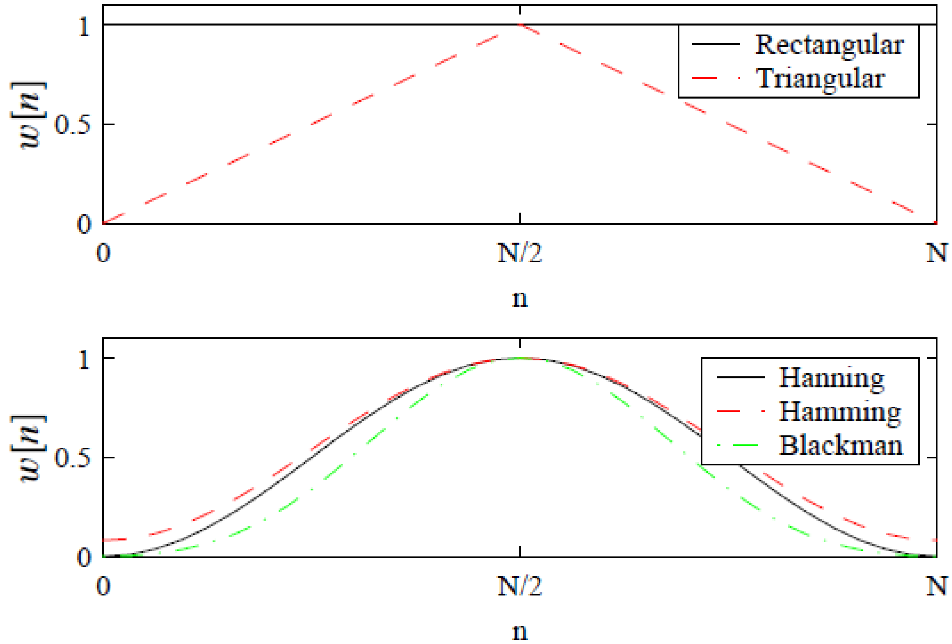
- **Hamming:**

$$w[n] = \begin{cases} 0.54 - 0.46\cos(2\pi n/N) & 0 \leq n \leq N \\ 0 & \text{otherwise} \end{cases}$$

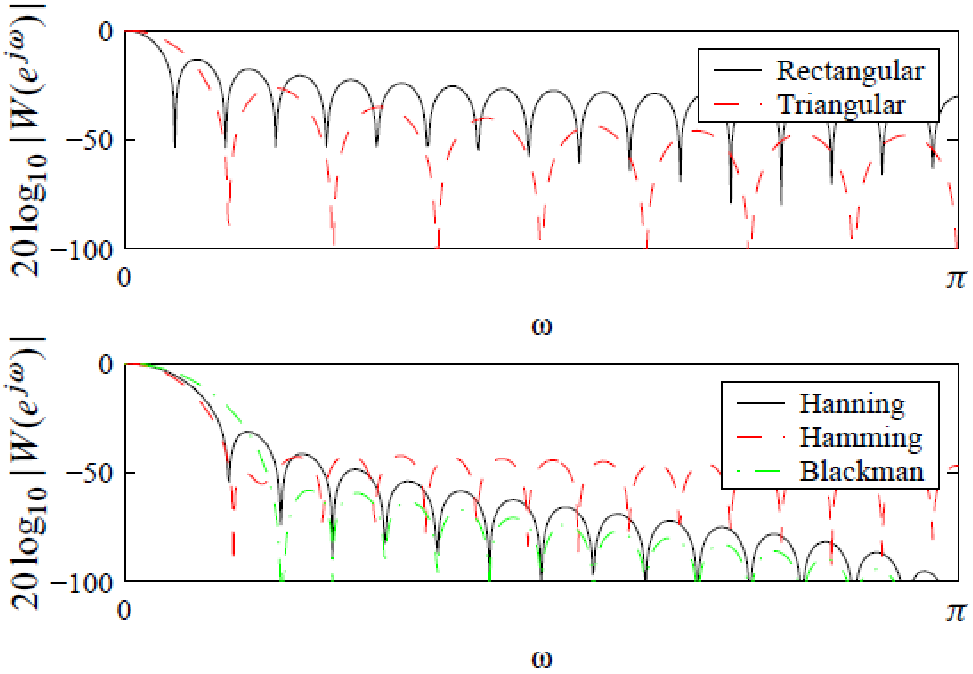
- **Kaiser:**

$$w[n] = \begin{cases} I_0[\beta(1 - [(n-\alpha)/\alpha]^2)^{1/2}] & 0 \leq n \leq N \\ 0 & \text{otherwise} \end{cases}$$

Examples of five of these windows are shown below:



All windows trade off a reduction in sidelobe level against an increase in mainlobe width. This is demonstrated below in a plot of the frequency response of each of the windows:



Some important window characteristics are compared in the following table:

Window	Peak sidelobe amplitude (dB)	Mainlobe transition width	Peak approximation error (dB)
Rectangular	-13	$4\pi/(N+1)$	-21
Bartlett	-25	$8\pi/N$	-25
Hanning	-31	$8\pi/N$	-44
Hamming	-41	$8\pi/N$	-53

The Kaiser window has a number of parameters that can be used to explicitly tune the characteristics.

In practice, the window shape is chosen first based on passband and stopband tolerance requirements. The window size is then determined based on transition width requirements. To determine $h_d[n]$ from $H_d(e^{j\omega})$ one can sample $H_d(e^{j\omega})$ closely and use a large inverse DFT.

2.2 Frequency sampling method for FIR filter design

In this design method, the desired frequency response $H_d(e^{j\omega})$ is sampled at equally-spaced points, and the result is inverse discrete Fourier transformed.

Specifically, letting

$$H[k] = H_d(e^{j\omega})\big|_{\omega=\frac{2\pi k}{N}}, \qquad k = 0, \ldots, N-1,$$

the unit sample response of the filter is $h[n] = \text{IDFT}(H[k])$, so

$$h[n] = \frac{1}{N}\sum_{k=0}^{N-1} H[k]e^{j2\pi nk/N}.$$

The resulting filter will have a frequency response that is exactly the same as the original response at the sampling instants. Note that it is also necessary to specify the *phase* of the desired response $H_d(e^{j\omega})$, and it is usually chosen to be a linear function of frequency to ensure a linear phase filter. Additionally, if a filter with real-valued coefficients is required, then additional constraints have to be enforced.

The *actual* frequency response $H(e^{j\omega})$ of the filter $h[n]$ still has to be determined. The z-transform of the impulse response is

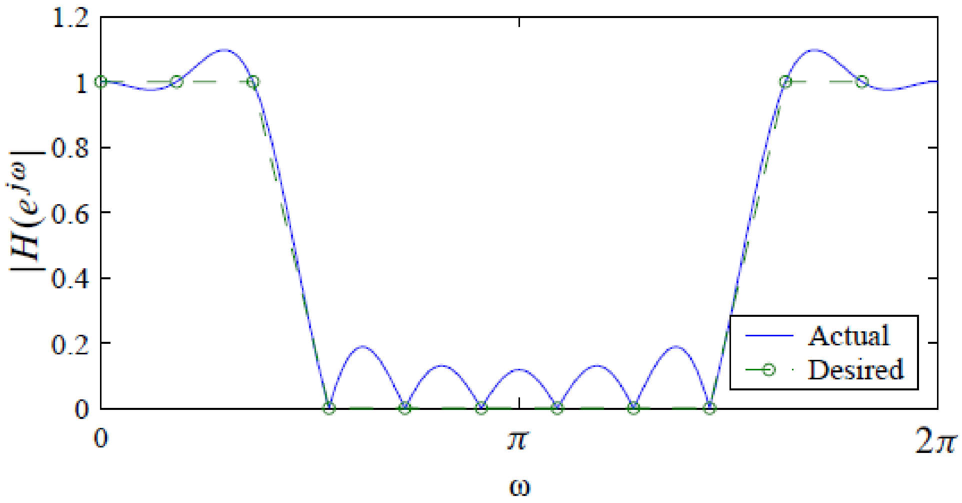
$$\begin{aligned} H(z) &= \sum_{n=0}^{N-1} h[n] z^{-n} = \sum_{n=0}^{N-1} \left[\frac{1}{N} \sum_{k=0}^{N-1} H[k] e^{j2\pi nk/N} \right] z^{-n} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} H[k] \sum_{n=0}^{N-1} e^{j2\pi nk/N} z^{-n} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} H[k] \left[\frac{1 - z^{-N}}{1 - e^{j2\pi k/N} z^{-1}} \right]. \end{aligned}$$

Evaluating on the unit circle $z = e^{j\omega}$ gives the frequency response

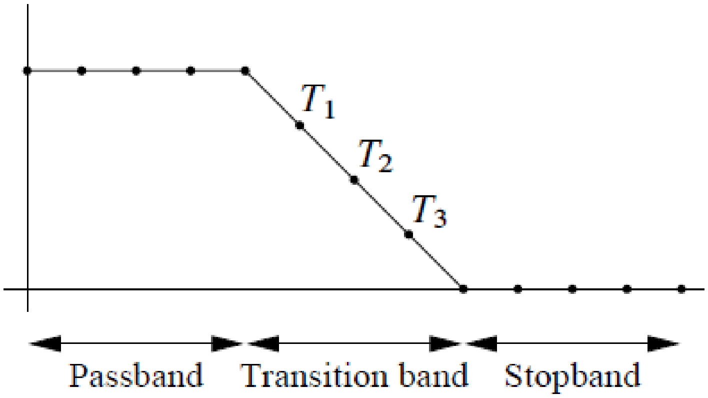
$$H(e^{j\omega}) = \frac{1 - e^{-j\omega N}}{N} \sum_{k=0}^{N-1} \frac{H[k]}{1 - e^{j2\pi k/N} e^{-j\omega}}.$$

This expression can be used to find the actual frequency response of the filter obtained, which can be compared with the desired response.

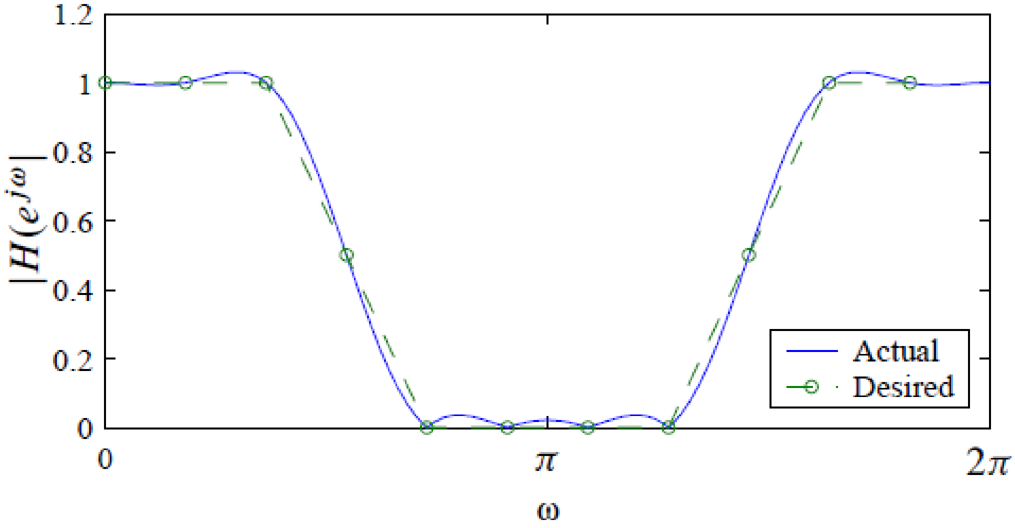
The method described only guarantees correct frequency response values at th points that were sampled. This sometimes leads to excessive ripple at intermediate points:



One way of addressing this problem is to allow **transition samples** in the region where discontinuities in $H_d(e^{j\omega})$ occur:



This effectively increases the transition width and can decrease the ripple, as observed below:



By leaving the value of the transition sample unconstrained, one can to some extent optimise the filter to minimise the ripple. Empirically, with three transition samples a stopband attenuation of 100dB is achievable. Recall however that for $h[n]$ real we require even or odd symmetry in the impulse response, so the values are not entirely unconstrained.

UNIT 5
APPLICATIONS OF DSP

5.5 APPLICATIONS OF DSP

1. SPEECH RECOGNITION

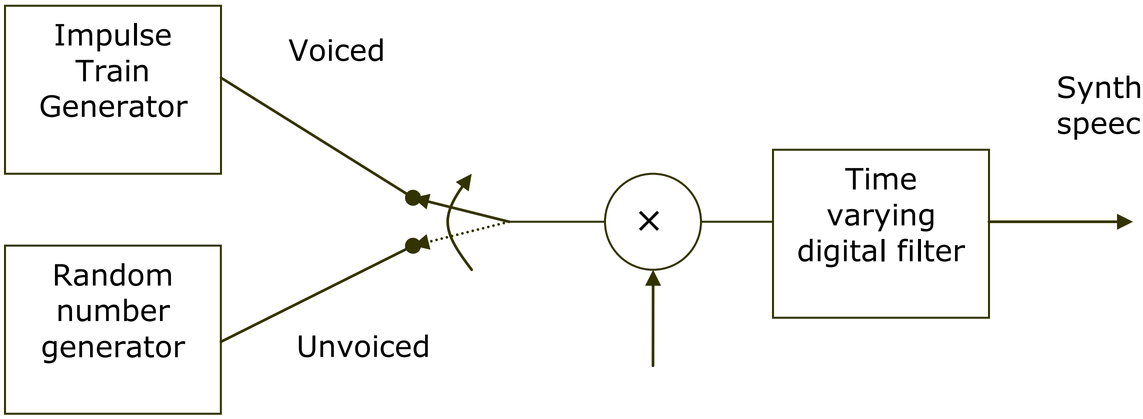
Basic block diagram of a speech recognition system is shown in Fig 1

1. In speech recognition system using microphone one can input speech or voice. The analog speech signal is converted to digital speech signal by speech digitizer. Such digital signal is called digitized speech.

2. The digitized speech is processed by DSP system. The significant features of speech such as its formats, energy, linear prediction coefficients are extracted. The template of this extracted features are compared with the standard

reference templates. The closed matched template is considered as the recognized word.

3. Voice operated consumer products like TV, VCR, Radio, lights, fans and voice operated telephone dialing are examples of DSP based speech recognized devices.



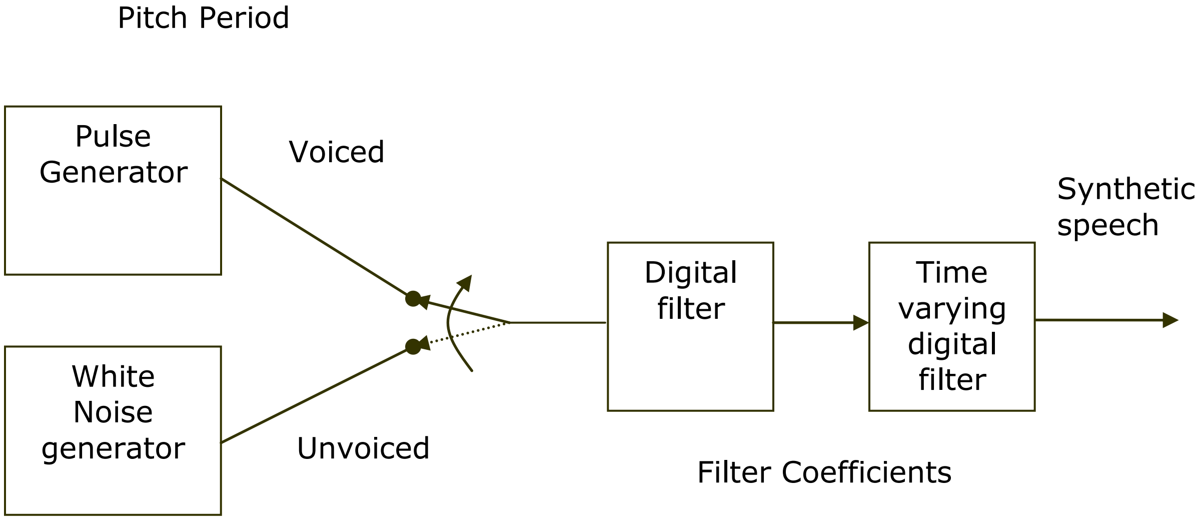
2. LINEAR PREDICTION OF SPEECH SYNTHESIS

Fig shows block diagram of speech synthesizer using linear prediction.

1. For voiced sound, pulse generator is selected as signal source while for unvoiced sounds noise generator is selected as signal source.

2. The linear prediction coefficients are used as coefficients of digital filter. Depending upon these coefficients , the signal is passed and filtered by the digital filter.

3. The low pass filter removes high frequency noise if any from the synthesized speech. Because of linear phase characteristic FIR filters are mostly used as digital filters.



<u>3. SOUND PROCESSING</u>

1. In sound processing application, Music compression(MP3) is achieved by converting the time domain signal to the frequency domain then removing frequencies which are no audible.

2. The time domain waveform is transformed to the frequency domain using a filter bank. The strength of each frequency band is analyzed and quantized based on how much effect they have on the perceived decompressed signal.

3. The DSP processor is also used in digital video disk (DVD) which uses MPEG-2 compression, Web video content application like Intel Indeo, real audio.

4. Sound synthesis and manipulation, filtering, distortion, stretching effects are also done by DSP processor. ADC and DAC are used in signal generation and recording.

4. ECHO CANCELLATION

In the telephone network, the subscribers are connected to telephone exchange by two wire circuit. The exchanges are connected by four wire circuit. The two wire circuit is bidirectional and carries signal in both the directions. The four wire circuit has separate paths for transmission and reception. The hybrid coil at the exchange provides the interface between two wire and four wire circuit which also provides impedance matching between two wire and four wire circuits. Hence there are no echo or reflections on the lines. But this impedance matching is not perfect because it is length dependent. Hence for echo cancellation, DSP techniques are used as follows.

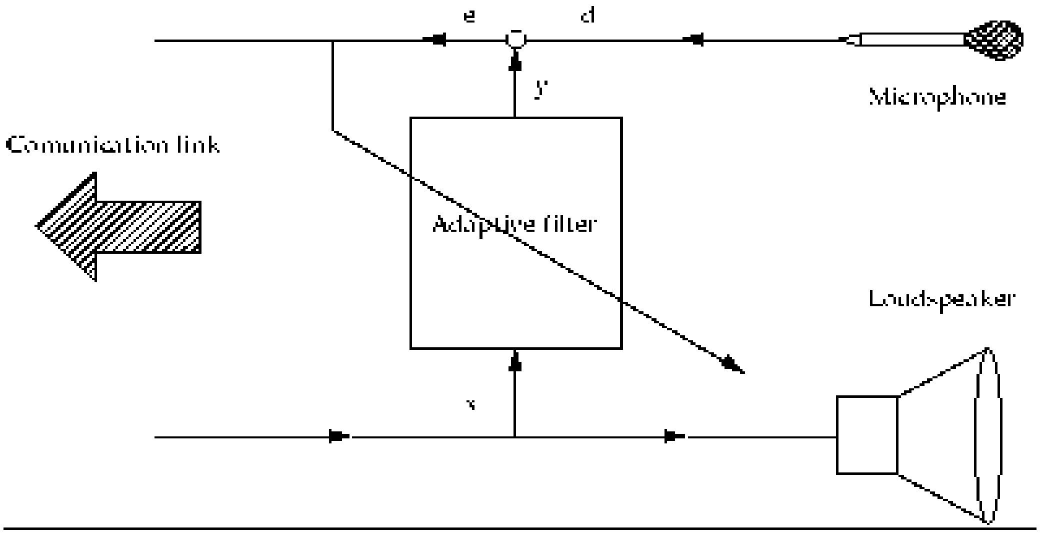


Figure : Echo canceller principle

1. An DSP based acoustic echo canceller works in the following fashion: it records the sound going to the loudspeaker and substract it from the signal coming from the microphone. The sound going through the echo-loop is transformed and delayed, and noise is added, which complicate the substraction process.

2. Let x be the input signal going to the loudspeaker; let d be the signal picked up by the microphone, which will be called the desired signal. The signal after

substraction will be called the error signal and will be denoted by ε. The adaptive filter will try to identify the equivalent filter seen by the system from the loudspeaker to the microphone, which is the transfer function of the room the loudpeaker and microphone are in.

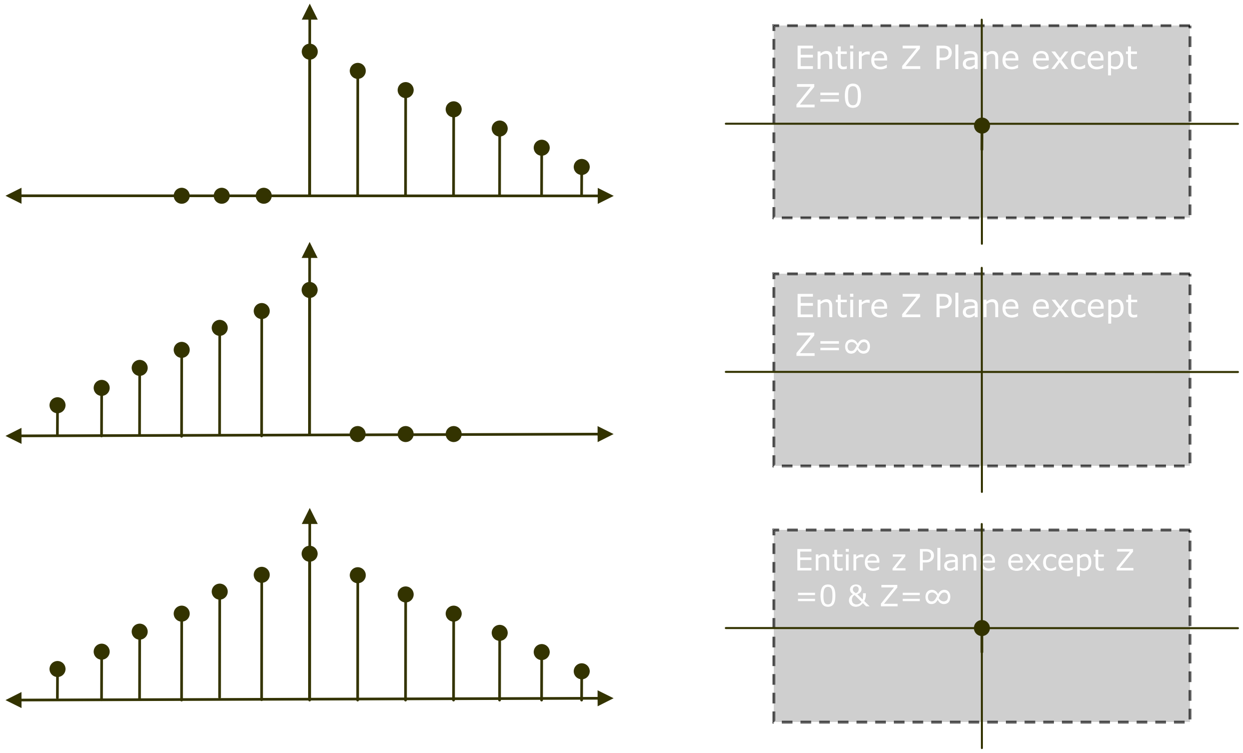
3. This transfer function will depend heavily on the physical characteristics of the environment. In broad terms, a small room with absorbing walls will origninate just a few, first order reflections so that its transfer function will have a short impulse response. On the other hand, large rooms with reflecting walls will have a transfer function whose impulse response decays slowly in time, so that echo cancellation will be much more difficult.

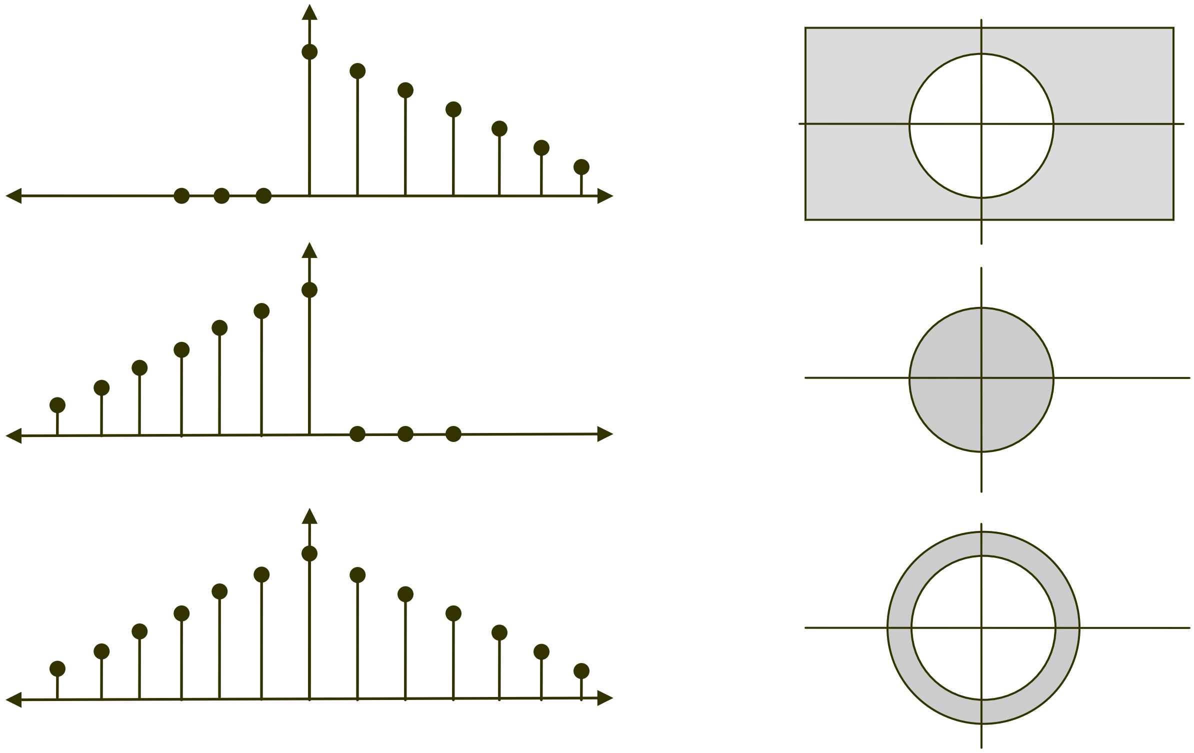
5. VIBRATION ANALYSIS

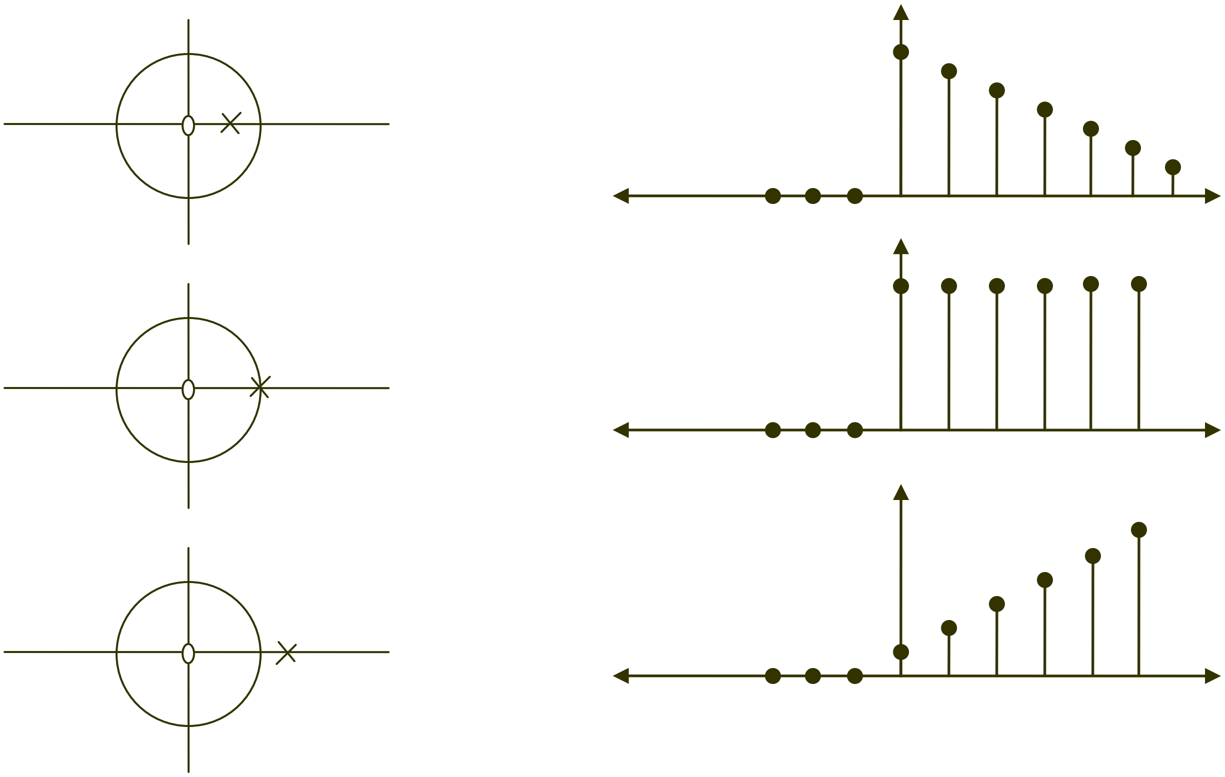
1. Normally machines such as motor, ball bearing etc systems vibrate depending upon the speed of their movements.

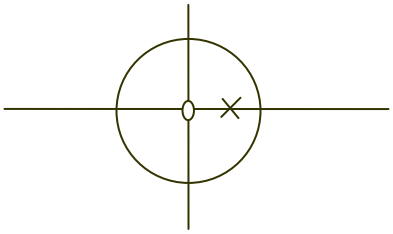
2. In order to detect fault in the system spectrum analysis can be performed. It shows fixed frequency pattern depending upon the vibrations. If there is fault in the machine, the predetermined spectrum is changes. There are new frequencies introduced in the spectrum representing fault.

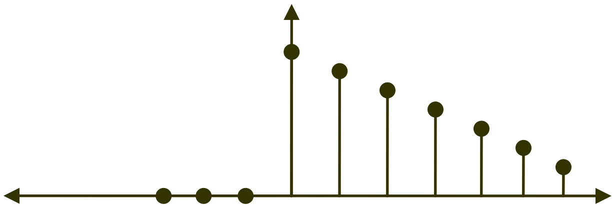
3. This spectrum analysis can be performed by DSP system. The DSP system can also be used to monitor other parameters of the machine simultaneously.

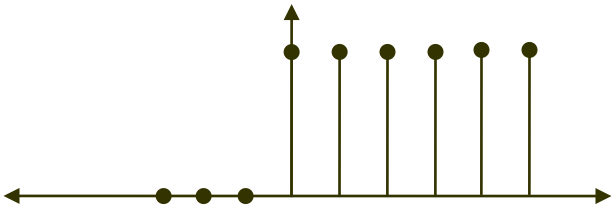

Entire Z Plane except Z=0
Entire Z Plane except Z=∞
Entire z Plane except Z =0 & Z=∞

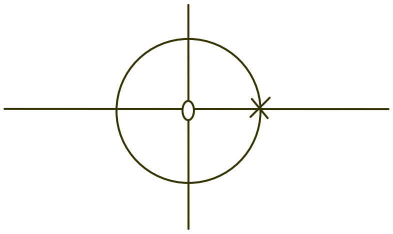


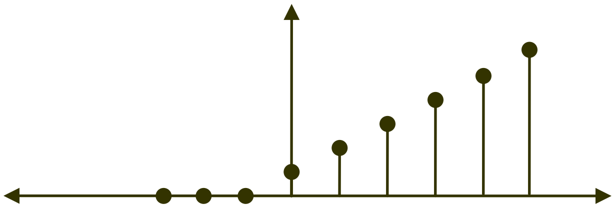


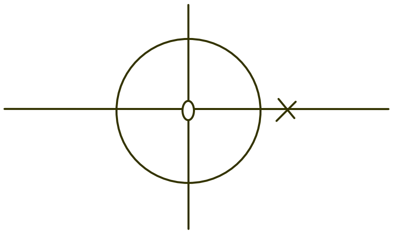




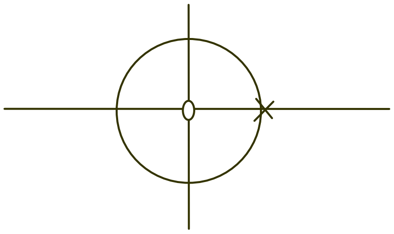


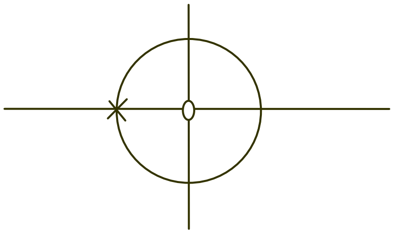






Match the Pairs.





A) -90 constant
B) +90 constant
C) -90 to 0 linear
D) +90 to 0 linear
E) 0 to +90 linear
F) 0 to -90 linear