A Course Material on

DIGITAL SIGNAL PROCESSING

By

DR.VIMALA

PROFESSOR

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

PRATHYUSHA ENGINEERING COLLEGE

EE8491 DISCRETE TIME SYSTEMS AND SIGNAL PROCESSING L T P C3 0 0 3

UNIT I INTRODUCTION 9

Classification of systems: Continuous, discrete, linear, causal, stable, dynamic, recursive, time variance; classification of signals: continuous and discrete, energy and power; mathematical representation of signals; spectral density; sampling techniques, quantization, quantization error, Nyquist rate, aliasing effect.

UNIT II DISCRETE TIME SYSTEM ANALYSIS 9

Z-transform and its properties, inverse z-transforms; difference equation – Solution by z-transform, application to discrete systems - Stability analysis, frequency response – Convolution – Discrete Time Fourier transform , magnitude and phase representation.

UNIT III DISCRETE FOURIER TRANSFORM & COMPUTATION 9

Discrete Fourier Transform- properties, magnitude and phase representation - Computation of DFT using FFT algorithm – DIT &DIF using radix 2 FFT – Butterfly structure.

UNIT IV DESIGN OF DIGITAL FILTERS 9

FIR & IIR filter realization – Parallel & cascade forms. FIR design: Windowing Techniques – Need and choice of windows – Linear phase characteristics. Analog filter design – Butterworth and Chebyshev approximations; IIR Filters, digital design using impulse invariant and bilinear transformation - Warping, pre warping.

UNIT V DIGITAL SIGNAL PROCESSORS 9

Introduction – Architecture – Features – Addressing Formats – Functional modes - Introduction to Commercial DSProcessors.

TOTAL : 45 PERIODS

OUTCOMES:

Ability to understand and apply basic science, circuit theory, Electro-magnetic field theory control theory and apply them to electrical engineering problems.

TEXT BOOKS:

J.G. Proakis and D.G. Manolakis, ‗Digital Signal Processing Principles, Algorithms and Applications‘, Pearson Education, New Delhi, PHI. 2003.

S.K. Mitra, ‗Digital Signal Processing – A Computer Based Approach‘, McGraw Hill Edu, 2013.

Robert Schilling & Sandra L.Harris, Introduction to Digital Signal Processing using Matlab‖, Cengage Learning,2014.

REFERENCES:

Poorna Chandra S, Sasikala. B ,Digital Signal Processing, Vijay Nicole/TMH,2013.

B.P.Lathi, ‗Principles of Signal Processing and Linear Systems‘, Oxford University Press, 2010

Taan S. ElAli, ‗Discrete Systems and Digital Signal Processing with Mat Lab‘, CRC Press, 2009.

Sen M.kuo, woonseng…s.gan, ―Digital Signal Processors, Architecture, Implementations & Applications, Pearson,2013

Dimitris G.Manolakis, Vinay K. Ingle, applied Digital Signal Processing,Cambridge,2012

Lonnie C.Ludeman ,‖Fundamentals of Digital Signal Processing‖,Wiley,2013

DISCRETE SYSTEMS & SIGNAL PROCESSING

CONTENT **PG.NO**

CHAPTER -1

SIGNALS AND SYSTEMS

CHAPTER – 2

DISCRETE TIME SYSTEMS ANALYSIS

CHAPTER – 3

DISCRETE FOURIER TRANSFORM & COMPUTATION

CHAPTER – 4

DESIGN OF DIGITAL FILTERS

CHAPTER – 5

DIGITAL SIGNAL PROCESSOR

CHAPTER 1

SIGNALS AND SYSTEM

1.0 PREREQISTING DISCUSSION

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.

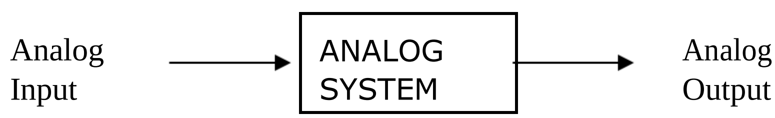
Ex $x(t) = 10t$

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

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

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.



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 this digital signals. The digital output signal from the DSP is given to the DAC.

1.1.1 ADVANTAGES OF DSP OVER ASP

Physical size of analog systems are quite large while digital processors are more compact and light in weight.

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.

Digital system are most flexible as software programs & control programs can be easily modified.

Digital signal can be stores on digital hard disk, floppy disk or magnetic tapes. Hence becomes transportable. Thus easy and lasting storage capacity.

Digital processing can be done offline.

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.
Digital signal processing systems are upgradeable since that are software controlled.
Possibility of sharing DSP processor between several tasks.
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.
Single chip microprocessors, controllers and DSP processors are more versatile and powerful.

1.1.2 Disadvantages Of DSP over ASP

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

1.2 CLASSIFICATION OF SIGNALS

Single channel and Multi-channel signals
Single dimensional and Multi-dimensional signals
Continuous time and Discrete time signals.
Continuous valued and discrete valued signals.
Analog and digital signals.
Deterministic and Random signals
Periodic signal and Non-periodic signal
Symmetrical(even) and Anti-Symmetrical(odd) signal
Energy and Power signal

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.

2) Single 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 are examples of M-D signals.

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 $-\infty$ & 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 techogenerator feedback or Sine or	Microprocessors and computer based systems uses discrete time signals.

	exponential waveforms.	

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.

5) Analog and digital signal

S. 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.

6) Deterministic and Random signals

S. 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 representedordescribedby 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 cannot be evaluated at any instant of time.
4	Example Sine or exponential waveforms.	Example Noise signal or Speech signal

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.

Tutorial problems:

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

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

$$= \frac{+ (\ )}{2}$$

Rearranging the above terms we have

$$= \frac{- (\ )}{2}$$

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

Even component of discrete time signal is given by

$$= \frac{+ (\ )}{2}$$

Odd component of discrete time signal is given by

$$= \frac{- (\ )}{2}$$

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

Find out the even and odd parts of the discrete signal x(n)={2,<u>4</u>,3,2,1}

Find out the even and odd parts of the discrete signal x(n)={<u>2</u>,2,2,2}

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

$$E= \sum^{\infty} (\ )^{2}$$

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

$$= \lim \frac{1}{2+1} \sum^{\infty} {}^{2}(\ )$$

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.

Tutorial problems:

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

Functional Representation

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

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	

Sequence Representation

X(n) = { 0 , 4 , -2 , 4 , 0 ,……}

$\uparrow$
n=0

1.4 STANDARD SIGNAL SEQUENCES

Unit sample signal (Unit impulse signal)

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

Unit step signal

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

Unit ramp signal

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

Exponential signal

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

Sinusoidal waveform

x(n) = A Sin wn

1.5 PROPERTIES OF DISCRETE TIME SIGNALS

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 , -2 , 4 , 0 ,……}
↑
n=0

Delayed by 2 samples : X(n-2)= { 1, -1 , 0 , 4 , -2 , 4 , 0 ,……}
↑
n=0

Advanced by 2 samples : X(n+2) = { 1, -1 , 0 , 4 , -2 , 4 , 0 ,……}
↑
n=0

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 , -2 , 4 , 0}
↑
n=0

Folded signal:

X(-n) = { 0 , 4 , -2 , 4 , 0 , -1 , 1}
↑
n=0

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.

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)

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

1.6 SYMBOLS USED IN DISCRETE TIME SYSTEM

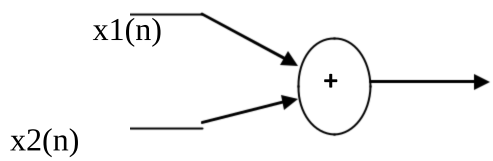
1. Unit delay

$$x(n) \longrightarrow \boxed{Z^{-1}} \longrightarrow y(n) = x(n-1)$$

2. Unit advance

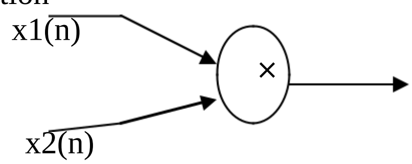
$$x(n) \longrightarrow \boxed{Z^{+1}} \longrightarrow y(n) = x(n+1)$$

3. Addition



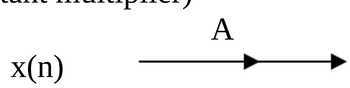
y(n) =x1(n)+x2(n)

Multiplication



y(n) =x1(n)*x2(n)

5. Scaling (constant multiplier)



y(n) = A x(n)

1.7 CLASSIFICATION OF DISCRETE TIME SYSTEMS

1) STATIC v/s DYNAMIC

S. 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 systemsoutput 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.

S. 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

S. 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)

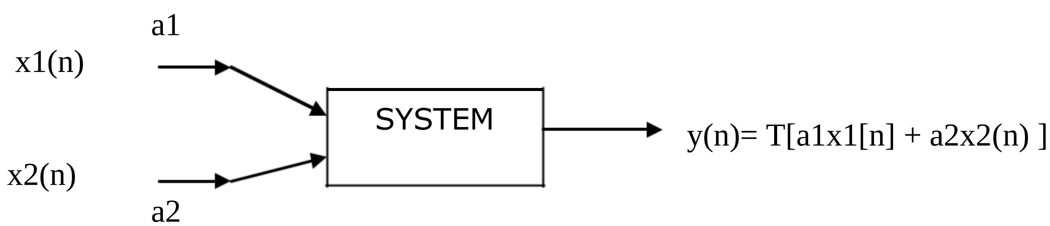
x(n) → y(n)

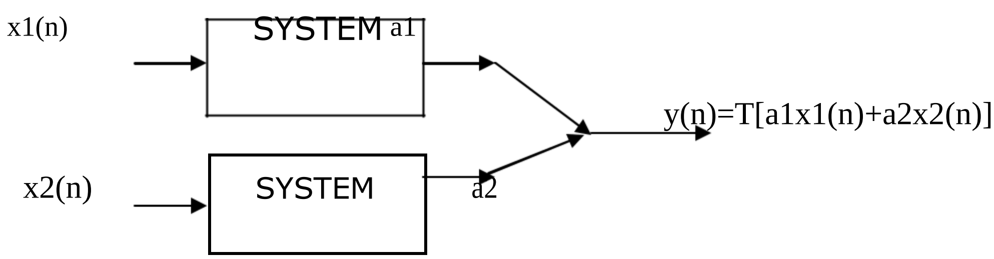
If we delay same input by k units x(n-k) and apply it to same systems, the system produces output y(n-k)

x(n-k) → y(n-k)

3) LINEAR v/s NON-LINEAR SYSTEMS

S.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.

S. 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	$\log_{10}(\lvert x(n)\rvert)$	Non-Linear

4) CAUSAL v/s NON CAUSAL SYSTEMS

S.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.

S.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

S.No	STABLE	UNSTABLE (Stability Property)
1A	System is BIBO stable if every bounded input produces a bounded output.	A System is unstable if any bounded input produces a unbounded output.
	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 $\lvert y(n)\rvert \le M_Y < \infty$	

1.8 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

∞ $=-\infty$ $() < \infty$

S.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

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

CONVOLUTION SUM METHOD
DIFFERENCE EQUATION

1.9.1 LINEAR CONVOLUTION SUM METHOD

This method is powerful analysis tool for studying LSI Systems.

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.

Convolution involves folding, shifting, multiplication and summation operations.

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

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

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

Example 1: $h(n) = \{ 1, \underset{\uparrow}{\underline{2}}, 1, -1 \}$ & $x(n) = \{ \underset{\uparrow}{\underline{1}}, 2, 3, 1 \}$ 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 :	$y(n) = \{1, 4, \underset{\uparrow}{8}, 8, 3, -2, -1\}$

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			
h3	h2x1	h2x2	h2x3

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

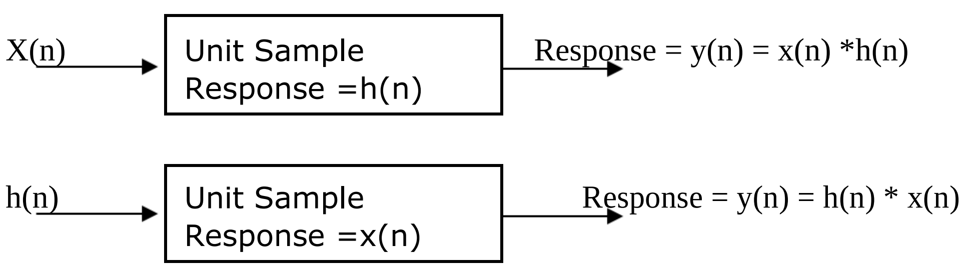
METHOD 4: SIMPLE MULTIPLICATION FORM
X(n)= {x1,x2,x3} & h(n) ={ h1,h2,h3} x1 x2 x3

y(n) = ×

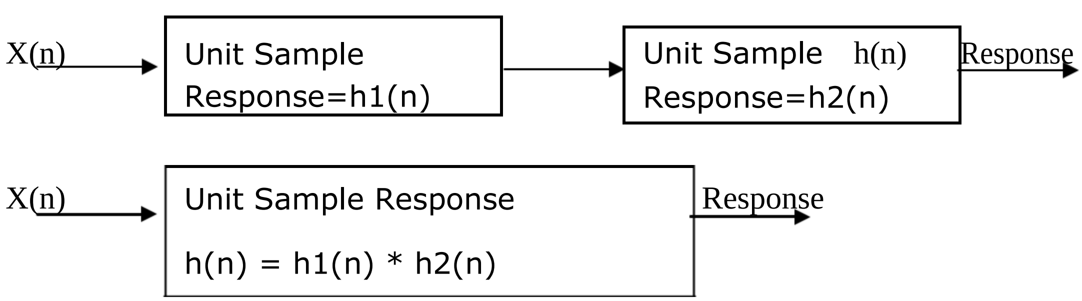
y1 y2 y3

1.9.2 PROPERTIES OF LINEAR CONVOLUTION x(n) = Excitation Input signal
y(n) = Output Response
h(n) = Unit sample response

Commutative Law: (Commutative Property of Convolution)
x(n) * h(n) = h(n) * x(n)



Associate Law: (Associative Property of Convolution) [x(n) * h1(n)] * h2(n) = x(n) * [h1(n) * h2(n)]



Distribute Law: (Distributive property of convolution)
x(n) * [h1(n) + h2(n)] = x(n) * h1(n) + x(n) * h2(n)

1.9.3 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 $n \leq n_0$. 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

$$h(n) = 0 \qquad \text{for } n<0$$

This is the necessary and sufficient condition for causality of the system.

Linear convolution of the causal LSI system is given by

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

1.9.4 STABILITY FOR LTI 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

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

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.

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.

1.10 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

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

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

Correlation is also used in detection and identifying of signals in noise.
Computation of average power in waveforms.
Identification of binary codeword in pulse code modulation system.

DIFFERENCE BETWEEN LINEAR CONVOLUTION AND CORRELATION

S.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 $Y(n) = \alpha\, x(n-D) + \omega(n)$ Where α= Attenuation Factor D= Delay $\omega(n)$ = Noise signal
4	Linear convolution is commutative	Not commutative.

1.10.1 TYPES OF CORRELATION

Under Correlation there are two classes.

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

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

OR

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

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

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

OR

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

1.10.2 PROPERTIES OF CORRELATION

1) The cross-correlation is not commutative.

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

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

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

The autocorrelation sequence is an even function. $r_{xx}(l) = r_{xx}(-l)$

Examples:

Determine cross-correlation sequence

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

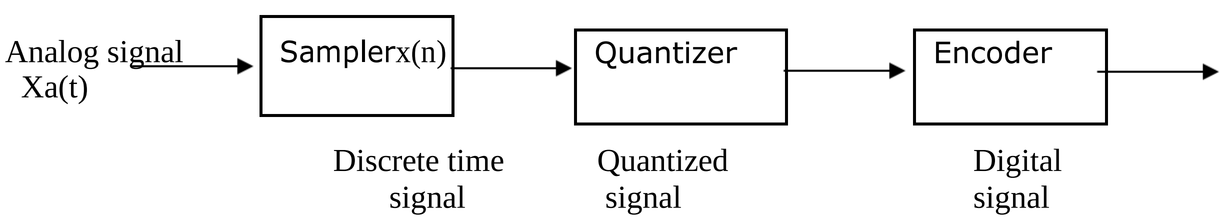
Answer: $r_{xy}(l) = \{10, -9, 19, 36, -14, 33, 0,\underline{7}, 13, -18, 16, -7, 5, -3\}$

Determine autocorrelation sequence

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

1.11 A/D CONVERSION

BASIC BLOCK DIAGRAM OF A/D CONVERTER



1.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.

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

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: $X1(t) = \cos 2\pi\,(10)\,t$ Fs= 40 Hz i.e t= n/Fs

$x1[n] = \cos 2\prod(n/4) = \cos (\prod/2)n$

Case 2: $X1(t) = \cos 2\prod (50) t$ Fs= 40 Hz i.e t= n/Fs

$x1[n] = \cos 2\prod(5n/4) = \cos 2\prod(1+ 1/4)n$

$= \cos (\prod/2)n$

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

2.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 $0 < n < \infty$ & 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 = $X_{Max} - x_{Min}$ / L-1 where L indicates Number of quantization levels.

3.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.

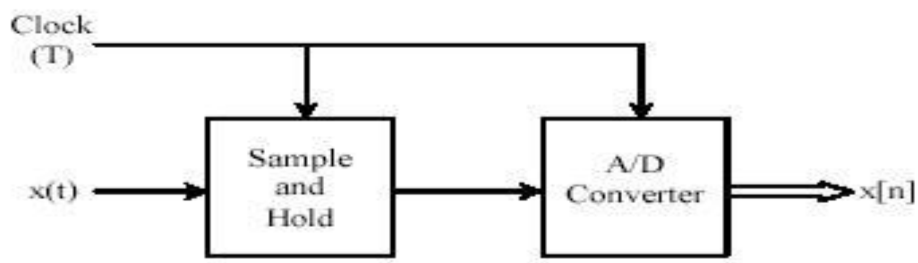
4.ANTI-ALIASING FILTER

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.

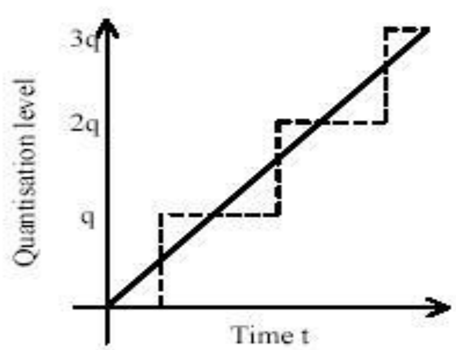
The signal can be band limited by passing it through a filter (LPF) which blocks or attenuates all the frequency components outside the specific bandwidth. Hence called as Anti aliasing filter or pre-filter. (Block Diagram).

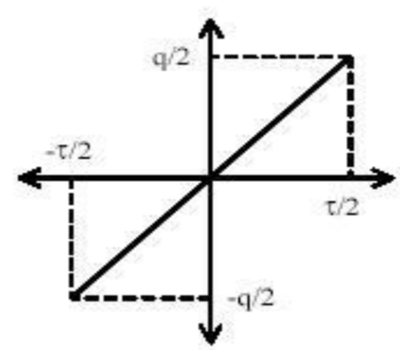
5.SAMPLE-AND-HOLD CIRCUIT:

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.





Tutorial problems:

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

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∏(90)t
X4(t)= cos 2∏(130)t

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

Signal x1(t)=3 cos 600∏t+ 2cos800∏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.

1.12 DIFFERENCE EQUATION

S.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.

Recursive Systems

Non-Recursive Systems

S. 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)

First order Difference Equation

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

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 $\quad y(0) = x(0) + a\, y(-1)$ ….(1)

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

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

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

hence

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

The first part (A) is response depending upon initial condition.
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 $y(-1) \neq 0$
Total response is the sum of zero state response and zero input response.

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

CHAPTER 2

DISCRETE TIME SYSTEM ANALYSIS

2.0 PREREQISTING DISCUSSION ABOUT 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

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

Z transform is an infinite power series because summation index varies from $-\infty$ 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).

2.0.1 ADVANTAGES OF Z TRANSFORM

The DFT can be determined by evaluating z transform.

Z transform is widely used for analysis and synthesis of digital filter.

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.

2.0.2 ADVANTAGES OF ROC(REGION OF CONVERGENCE)

ROC is going to decide whether system is stable or unstable.

ROC decides the type of sequences causal or anti-causal.

ROC also decides finite or infinite duration sequences.

2.1 Z TRANSFORM PLOT

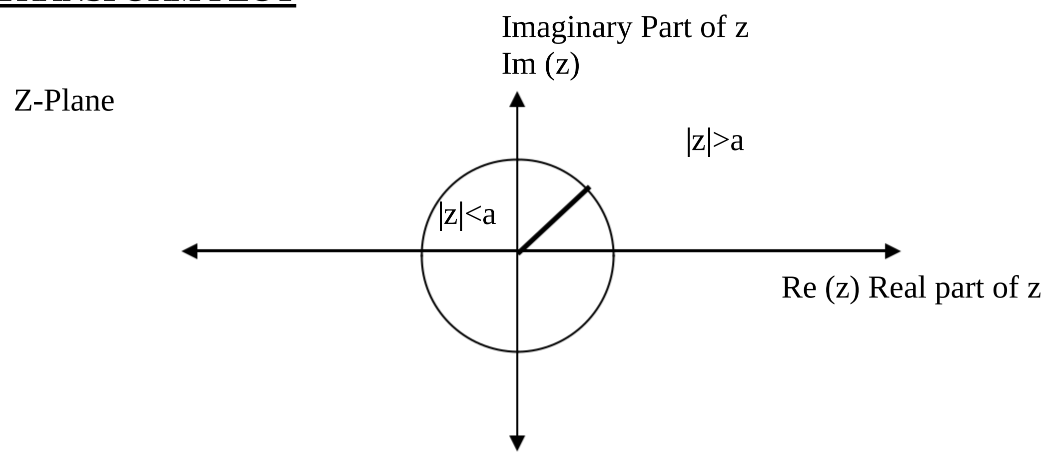


Fig show the plot of z transforms. The z transform has real and imaginary parts. Thus a plot of imaginary part versus real part is called complex z-plane. The radius of circle is 1 called as unit circle.

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 ∟z.

S.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	$\cos(\omega_0 n)$ u(n)		$\frac{1- z^{-1}\cos\omega_0}{1- 2z^{-1}\cos\omega_0+z^{-2}}$	|z| > 1
15	$\sin(\omega_0 n)$ u(n)		$\frac{z^{-1}\sin\omega_0}{1- 2z^{-1}\cos\omega_0+z^{-2}}$	|z| > 1
16	$a^n \cos(\omega_0 n)$ 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	$a^n \sin(\omega_0 n)$ u(n)	Time scaling	$\frac{(z/a)^{-1}\sin\omega_0}{1- 2(z/a)^{-1}\cos\omega_0+(z/a)^{-2}}$	|z| > |a|

Tutorial problems:

Determine z transform of following signals. Also draw ROC. i) $x(n) = \{\underline{1},2,3,4,5\}$

ii) $x(n)=\{1,2,3,\underline{4},5,0,7\}$

Determine z transform and ROC for $x(n) = (-1/3)^n u(n) –(1/2)^n u(-n-1)$.

Determine z transform and ROC for $x(n) = [3.(4^n)–4(2^n)] u(n)$.

Determine z transform and ROC for $x(n) = (1/2)^n u(-n)$.

Determine z transform and ROC for $x(n) = (1/2)^n \{u(n) – u(n-10)\}$.

Find linear convolution using z transform. X(n)={1,2,3} & h(n)={1,2}

2.1.1 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

$$a1\, x1(n) + a2\, x2(n) \xleftrightarrow{z} a1\, X1(z) + a2\, X2(z)$$

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

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

Then $$x(n-k) \xleftrightarrow{z} X(z)\, z^{-k}$$

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

Scaling in z domain

This property states that if

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

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

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

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

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

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

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

Then $$n\, x(n) \xleftrightarrow{z} -z\, d/dz\, (X(z))$$

6) Convolution Theorem

The Circular property states that if

$$x_1(n) \overset{z}{\longleftrightarrow} X_1(z) \text{ And}$$

$$x_2(n) \overset{z}{\longleftrightarrow} X_2(z) \text{ Then}$$

$$\text{Then } x_1(n) * x_2(n) \underset{N}{\overset{z}{\longleftrightarrow}} X_1(z)\, X_2(z)$$

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

$$x_1(n) \overset{z}{\longleftrightarrow} X_1(z) \text{ And}$$

$$x_2(n) \overset{z}{\longleftrightarrow} X_2(z) \text{ Then}$$

$$\text{then } \sum_{n=-\infty}^{\infty} x_1(l)\, x_2(-l) \overset{z}{\longleftrightarrow} X_1(z)\, x_2(z^{-1})$$

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

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

2.2 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

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

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

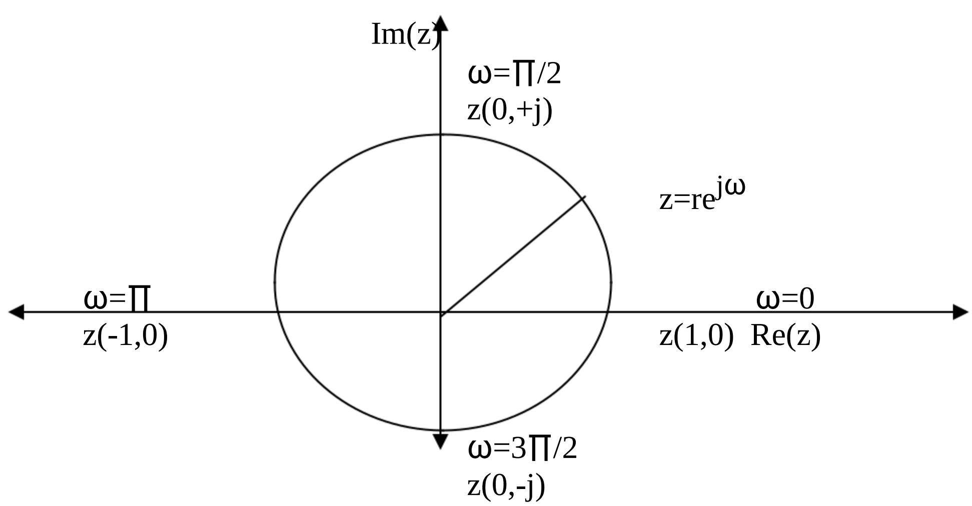
$$X(z)\Big|_{z=e^{jw}} = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j\omega n}$$

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

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

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

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



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

2.3 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.

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

PARTIAL FRACTION EXPANSION METHOD

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

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

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)

S.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	$\dfrac{1+2z^{-1}+z^{-2}}{1-3/2\,z^{-1}+0.5z^{-2}}$	$2\delta(n)+8(1)^n u(n)-9(0.5)^n u(n)$ for $\|z\|>1$	causal system
6	$\dfrac{1+z^{-1}}{1-z^{-1}+0.5z^{-2}}$	$(1/2-j3/2)(1/2+j1/2)^n u(n)+$ $(1/2+j3/2)(1/2+j1/2)^n u(n)$	causal system
7	$\dfrac{1-(0.5)z^{-1}}{1-3/4\,z^{-1}+1/8\,z^{-2}}$	$4(-1/2)^n u(n)-3(-1/4)^n u(n)$ for $\|z\|>1/2$	causal system
8	$\dfrac{1-1/2\,z^{-1}}{1-1/4\,z^{-2}}$	$(-1/2)^n u(n)$ for $\|z\|>1/2$	causal system
9	$\dfrac{z+1}{3z^2-4z+1}$	$\delta(n)+u(n)-2(1/3)^n u(n)$ for $\|z\|>1$	causal system
10	$\dfrac{5z}{(z-1)(z-2)}$	$5(2^n-1)$ for $\|z\|>2$	causal system
11	$\dfrac{z^3}{(z-1)(z-1/2)^2}$	$4-(n+3)(1/2)^n$ for $\|z\|>1$	causal system

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

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

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) ={ ….. ,x(-2),x(-1),x(0),x(1),x(2),…………..}.

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

SOLVE USING —POWER SERIES EXPANSION— METHOD

S.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	$\frac{1}{1-1.5z^{-1}+0.5z^{-2}}$	{$\underline{1}$,3/2,7/4,15,8,……….} For |z| > 1 {….14,6,2,0,$\underline{0}$} For |z| < 0.5
3	$\frac{z^2+z}{z^3-3z^2+3z-1}$	{0,1,4,9,…..} For |z| > 3
4	$z^2(1-0.5z^{-1})(1+z^{-1})(1-z^{-1})$	X(n) ={1,-0.5,$\underline{-1}$,0.5}
5	$\log(1+az^{-1})$	$(-1)^{n+1} a^n/n$ for $n\geq 1$ and |z|>|a|

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

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

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

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

2.4 POLE –ZERO PLOT

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.

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.

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)

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

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

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

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

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

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 $|z| = 1$

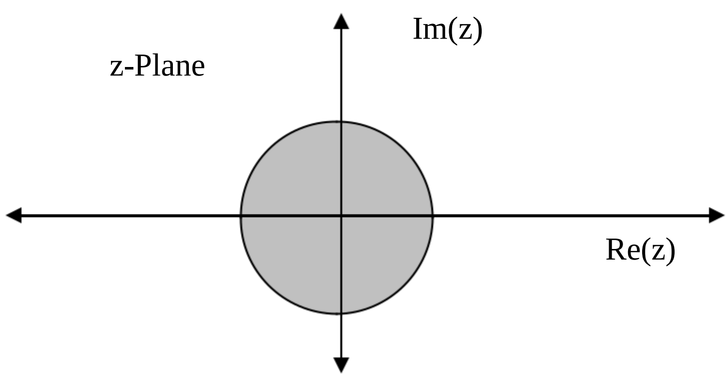


Fig: Stable system

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

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

STANDARD INVERSE Z TRANSFORMS

S. 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}	$\delta(n-k)$	$\delta(n-k)$

2.5 ONE SIDED Z TRANSFORM

S.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

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

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

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

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

2.6 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.

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

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

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

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

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

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 is sequences. It also used to relate system function H(z) and Z transform.

The transfer function $H(\omega)$ 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.

Tutorial problems:

A difference equation of the system is given below $Y(n)= 0.5\, y(n-1) + x(n)$

Determine a) System function
Pole zero plot
Unit sample response

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)
System Function
Pole zero plot
Response of system to the input x(n) = nu(n)
Is the system stable? Comment on the result.

A difference equation of the system is given below Y(n)= 0.5 x(n) + 0.5 x(n-1)

Determine a) System function
Pole zero plot
Unit sample response
Transfer function
Magnitude and phase plot

A difference equation of the system is given below
Y(n)= 0.5 y(n-1) + x(n) + x(n-1)
Y(n)= x(n) + 3x(n-1) + 3x(n-2) + x(n-3)
System Function
Pole zero plot
Unit sample response
Find values of y(n) for n=0,1,2,3,4,5 for x(n)= δ(n) for no initial condition.

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.

Solve second order difference equation x(n+2) + 3x(n+1) + 2x(n) with x(0)=0 and x(1)=1.

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.

2.7 JURY'S STABILITY CRITERIA / ALGORITHM:

Jury's stability algorithm says

Form the first rows of the table by writing the coefficients of D(z).

B_0 B_1 B_2 ---------B_N

B_N B_{N-1} B_{N-2} --------- B_0

Form third and fourth rows of the table by evaluating the determinant C_J

$$\begin{vmatrix} B_0 & B_{N-J} \\ B_N & B_J \end{vmatrix}$$

This process will continue until you obtain 2N-3 rows with last two having 3 elements. Y_0, Y_1, Y_2

A digital filter with a system function H(z) is stable, if and only if it passes the following terms.

a. $D(Z)|_{Z=1} > 0$

$(-1)^N D(Z)|_{Z=-1} > 0$

$|b_0| > |b_N|$, $|C_0|$

CHAPTER 3

DISCRETE FOURIER TRANSFORM AND COMPUTATION

3.1 PREREQISTING DISCUSSION

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

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

3.2 DIFFERENCE BETWEEN FT & DFT

S. 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

Tutorial problems:

Prove that FT x(ω) is periodic with period 2∏.
Determine FT of $x(n)= a^n u(n)$ for $-1< a < 1$.
Determine FT of x(n)= A for $0 \le n \le L-1$.

Determine FT of x(n)= u(n)

Determine FT of x(n)= δ(n)

Determine FT of $x(n)= e^{-at} u(t)$

3.3 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)

S. 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	- ∏/2	0	-j	-j
3	W_4^2	- ∏	-1	0	-1
4	W_4^3	-3∏/2	0	J	J

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

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

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

EIGHT POINT DFT (8-DFT)

S. 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}	-2π	$1/\sqrt{2}$	$+j\,1/\sqrt{2}$	$1/\sqrt{2} + j\,1/\sqrt{2}$

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 $x(k) = \tan^{-1} (X_I(k) / X_R(k))$

Tutorial problems:

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]

3.4 DIFFERENCE BETWEEN DFT & IDFT

S.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 $$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi kn/N}$$	Mathematical Equation to calculate IDFT is given by $$x(n) = 1/N \sum_{n=0}^{N-1} X(k) e^{j2\pi kn/N}$$
4	Thus DFT is given by **$X(k)= [W_N][x_n]$**	In DFT and IDFT difference is of factor 1/N & sign of exponent of twiddle factor. Thus **$x(n)= 1/N\, [W_N]^{-1}[X_K]$**

3.5 PROPERTIES OF DFT

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

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

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

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

Then

$$a1\, x1(n) + a2\, x2(n) \overset{DFT}{\underset{N}{\longleftrightarrow}} a1\, X1(k) + a2\, X2(k)$$

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

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)

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

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

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

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

5. Circular Convolution

The Circular Convolution property states that if

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

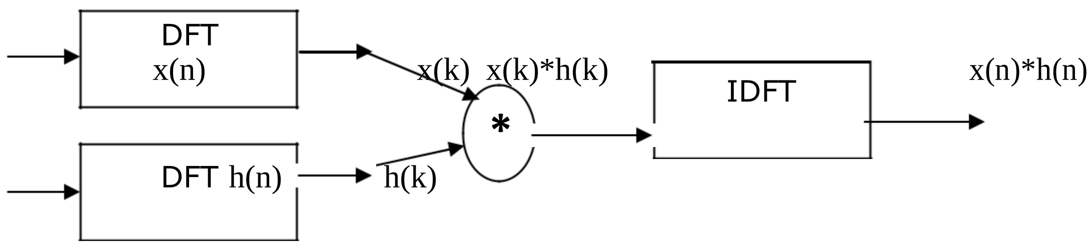
$$x2(n) \xleftrightarrow[N]{DFT} X2(k) \text{ Then}$$

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

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

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

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
- Graphical representation form
- Matrix approach

<u>DIFFERENCE BETWEEN LINEAR CONVOLUTION & CIRCULAR CONVOLUTION</u>

S. 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 $y(n) = \sum_{k=-\infty}^{\infty} x(k)\, h(n-k)$	Circular Convolution is calculated as $y(m) = \sum_{n=0}^{N-1} x1(n)\, x2(m-n)_N$
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.

Tutorial problems:

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} 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.

6. Multiplication

The Multiplication property states that if

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

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

Then $x1(n)\, x2(n) \underset{N}{\overset{DFT}{\longleftrightarrow}} 1/N\ x1(k)\ \ x2(k)$

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

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

Then $x((-n))_N = x(N-n) \underset{N}{\overset{DFT}{\longleftrightarrow}} x((-k))_N = x(N-k)$

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

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

Then $x((n-l))_N \underset{N}{\overset{DFT}{\longleftrightarrow}} x(k)\, e^{-j2\pi k l / N}$

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

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

$$\text{Then} x(n)\, e^{j2 \prod l\, n / N} \underset{N}{\overset{DFT}{\longleftrightarrow}} x((n-l))_N$$

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

10. Complex conjugate property

The Complex conjugate property states that if

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

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

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

11. Circular Correlation

The Complex correlation property states

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

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

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

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.

3.6 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)}

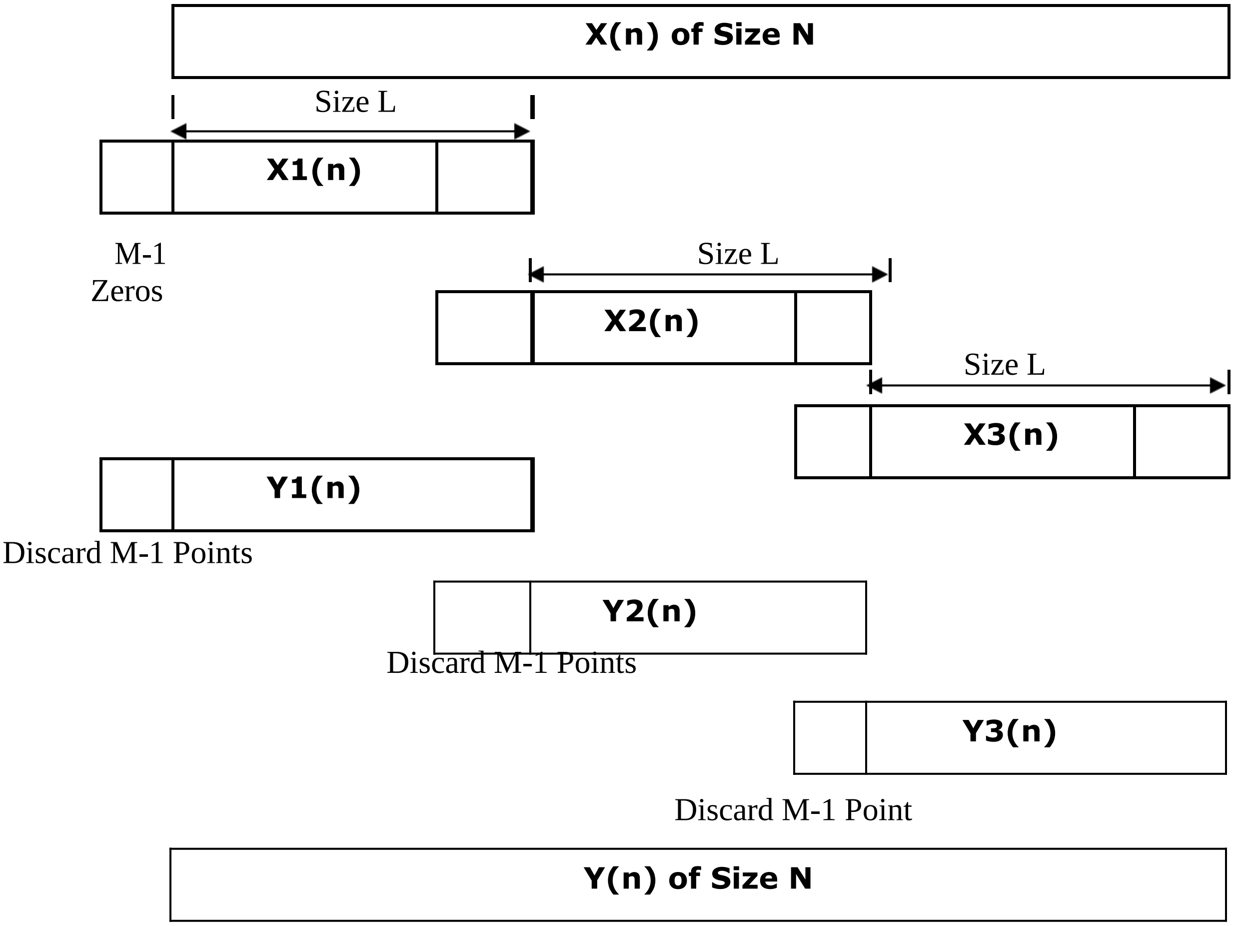
Step2> Unit sample response h(n) contains M samples hence its length is made N by padding zeros.
Thus h(n) also contains N samples.

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 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 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.

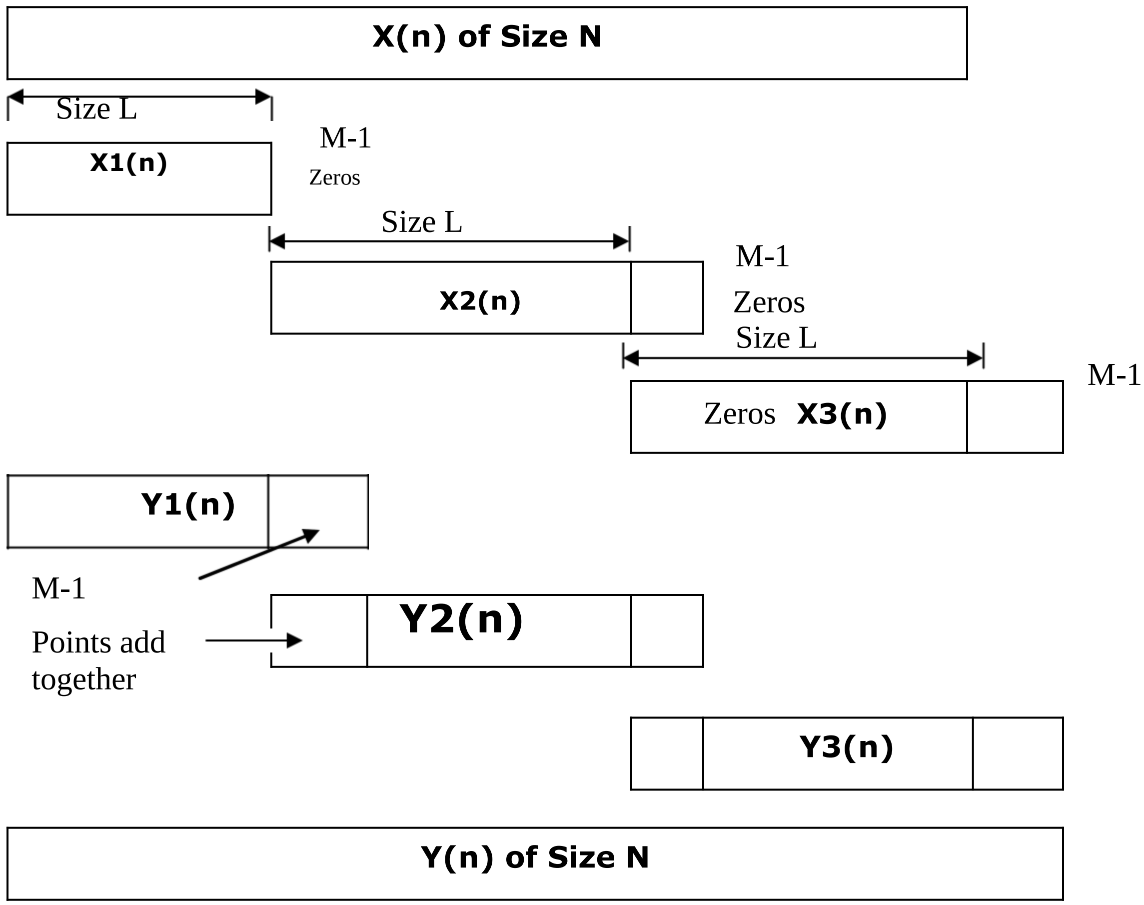
h(n)={ h(0), h(1), …………….h(M-1), 0,0,0,……………………(L-1 zeros)}

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

S. 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 $F_s \geq 2 F_{max}$. 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 2Π 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.

3.7 FAST FOURIER ALGORITHM (FFT)

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).

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.

3.7.1 RADIX-2 FFT ALGORITHMS

1. DECIMATION IN TIME (DITFFT)

There are three properties of twiddle factor W_N

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

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

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

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

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

$$X(k) = F1(k) + W_N^{k}\, F2(k) \qquad (3)$$

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

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

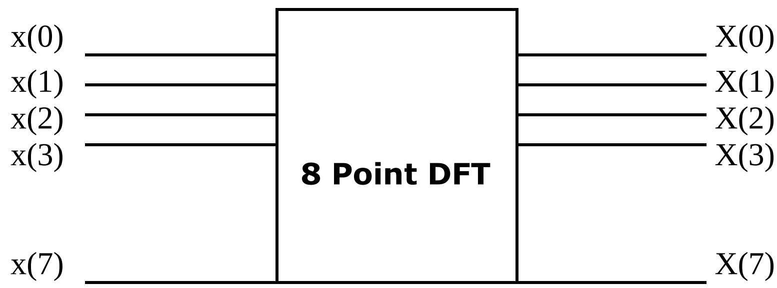


Fig 1. DIRECT COMPUTATION FOR N=8

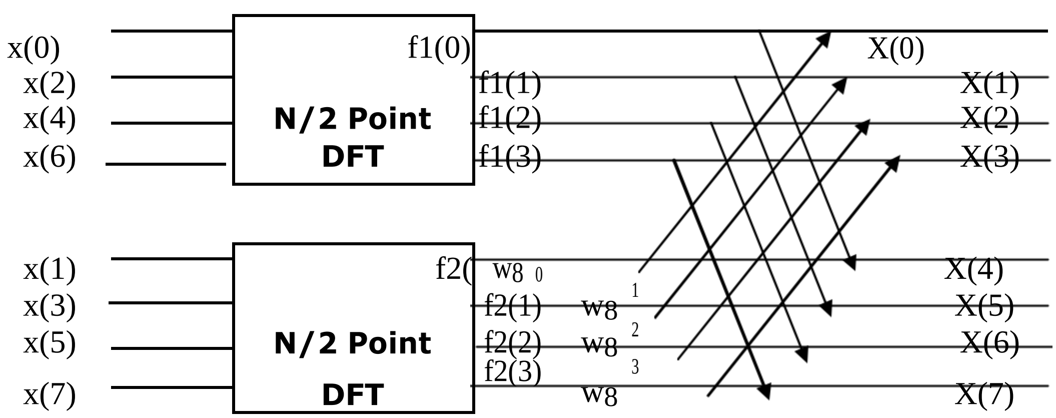


Fig 2. FIRST STAGE FOR FFT COMPUTATION FOR N=8

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

$$g1(k) = P1(k) + W_N^{2k}\, P2(k) \quad \text{-------} \qquad (5)$$

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

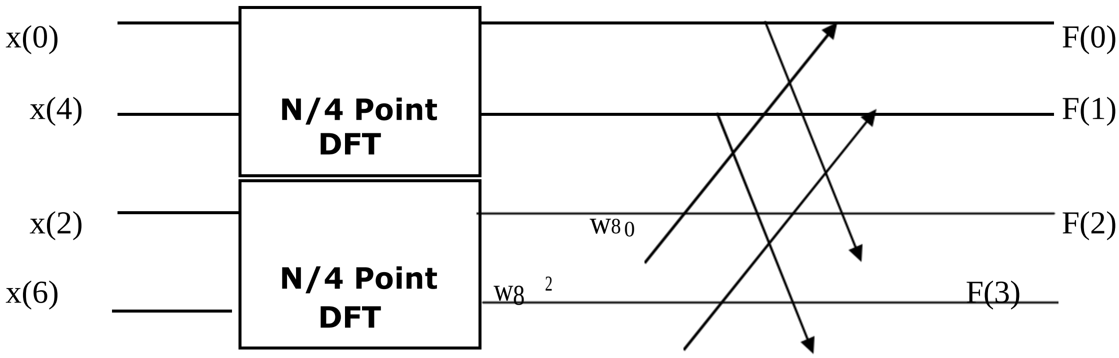


Fig 3. SECOND STAGE FOR FFT COMPUTATION FOR N=8

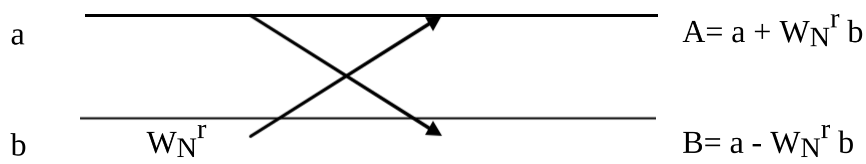


Fig 4. BUTTERFLY COMPUTATION (THIRD STAGE)

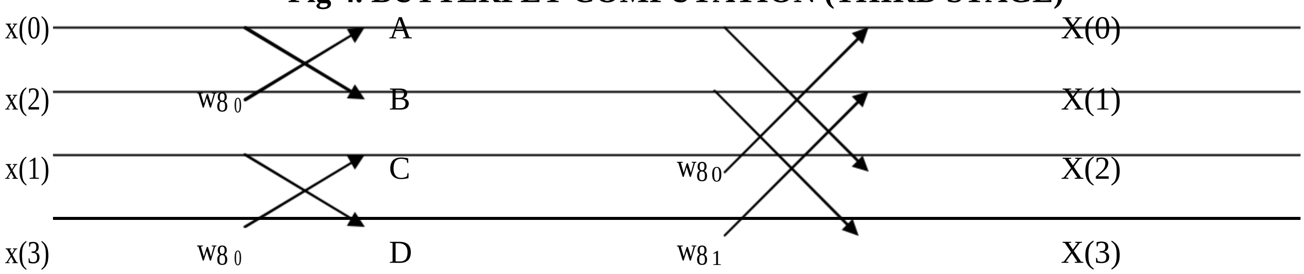


Fig 5. SIGNAL FLOW GRAPH FOR RADIX- DIT FFT N=4

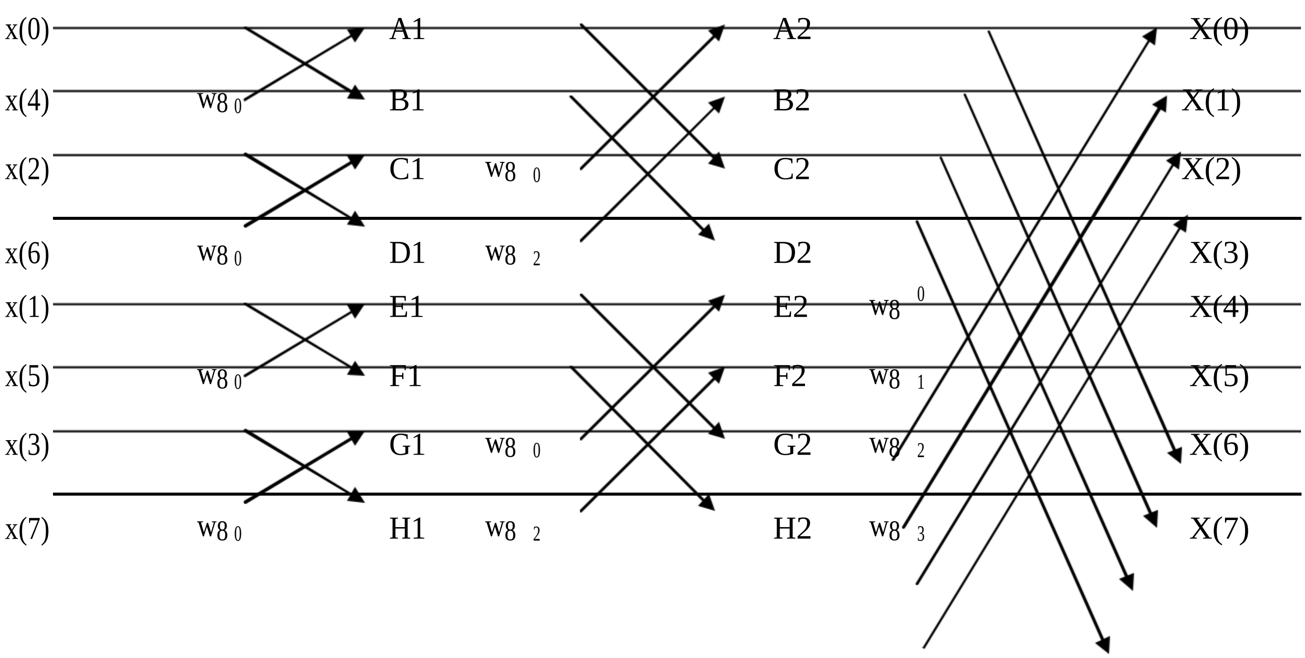


Fig 6. SIGNAL FLOW GRAPH FOR RADIX- DIT FFT N=8

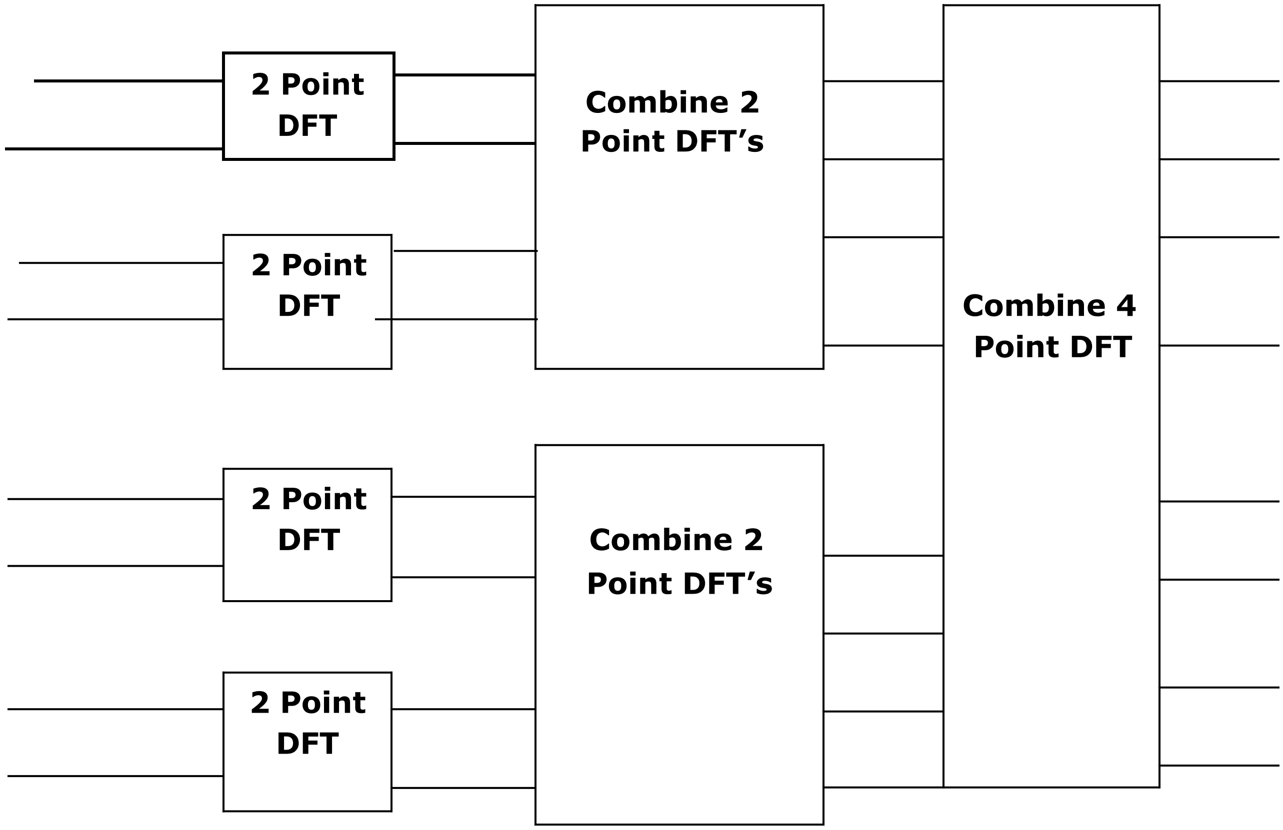


Fig 7. BLOCK DIAGRAM FOR RADIX- DIT FFT N=8

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

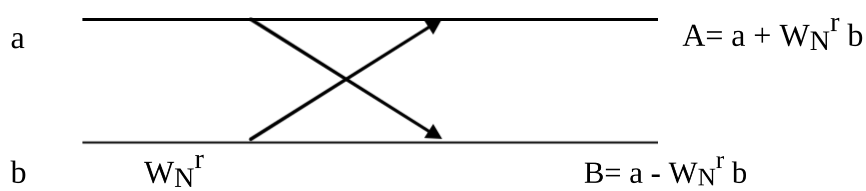


Fig. BUTTERFLY COMPUTATION

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.

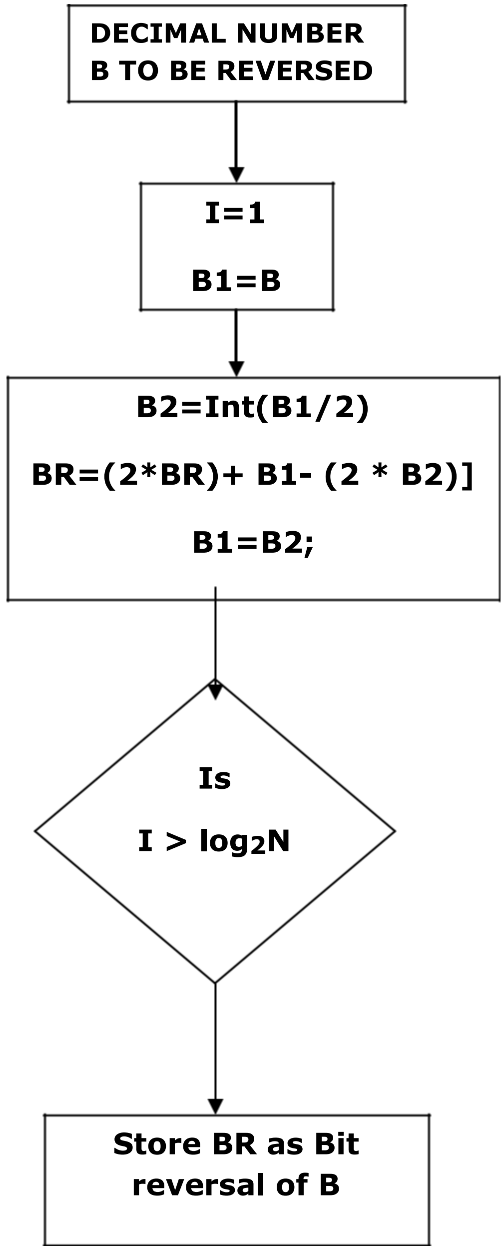
Thus for computation of one butterfly, four memory locations are required for storing two complex numbers A and B. In every stage there are N/2 butterflies hence total 2N memory locations are required. 2N locations are required for each stage. Since stages are computed successively these memory locations 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,



FLOW CHART

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

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

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

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

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

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.

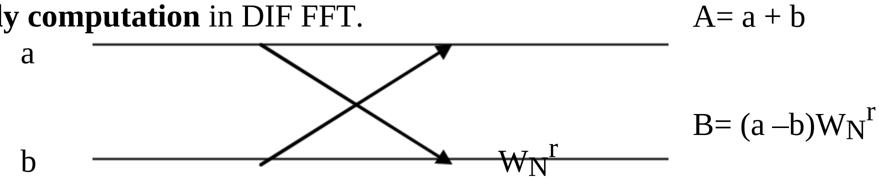


Fig 1. BUTTERFLY COMPUTATION

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

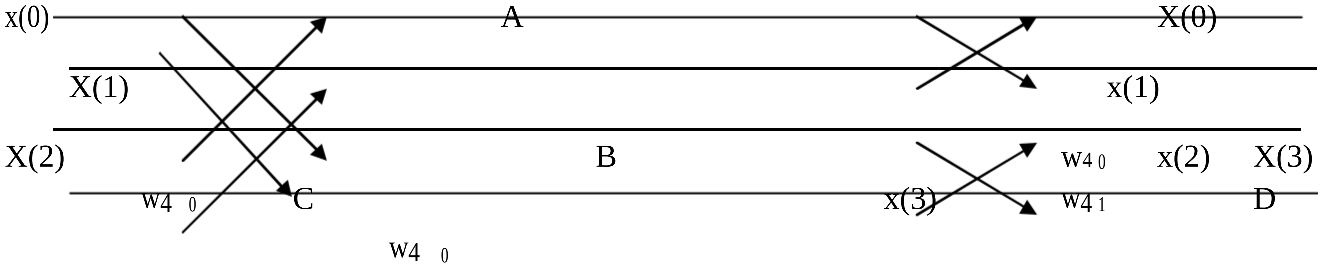


Fig 2. SIGNAL FLOW GRAPH FOR RADIX- DIF FFT N=4

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

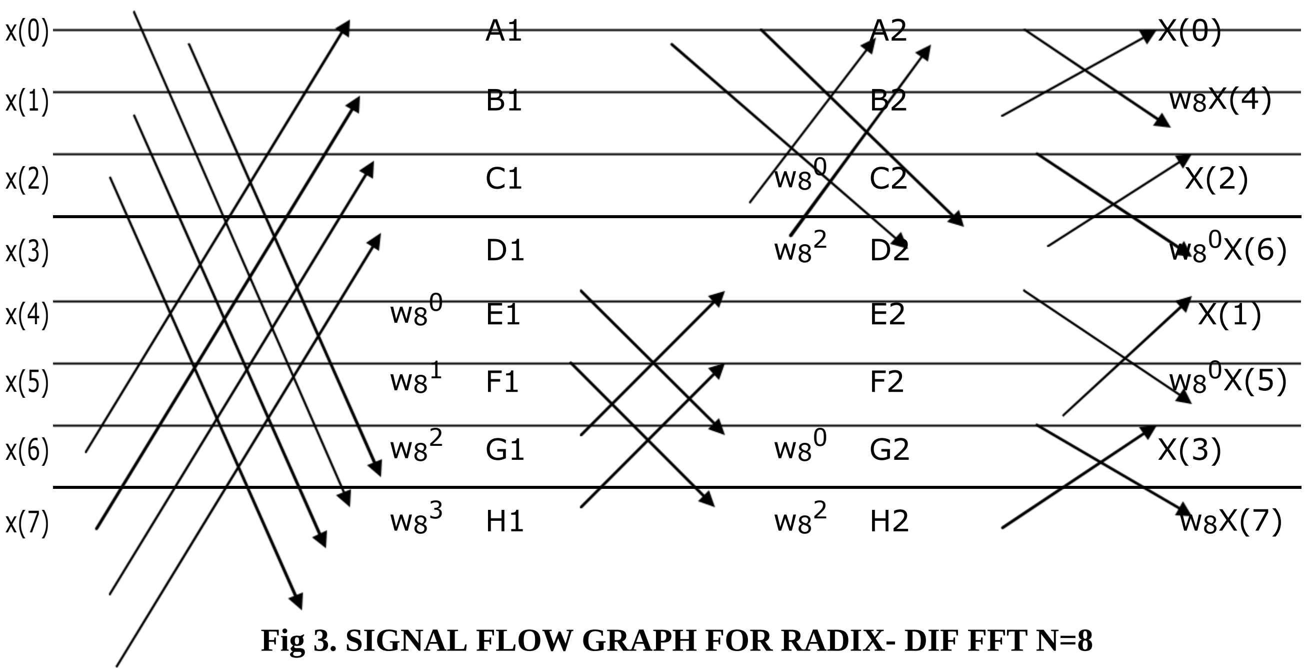


Fig 3. SIGNAL FLOW GRAPH FOR RADIX- DIF FFT N=8

<u>DIFFERENCE BETWEEN DITFFT AND DIFFFT</u>

S. No	DIT FFT	DIF FFT
1	DITFFT algorithms are based upon decomposition of the input sequence into smaller and smaller sub sequences.	DIFFFTalgorithmsarebasedupon 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

S. 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

Tutorial problems:

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

3.8 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

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

$$X(n) \longrightarrow \boxed{h_k(n) = W_N^{-kn}\, u(n)} \longrightarrow y_k(n)$$

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.

CHAPTER 4

DESIGN OF DIGITAL FILTERS

4.1 PREREQISTING DISCUSSION

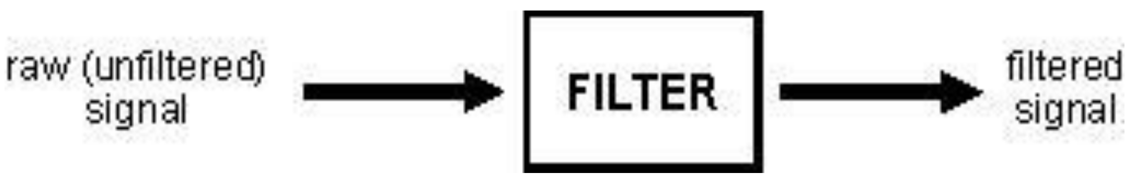
To remove or to reduce strength of unwanted signal like noise and to improve the quality of required signal filtering process is used. To use the channel full bandwidth we mix up two or more signals on transmission side and on receiver side we would like to separate it out in efficient way. Hence filters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

Removal of undesirable noise from the desired signals
Equalization of communication channels
Signal detection in radar, sonar and communication
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.

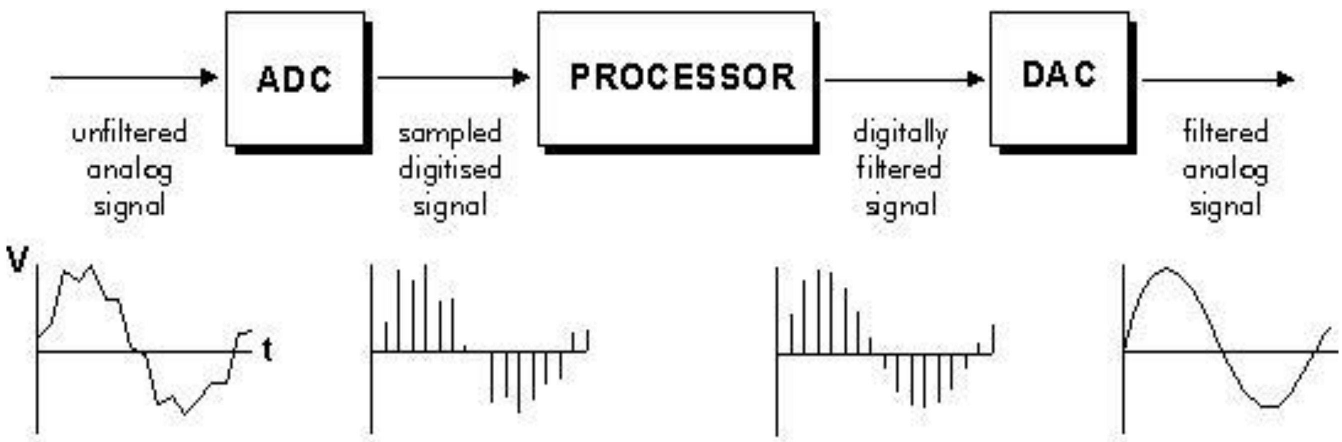
In analog filters the signal being filtered is an electrical voltage or current which is the direct analogue of the physical quantity (e.g. a sound or video signal or transducer output) involved.

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.

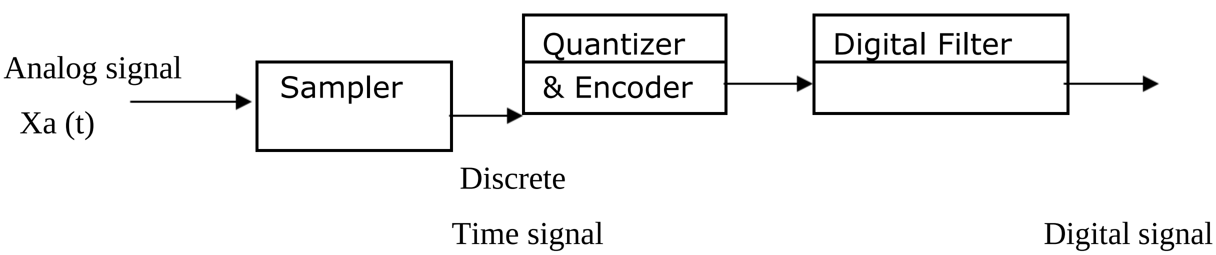
The analog input signal must first be sampled and digitized using an ADC (analog to digital converter). The resulting binary numbers, representing successive sampled values of the input signal, are transferred to the processor, which carries out numerical calculations on them. These calculations typically 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



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.

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.

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

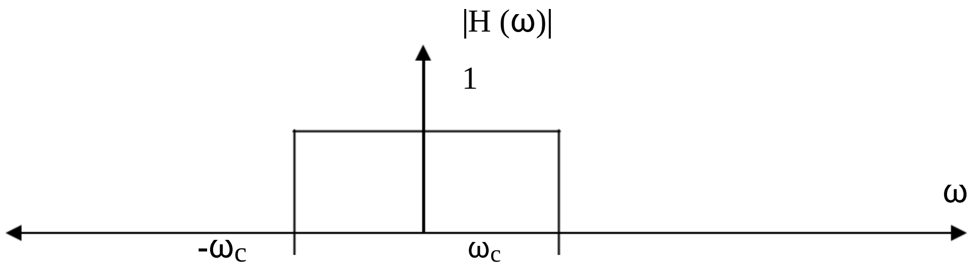
S. No	Analog Filter	Digital Filter
1	Analog filters are used for filtering analog signals.	Digital filters are used for filtering digital sequences.
2	Analog filters are designed with various components like resistor, inductor and capacitor	Digital Filters are designed with digital hardware like FF, 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.

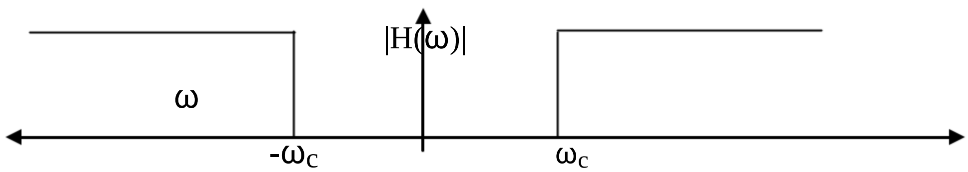
1.Lowpass Filter

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.



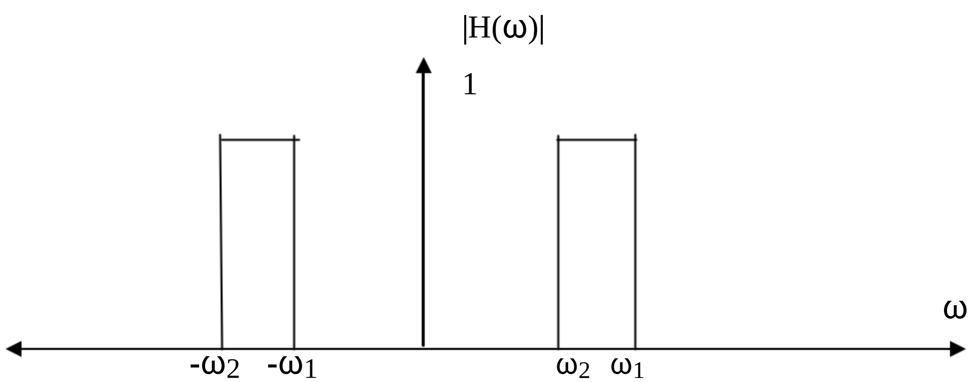
2.Highpass Filter

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.



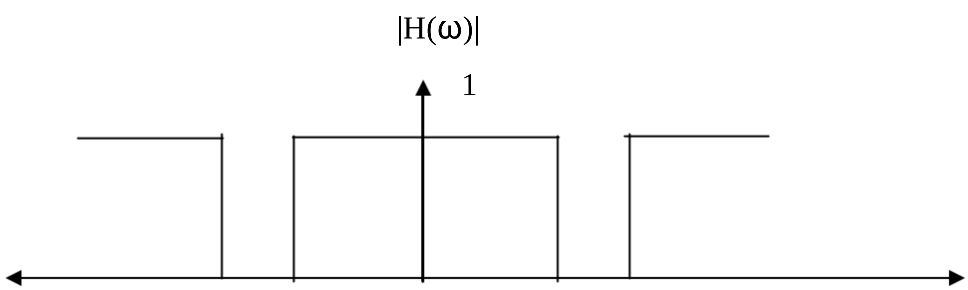
3.Bandpass Filter

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.



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.



ω

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.

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

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

Ideal filters have a constant gain (usually taken as unity gain) passband characteristic and zero gain in their stop band.

Ideal filters have a linear phase characteristic within their passband.

Ideal filters also have constant magnitude characteristic.

Ideal filters are physically unrealizable.

4.2 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

S. 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 $n \geq M$ 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	IIR systems are recursive. Thus they use

	FIR filter depends upon present and past inputs.	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 $n\geq 0$

4. 3 STRUCTURES FOR FIR SYSTEMS

FIR Systems are represented in four different ways

Direct Form Structures

Cascade Form Structure

Frequency-Sampling Structures

Lattice structures.

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,

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

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

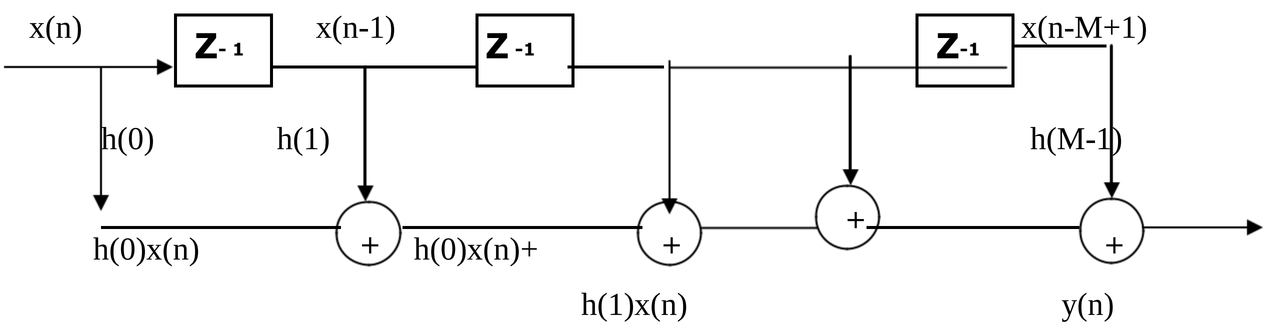


FIG - DIRECT FORM REALIZATION OF FIR SYSTEM

There are M-1 unit delay blocks. One unit delay block requires one memory location.

Hencedirect form structure requires M-1 memory location

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.

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

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

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

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

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

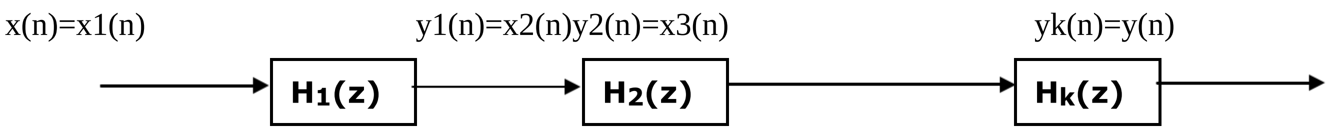


FIG- CASCADE FORM REALIZATION OF FIR SYSTEM

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

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

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

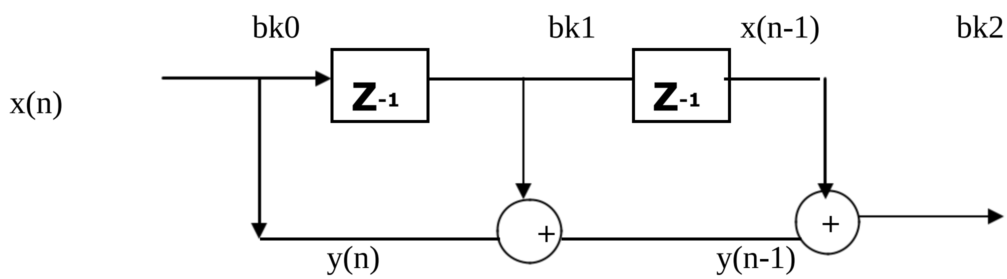


FIG - DIRECT FORM REALIZATION OF FIR SECOND ORDER SYSTEM

4. 4 STRUCTURES FOR IIR SYSTEMS

IIR Systems are represented in four different ways

- Direct Form Structures Form I and Form II
- Cascade Form Structure
- Parallel Form Structure
- 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

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

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).

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 numberof additions and M+N+1 number of memory locations.

DIRECT FORM – I

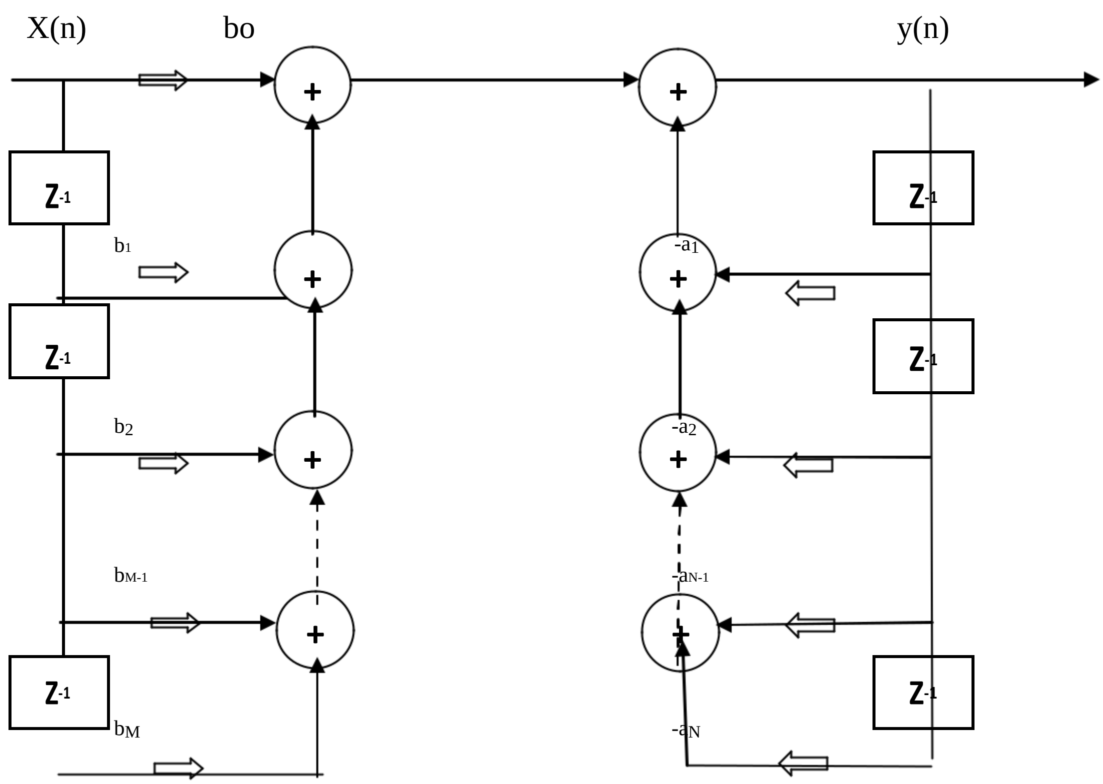


FIG - DIRECT FORM I REALIZATION OF IIR SYSTEM

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.

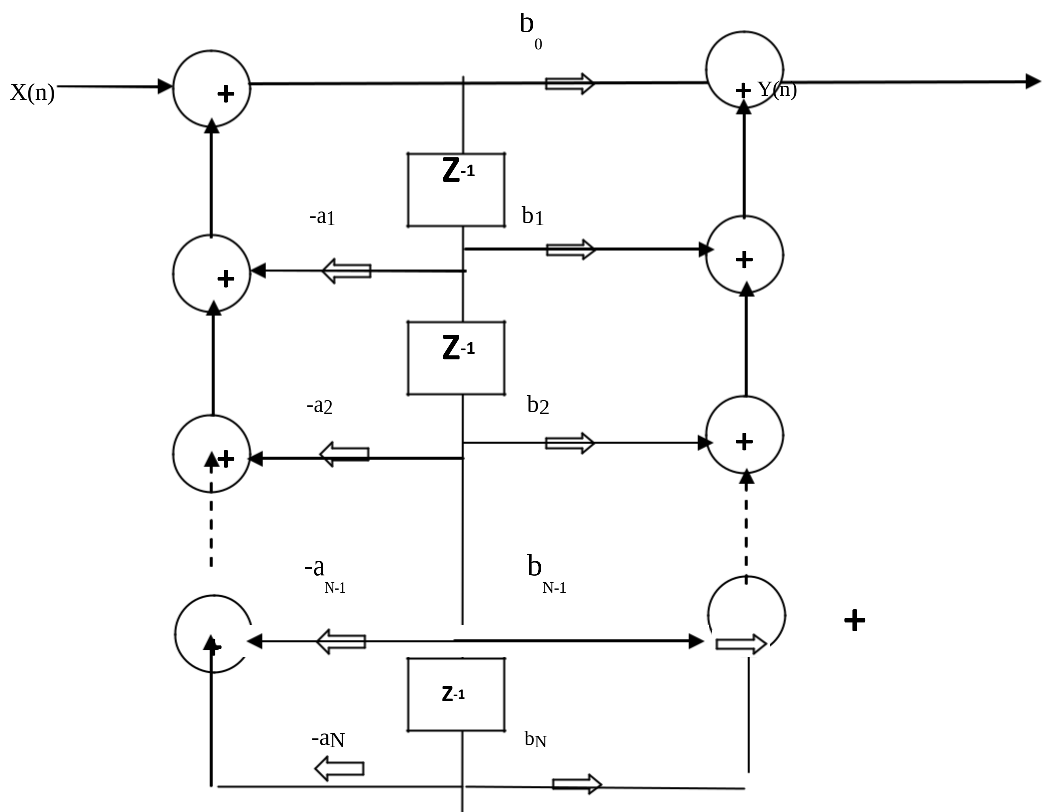


FIG - DIRECT FORM II REALIZATION OF IIR SYSTEM

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

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

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

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

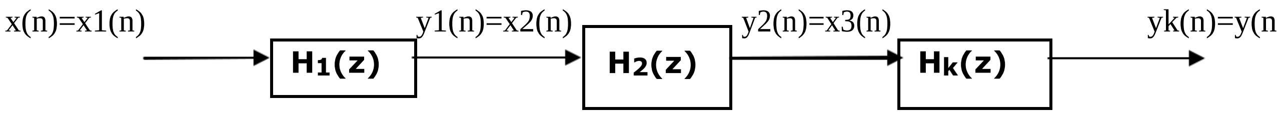


FIG - CASCADE FORM REALIZATION OF IIR SYSTEM

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

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

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

Thus Direct form of second order IIR system is shown as

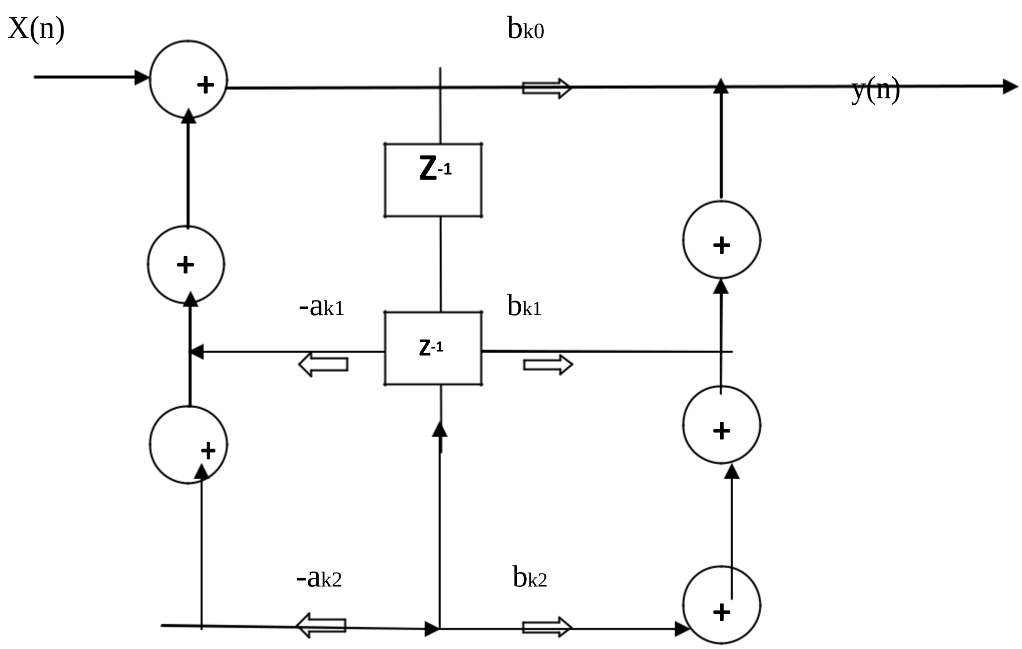


FIG - DIRECT FORM REALIZATION OF IIR SECOND ORDER SYSTEM (CASCADE)

PARALLEL FORM STRUCTURE FOR IIR SYSTEMS System function for IIR systems is given as

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

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

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

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

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

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

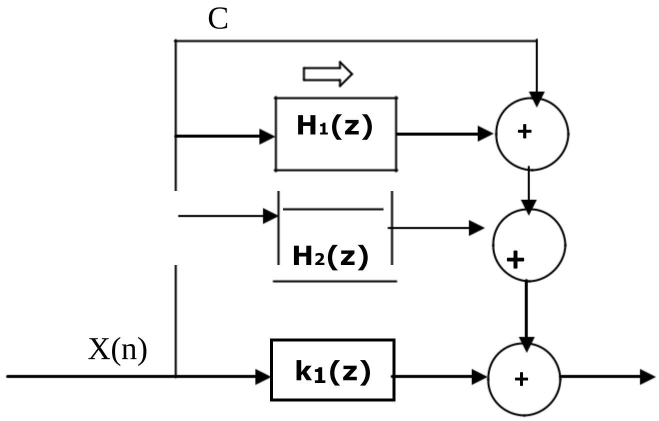


FIG - PARALLEL FORM REALIZATION OF IIR SYSTEM

Thus Direct form of second order IIR system is shown as

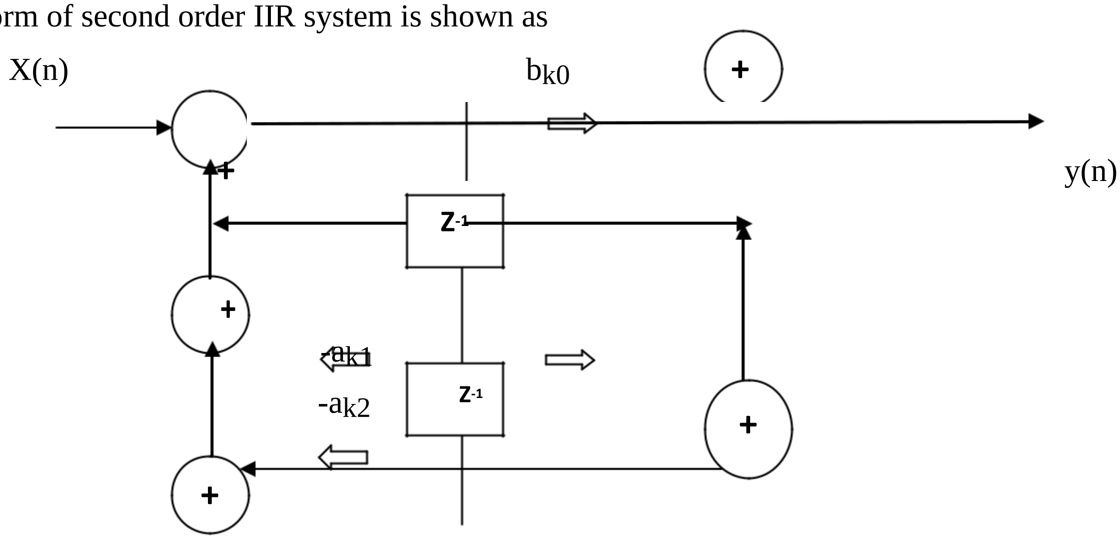


FIG - DIRECT FORM REALIZATION OF IIR SECOND ORDER SYSTEM (PARALLEL)

IIR FILTER DESIGN

IMPULSE INVARIANCE

BILINEAR TRANSFORMATION

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

$(2k-1)\prod/T \leq \Omega \leq (2k+1)\prod/T$ of $j\Omega$ axis are all mapped on the unit circle $\prod \leq \omega \leq \prod$. This takes place because of sampling.

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 $s= \sigma + j\,\Omega$.

$$Z= e^{ST} \qquad \text{(Relationship Between Z plane and S plane)}$$

$$Z= e^{(\sigma + j\,\Omega)\,T}$$

$$= e^{\sigma T} \cdot e^{j\,\Omega\,T}$$

Comparing Z value with the polar form we have.

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

Here we have three condition

If $\sigma = 0$ then $r=1$

If $\sigma < 0$ then $0 < r < 1$

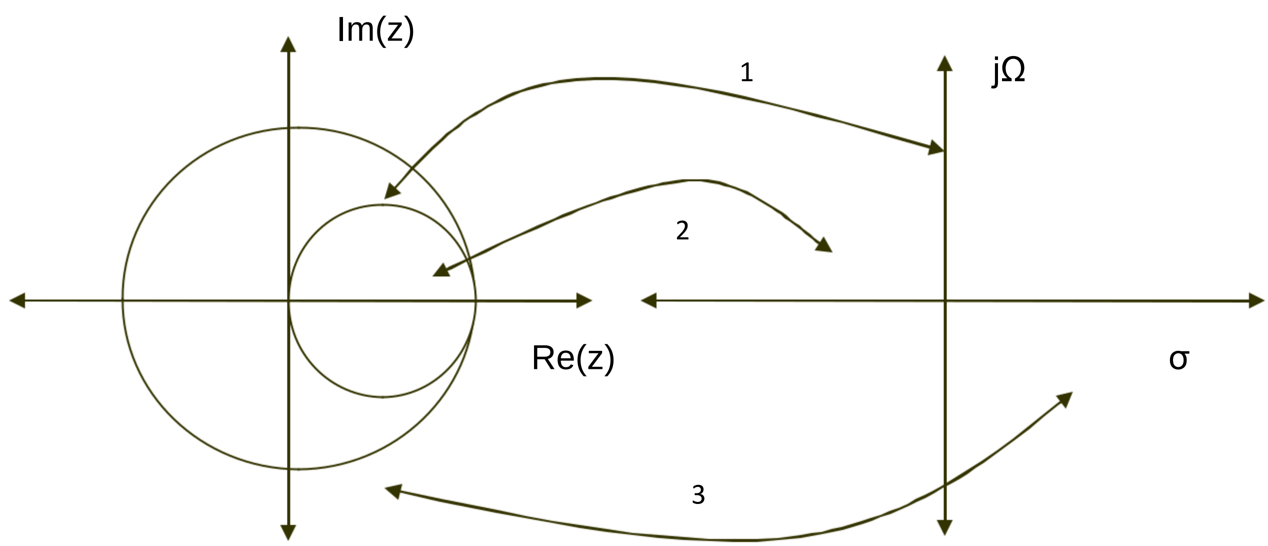
If $\sigma > 0$ then $r > 1$

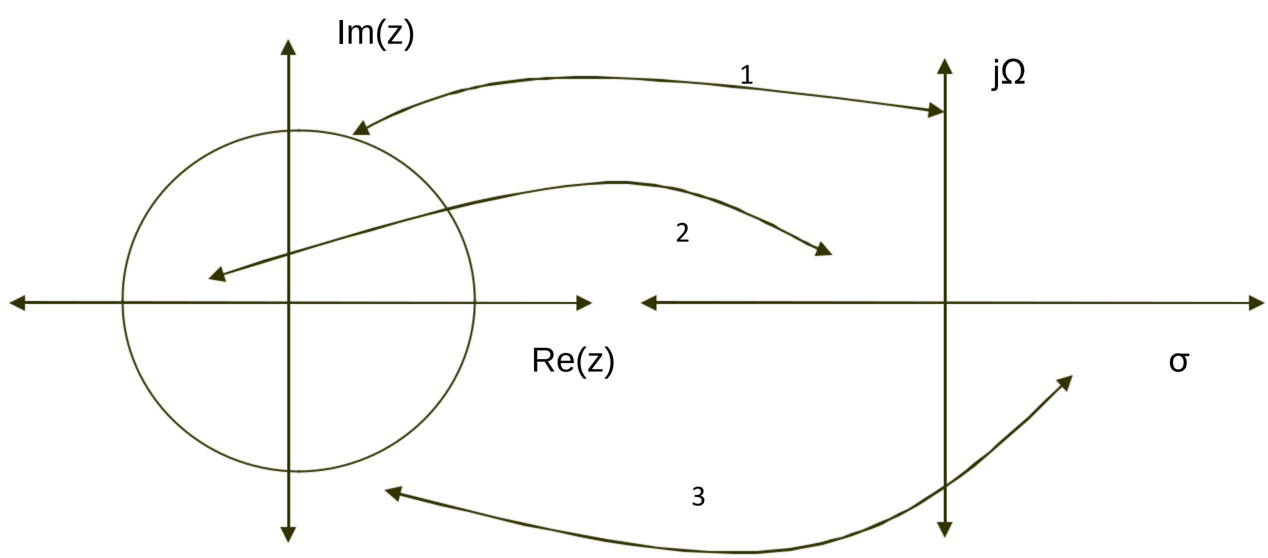
Thus

Left side of s-plane is mapped inside the unit circle.

Right side of s-plane is mapped outside the unit circle.

$j\Omega$ 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

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

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

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

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

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

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

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

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

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

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

STANDARD RELATIONS IN IIR DESIGN

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

EXAMPLES - IMPULSE INVARIANCE METHOD

S. 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

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.

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

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 $-\pi \leq \omega \leq \pi$.

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.

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

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

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

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

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

Comparing the above equation with $S = \sigma + j\Omega$. We have

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

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

Here we have three condition

If $\sigma < 0$ then $0 < r < 1$

If $\sigma > 0$ then $r > 1$

If $\sigma = 0$ then $r=1$ When $r=1$

$$= \frac{2}{T}\frac{\sin\omega}{1+\cos\omega}$$

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

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

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

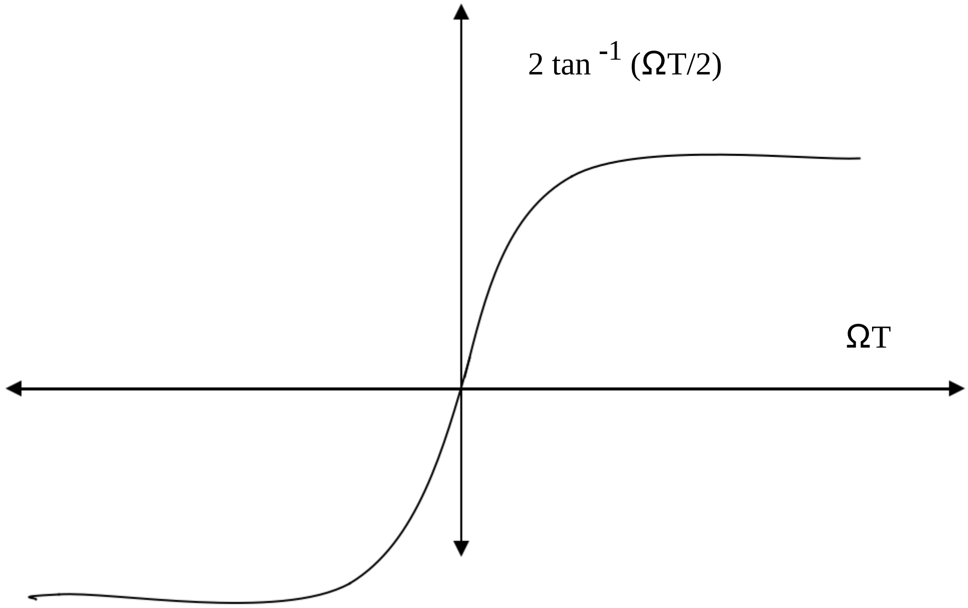


FIG - MAPPING BETWEEN FREQUENCY VARIABLE AND Ω IN BZT METHOD.

DIFFERENCE - IMPULSE INVARIANCE Vs BILINEAR TRANSFORMATION

S. 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	All poles and zeros are mapped.

	satisfy the same relationship.	

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^*.

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

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:

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

$$\omega c = 2\prod f$$

$$= 2000 \text{ rad/sec}$$

Step 2. To find the prewarp frequency

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

$$= \tan(\prod/10)$$

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)

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

Tutorial problems:

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

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.

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.

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

Notch Frequency= 60 Hz

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 Ω_c. This is called stopband of the filter. Ω_c is called cutoff frequency or critical frequency.

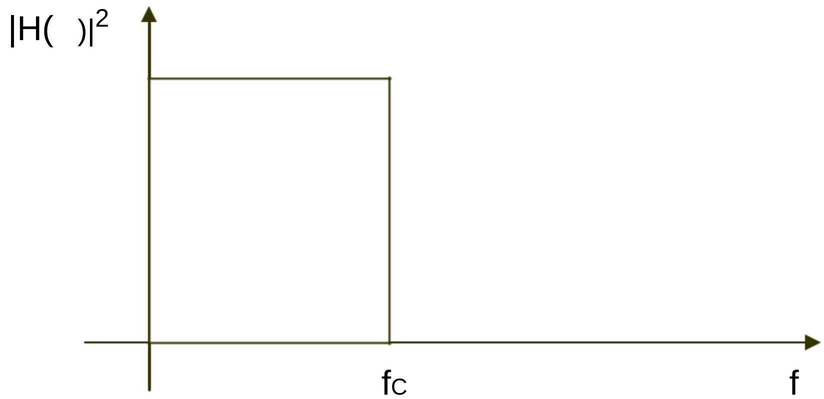
No Practical filters can provide the ideal characteristic. Hence approximation of the ideal characteristic are used. Such approximations are standard and used for filter design. Such three approximations are regularly used.

Butterworth Filter Approximation

Chebyshev Filter Approximation

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.

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

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

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

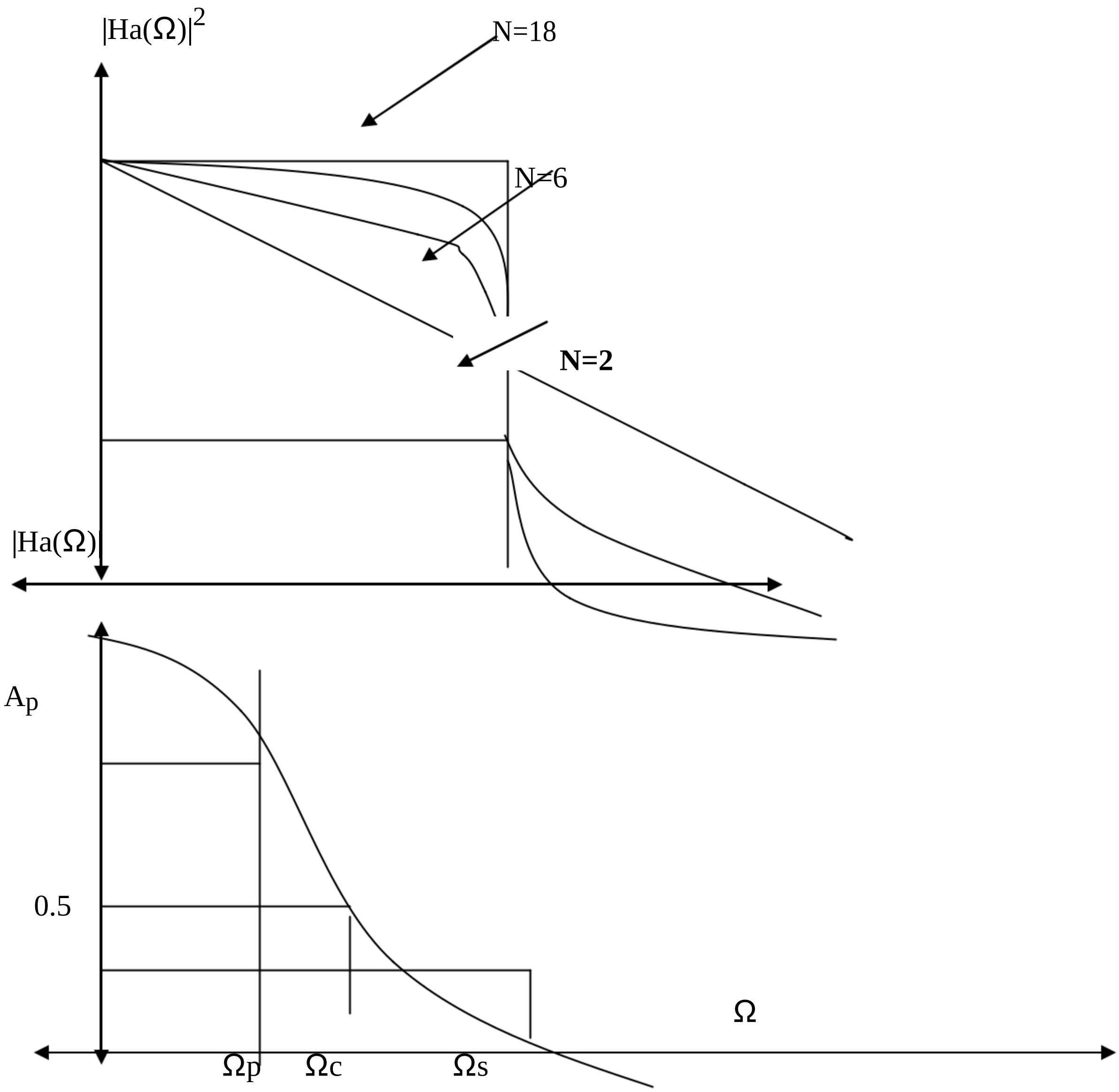
$$P_k = \pm \Omega_c\, e^{j\Pi/2}\, e^{j(2k+1)\Pi/2N}$$

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

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 ($N \rightarrow \infty$) 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

$|Ha(\Omega)| \geq Ap$ for $\Omega \leq \Omega p$ and $|Ha(\Omega)| \leq As$ for $\Omega \geq \Omega s$. 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

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

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

And cutoff frequency Ωc is calculated as

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

If As and Ap values are given in DB then

As (DB) = - 20 log As

log As = -As /20

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

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

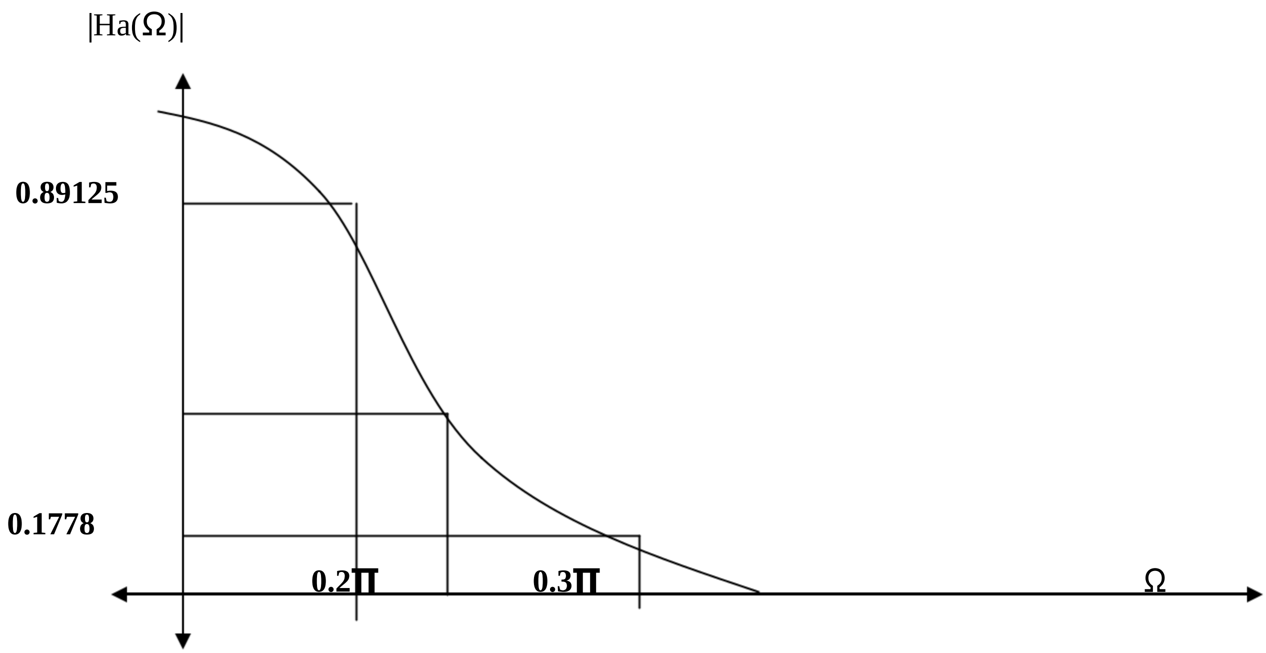
$$(As)^{-2} = 10^{0.1\,As} \text{ DB}$$

Hence equation (2) is modified as

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

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(Ω)| ≥ 0.89125 for Ω ≤ 0.2∏/T and |Ha(Ω)| ≤ 0.17783 for Ω ≥ 0.3∏/T.

Step 2) To determine the order of the filter.

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

N= 5.88

Order of the filter should be integer.

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 $\Omega c = 0.7032$

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

$S1 = -0.182 + j\,0.679$ $\quad S1^* = -0.182 - j\,0.679$

$S2 = -0.497 + j\,0.497$ $\quad S2^* = -0.497 - j\,0.497$

$S3 = -0.679 + j\,0.182$ $\quad S3^* = -0.679 - j\,0.182$

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

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

Hence

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

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

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

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)

$$H(z) = 1.97 \times \frac{0.5235\, z^{-1}}{1-1.297z^{-1}+0.695z^{-2}} \times\times \frac{0.29\, z^{-1}}{1-1.07z^{-1}+0.37z^{-2}} \times \frac{0.09\, z^{-1}}{1-0.99z^{-1}+0.26z^{-2}}$$

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

Tutorial problems:

Given for low pass butterworth filter

Ap= -1 db at 0.2∏

As= -15 db at 0.3∏

Calculate N and Pole location

Design digital filter using BZT method.

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.

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.

4.8 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)

S. 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	$s(\omega_h - \omega_l)$

		$s^2 + \omega_h \omega_l$ ω_h - higher band edge frequency ω_l - Lower band edge frequency

FREQUENCY TRANSFORMATION (DIGITAL FILTER)

S. No	Type of transformation	Transformation (Replace z^{-1} by)
1	Low Pass	$\dfrac{z^{-1} - a}{1 - az^{-1}}$
2	High Pass	$\dfrac{-(z^{-1} + a)}{1 + az^{-1}}$
3	Band Pass	$\dfrac{-(z^{-2} - a_1 z^{-1} + a_2)}{a_2 z^{-2} - a_1 z^{-1} + 1}$
4	Band Stop	$\dfrac{z^{-2} - a_1 z^{-1} + a_2}{a_2 z^{-2} - a_1 z^{-1} + 1}$

Example:

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

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

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)

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

Tutorial problems:

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.

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.

4.9 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

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

k=0

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

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

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

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

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

<u>FIR Filter Design Methods</u>

The various method used for FIR Filer design are as follows

Fourier Series method

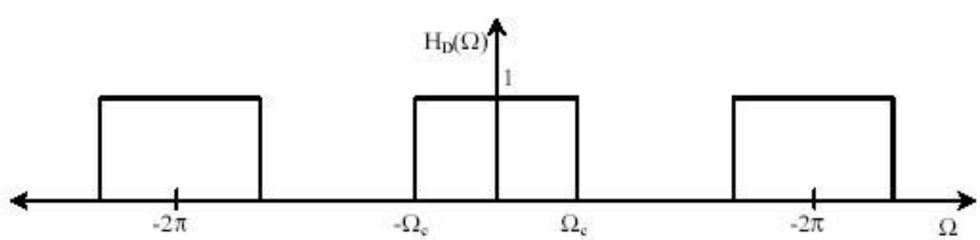
Windowing Method

DFT method

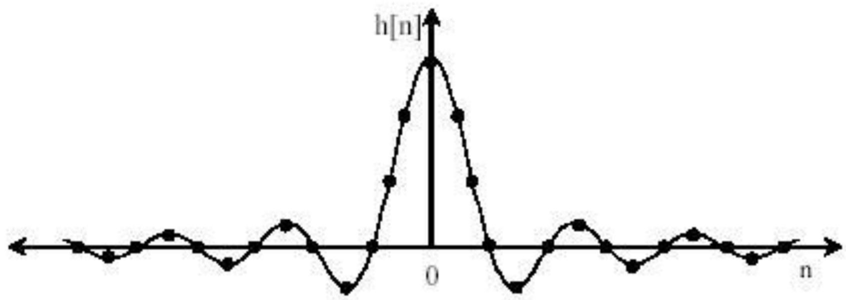
Frequency sampling Method. (IFT Method)

<u>GIBBS PHENOMENON</u>

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.

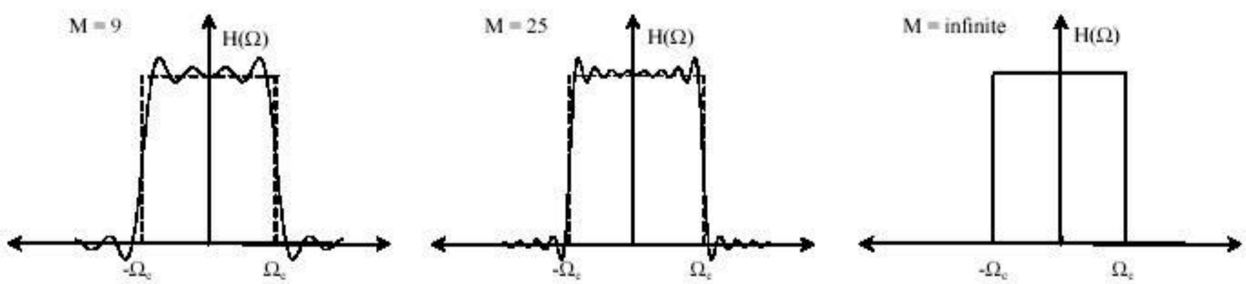


1.In Fourier series method, limits of summation index is -∞ to ∞. But filter must have finite terms. Hence limit of summation index change to -Q to Q where Q is some finite integer. But this type of truncation may result in poor convergence of the series. Abrupt truncation of infinite series is equivalent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

Consider the example of LPF having desired frequency response H_d (ω) 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(ω) of the window function. This oscillatory behavior is called "Gibbs Phenomenon".

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

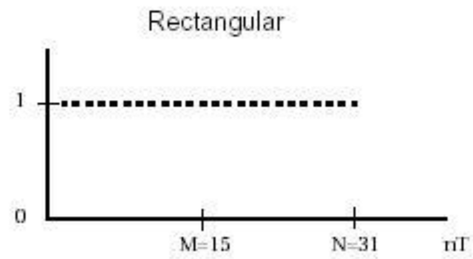


WINDOWING TECHNIQUE

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.

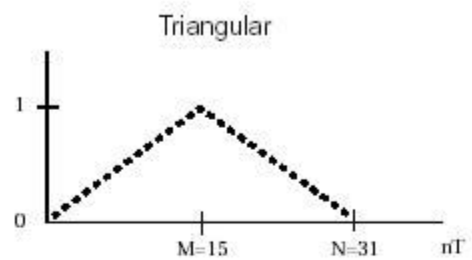
Rectangular Window: Rectangular This is the most basic of windowing methods. It does not require any operations because its values are either 1 or 0. It creates an abrupt discontinuity that results in sharp roll-offs but large ripples.



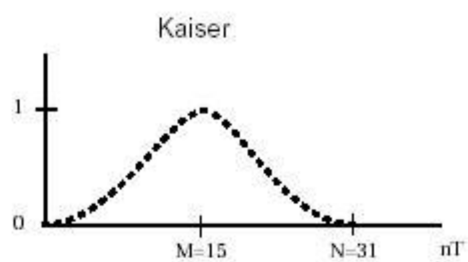
Rectangular window is defined by the following equation.

$$W[n] = \begin{cases} 1 & \text{for } 0 \le n \le N \\ 0 & \text{otherwise} \end{cases}$$

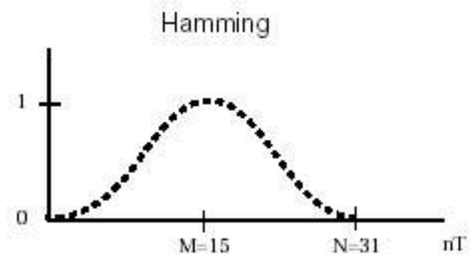
Triangular Window: The computational simplicity of this window, a simple convolution of two rectangle windows, and the lower sidelobes make it a viable alternative to the rectangular window.



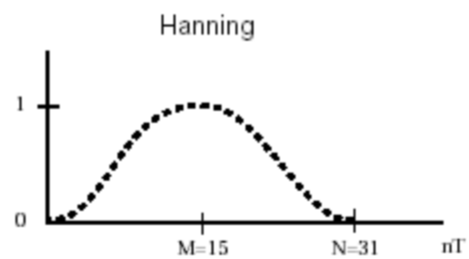
Kaiser Window: This windowing method is designed to generate a sharp central peak. It has reduced side lobes and transition band is also narrow. Thus commonly used in FIR filter design.



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.

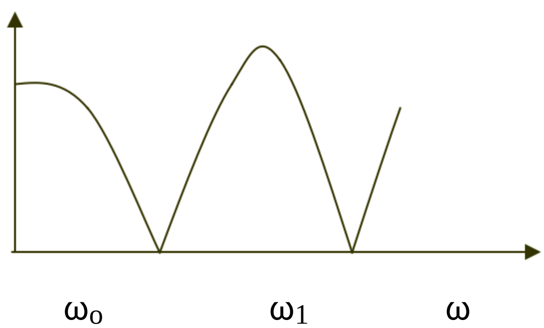
In the design of low pass filters, the poles should be placed near the unit circle at points corresponding to low frequencies (near ω=0)and zeros should be placed near or on unit circle at points corresponding to high frequencies (near $\omega=\pi$). The opposite is true for high pass filters.

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 $|H(\omega)|$ is as shown in figure.



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.

$H(\omega) = 1$ for $-\omega_c \leq \omega \leq \omega_c$

$= 0$ elsewhere

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

h(n)= ∞ −∞ ( ) (1)

$h(n)= {}^{1}$ (2)

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

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

Thus $h(n)= \sin \omega_C n / \prod n$ for $n\neq0$

Putting n=0 in equation (2) we have

$h(n) =$ 1 dω (3)

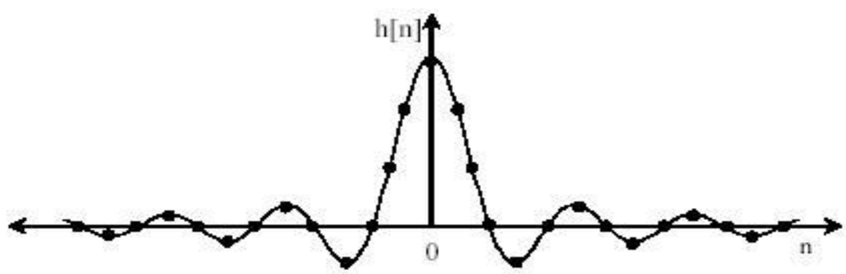
$$\frac{1}{2\prod} [\omega]_{-\omega c}^{\omega c}$$

and $h(n) = \omega_C / \prod$ for n=0

$$h(n) = \frac{\sin (\omega cn)}{\prod n} \quad \text{for } n\neq0$$

$$— \quad \text{for } n=0$$

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 $-\infty$ to ∞. Hence $h(n) \neq 0$ 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 inputx_{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 previousinputs.

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): $\mathbf{y_n = x_n}$

This is a zero order filter, since the current output y_n depends only on the current inputx_n and not on any previous inputs.

Example (2): $\mathbf{y_n = Kx_n}$

The order of this filter is again zero, since no previous outputs are required to give the current output value.

Example (3): $\mathbf{y_n = x_{n-1}}$

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): $\mathbf{y_n = x_n - x_{n-1}}$

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: $a_0 = 1$, $a_1 = -2$, $a_2 = 2$, $a_3 = 1$

CHAPTER 5

DIGITAL SIGNAL PROCESSOR

5.1 REQUIREMENTS OF DSP PROCESSORS

The most fundamental mathematical operation in DSP is sum of products also called as dot of products. Y(n)= h(0)*x(n) + h(1)*x(n-1) +………+ h(N-1)*x(n-N)

This operation is mostly used in digital filter designing, DFT, FFT and many other DSP applications. A DSP is optimized to perform repetitive mathematical operations such as the dot product. There are five basic requirements of s DSP processor to optimize the performance They are

Fast arithmetic

Extended precision

Fast Execution - Dual operand fetch

Fast data exchange

Fast operations

Circular buffering

S. No	Requirements	Features of DSP processor
1	Fast Arithmetic	Faster MACs means higher bandwidth. Able to support general purpose math functions, should have ALU and a programmable shifter function for bit manipulation. Powerful interrupt structure and timers
2	Extended precision	A requirement for extended precision in the accumulator register. High degree of overflow protection.
3	Fast Execution	Parallel Execution is required in place of sequential. Instructions are executed in single cycle of clock called as True instruction cycle as oppose to multiple clock cycle. Multiple operands are fetched simultaneously. Multi-processing Ability and queue, pipelining facility Address generation by DAG's and program sequencer.
4	Fast data Exchange	Multiple registers, Separate program and data memory and Multiple operands fetch capacity
5	Fast Operations	Program and data memories are on chip memory and extendable off-chip No multiplexing is used. Separate address, data and control bus. Powerful instruction set and various addressing modes.
6	Circular shift operations	Circular Buffers

5.2 MICROPROCESSOR ARCHITECTURES

There are mainly three types of microprocessor architectures present.

Von-Neumann architecture

Harvard architecture

Analog devices Modified Harvard architecture.

1) Von-Neumann Architecture

General purpose microprocessors uses the Von-Neumann Architectures. (named after the American mathematician John Von Neumann)

1.It consists of ALU, accumulator, IO devices and common address and data bus. It also consists of a single memory which contains data and instructions, a single bus for transferring data and instructions into and out of the CPU.

2.Multiplying two numbers requires at leased three cycles, two cycles are required to transfer the two numbers into the CPU and one cycle to transfer the instruction.

3.This architecture is giving good performance when all the required tasks can be executed serially.

4.For large processing applications like DSP applications Von-Neumann architecture is not suitable as processing speed is less. Processing speed can be increased by pipelining up to certain extend which is not sufficient for DSP applications. In order to perform a single FIR filter multiply-accumulate, an instruction is fetched from the program memory, and during the same cycle, a coefficient can be fetched from the data memory. A second cycle is required to fetch the data word from data memory

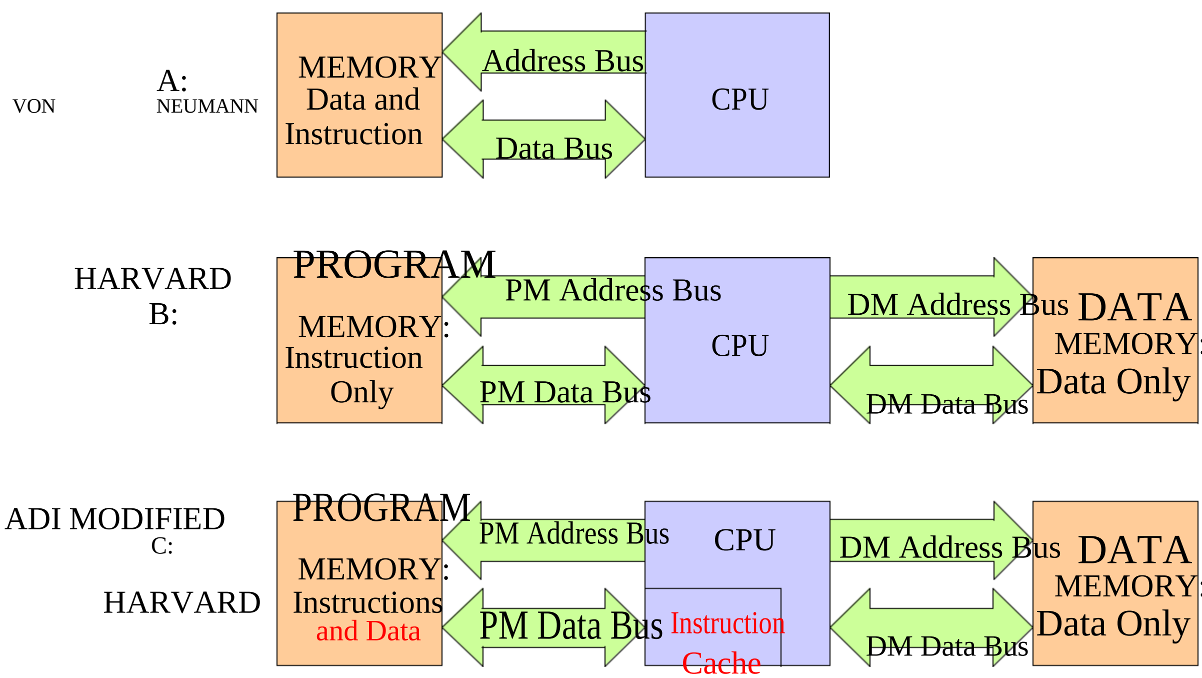


Figure: MICROPROCESSOR ARCHITECTURES

2) Harvard Architecture (named for the work done at Harvard university)

Data and program instructions each have separate memories and buses as shown. Program memory address and data buses for program memory and data memory address and data buses for data memory.

Since the buses operate independently, program instructions and data can be fetched at the same time. Therefore improving speed over the single bus Von Neumann design.

In order to perform a single FIR filter multiply-accumulate, an instruction is fetched from the program memory, and during the same cycle, a coefficient can be fetched from the data memory. A second cycle is required to fetch the data word from data memory.

5.3 CORE ARCHITECTURE OF ADSP-21xx

ADSP-21xx family DSP's are used in high speed numeric processing applications.

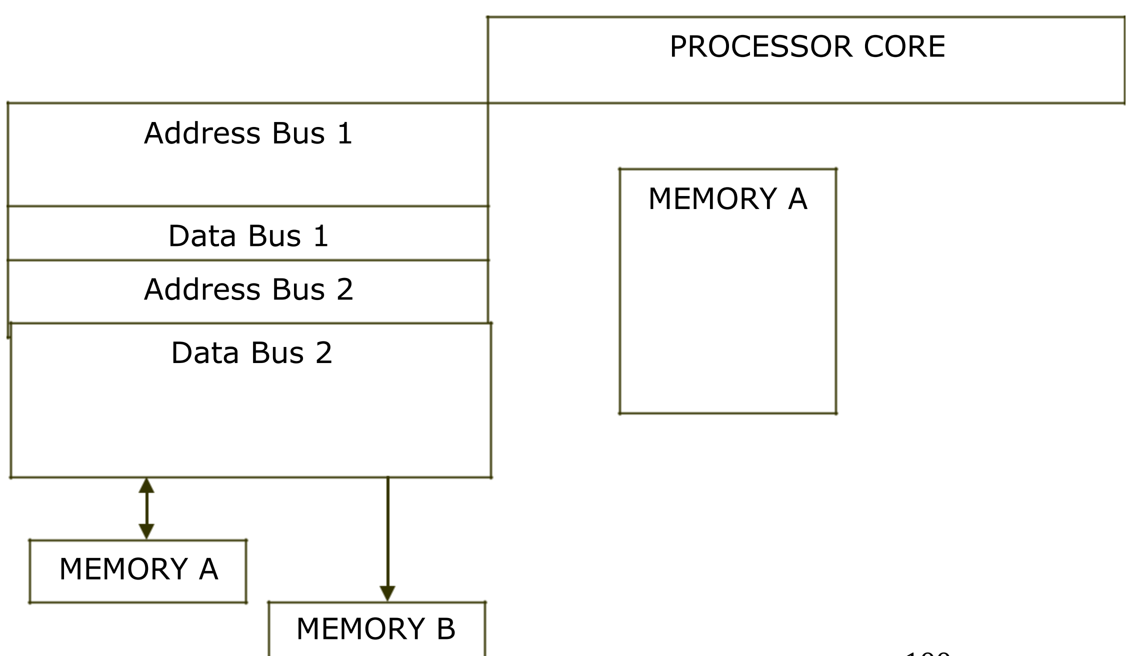
ADSP-21xx architecture consists of Five Internal Buses

Program Memory Address(PMA)

Data memory address (DMA)

Program memory data(PMD)

Data memory data (DMD) and Result (R)



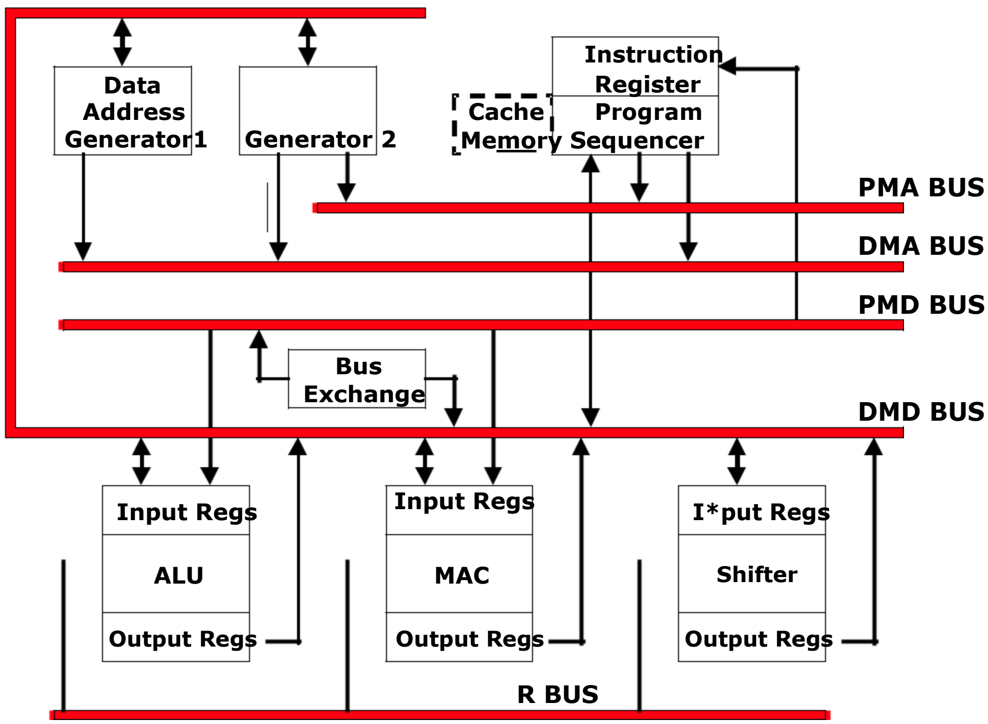
Three Computational Units are

Arithmetic logic unit (ALU)

Multiply-accumulate (MAC)

Shifter

Two Data Address generators (DAG)
Program sequencer
On chip peripheral Options
Data Memory
Timer
Serial Port



BUSES: **Block Diagram of Architecture**

The ADSP-21xx processors have five internal buses to ensure data transfer.

1.PMA and DMA buses are used internally for addresses associated with Program and data memory. The PMD and DMD are used for data associated with memory spaces. Off Chip, the buses are multiplexed into a single external address bus and a single external data bus. The address spaces are selected by the appropriate control signal.

2.The result (R) bus transfers the intermediate results directly between various computational units.

3.PMA bus is 14-bits wide allowing direct access of up to 16k words of code and data. PMD bus is 24 bits wide to accommodate the 24 bit instruction width.

The DMA bus 14 bits wide allowing direct access of up to 16k words of data.The DMD bus is 16 bit wide.

4.The DMD bus provides a path for the contents of any register in the processor to be transferred to any other register or to any external data memory location in a single cycle. DMA address comes from two sources. An absolute value specified in the instruction code (direct addressing) or the output of DAG (Indirect addressing). The PMD bus can also be used to transfer data to and from the computational units thro direct path or via PMD-DMD bus exchange unit.

COMPUTATIONAL UNITS:

The processor contains three -independent computational units. ALU, MAC (Multiplier-accumulator) and the barrel shifter. The computational units process 16-bit data directly. ALU is 16 bits wide with two 16 bit input ports and one output port. The ALU provides a standard set of arithmetic and logic functions. ALU Features

Add, subtract, Negate, increment, decrement, Absolute value, AND, OR, EX-OR, Not etc.

Bitwise operators, Constant operators

Multi-precision Math Capability

Divide Primitives and overflow support.

MAC:

The MAC performs high speed single-cycle multiply/add and multiply/subtract operations. MAC has two 16 bit input ports and one 32 bit product output port. 32 bit product is passed to a 40 bit adder/subtractor which adds or subtracts the new product from the content of the multiplier result (MR). It also contains a 40 bit accumulator which provides 8 bit overflow in successive additions to ensure that no loss of data occurs. 256 overflows would have to occur before any data is lost. A set of background registers is also available in the MAC for interrupts service routine.

SHIFTER:

The shifter performs a complete set of shifting functions like logical and arithmetic shifts (circular or linear shift) , normalization (fixed point to floating point conversion), demoralization (floating point to fixed point conversion) etc

ALU, MAC and shifter are connected to DMD bus on one side and to R bus on other side. All three sections contains input and output registers which are accessible from the internal DMD bus. Computational operations generally take the operands from input registers and load the result into an output register.

DATA ADDRESS GENERATORS (DAG):

Two DAG's and a powerful program sequencer ensure efficient use of these computational units. The two DAG's provides memory addresses when memory data is transferred to or from the input or output registers. Each DAG keeps track of up to four address pointers. Hence jumps, branching types of instructions are implemented within one cycle. With two independent DAG's, the processor can generate two address simultaneously for dual operand fetches.

DAG1 can supply addresses to data memory only. DAG2 can supply addresses to either data memory or program memory. When the appropriate mode bit is set in mode status register (MSTAT), the output address of DAG1 is bit-reversed before being driven onto the address bus. This feature facilitates addressing in radix-2 FFT algorithm.

PROGRAM SEQUENCER:

The program sequencer exchanges data with DMD bus. It can also take from PMD bus. It supplies instruction address to program memory. The sequencer is driven by the instruction register which holds the currently executing instruction. The instruction register introduces a single level of pipelining into the program flow. Instructions are fetched and loaded into the instruction register during one processor cycle, and executed during the following cycle while the next instruction is pre-fetched. The cache memory stores up to 16 previously executed instructions. Thus data memory on PMD bus is more efficient because of cache memory. This also makes pipelining and increase the speed of operations.

FEATURES OF ADSP-21xx PROCESSOR

16 bit fixed DSP microprocessor

Enhanced Harvard architecture for three bus performance.

Separate on chip buses for program and data memory.

25 MIPS, 40 ns maximum instruction set 25Mhz frequency.

Single cycle instruction execution i.e True instruction cycle.

Independent computational units ALU, MAC and shifter.

On chip program and data memories which can be extended off chip.

Dual purpose program memory for instruction and data.

Single cycle direct access to 16K × 16 of data memory.

Single cycle direct access to 16K × 24 of program m

5.4 ADSP-21xx DEVELOPMENT TOOLS

Various development tools such as assembler , linker, debugger, simulator are available for ADSP-21xx family.

SYSTEM BUILDER

The system builder is the software development tool for describing the configuration of the target system's memory and I/O. The ranges for program memory(PM) and data memory(DM) are described.

The program memory space is allotted with 16 K of memory for instructions and 16K for data. The data memory space is also divide into two blocks. The lowest 2K is the dual -port memory that is shared by both processors. Each processor has an additional 14K of private data memory.

.SYSTEM

.ADSP2108

{PROGRAM MEMORY SECTION}

.SEG/ROM/ABS=0/PM/CODE .SEG/RAM/ABS=0X4000/PM/DATA CODE_AREA

DATA_AREA

{DATA MEMORY SECTION

.SEG/RAM/ABS=0/DM/DATA COMMON

MEMORY

.SEG/RAM/ABS=0X800/DM/DATA PRIVATE MEMORY

.PORT/RAM/ABS=0x100 DATA MEMORY IO PORT

.PORT/RAM/ABS=0x200 AD PORT

.PORT/RAM/ABS=0x300 DA PORT

.ENDSYS

The sample system configuration file (.SYS) is shown. .SEG directive is used to declare the system's physical memory segments and its characteristics. .PORT directive declares the memory mapped IO port. .ENDSYS indicates the end of .SYS file.

ASSEMBLER

The assembler translated source code into object code modules. The source code is written in assembly language file (.DSP) Assembler reads .DSP file and generates four

output filed with the same root name. Object file(.OBJ), Code File(.CDE), Initialization File (.INT), List File(.LST) etc. The file can be assembled by the following command.

ASM21FILTER

S.No	File Extension	Application
1	.OBJ	Binary codes for instruction, Memory allocation and symbol declaration, Addresses of the instructions
2	.CDE	Instruction opcodes
3	.INT	Initialization information
4	.LST	Documentation

LINKER

The linker is a program used to join together object files into one large object file. The linker produces a link file which contains the binary codes for all the combined modules. The linker uses .INT, .CDE, .OBJ and .ACH file and generates three files. Map listing file (.MAP), Memory Image File (.EXE) and Symbol table (.SYM).

LD21 FILTER -a SAMPLE -e FILTER

Here Filter is the input file name, sample.ACH file is system architecture file name and FILTER.EXE is output file name.

DEBUGGER

A debugger is a program which allows user to load object code program into system memory, execute the program and debug it. The debugger allows the user to look at the contents of registers and memory locations etc. We can change the contents of registers and memory locations at run time, generates breakpoints etc.

SIMULATOR

The multiprogramming environment can be tested using the simulator. When simulated, the filter program produces output data and stores in common data memory. Simulator command is used to dump form data memory to store the dual port data memory image on disk which can be reloaded and tested.

5.5 APPLICATIONS OF DSP

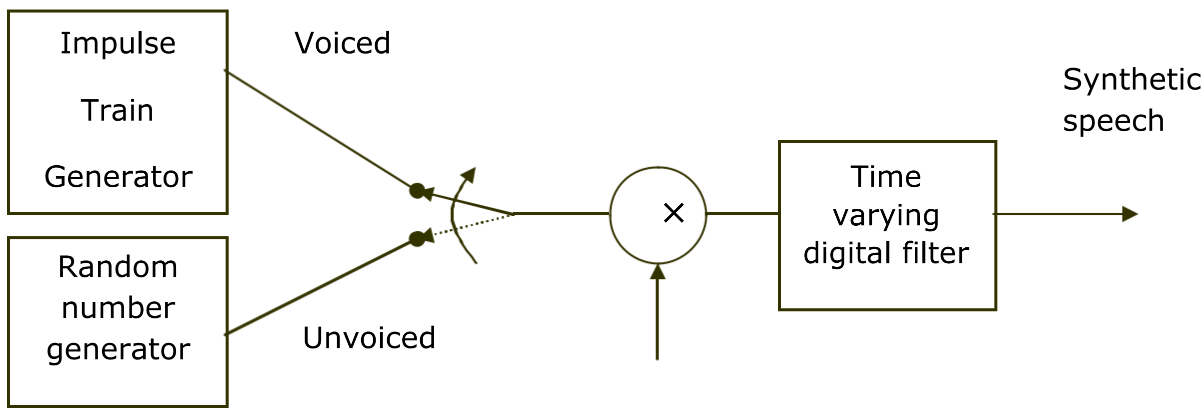
1. SPEECH RECOGNITION

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

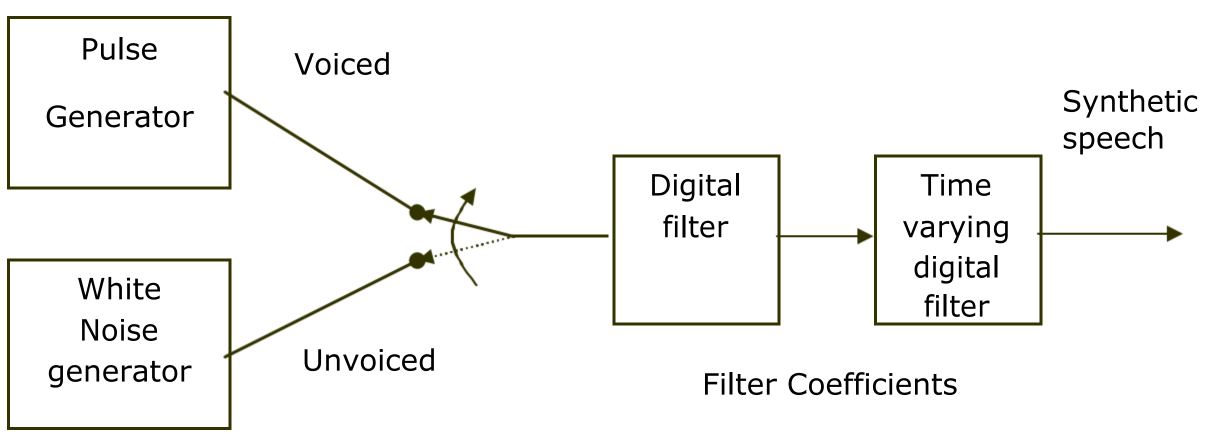
Fig shows block diagram of speech synthesizer using linear prediction

For voiced sound, pulse generator is selected as signal source while for unvoiced sounds noise generator is selected as signal source.

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.

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.

Pitch Period



3. SOUND PROCESSING

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.

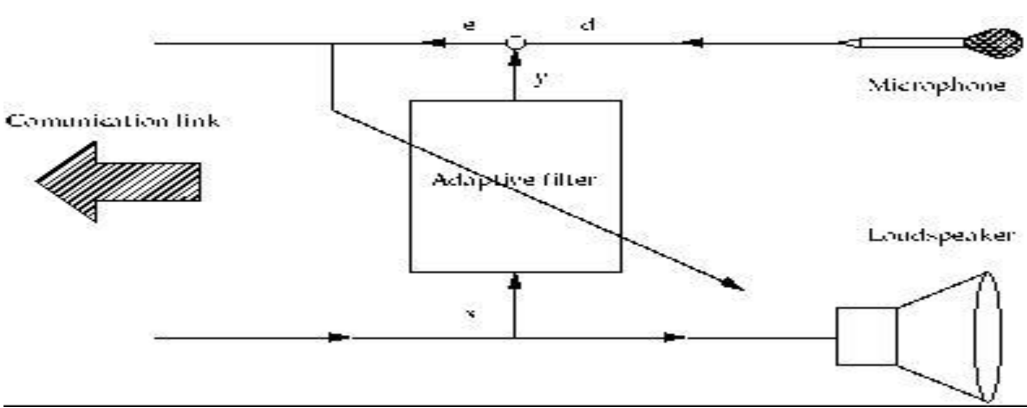


Figure : Echo canceller principle

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 e. 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.

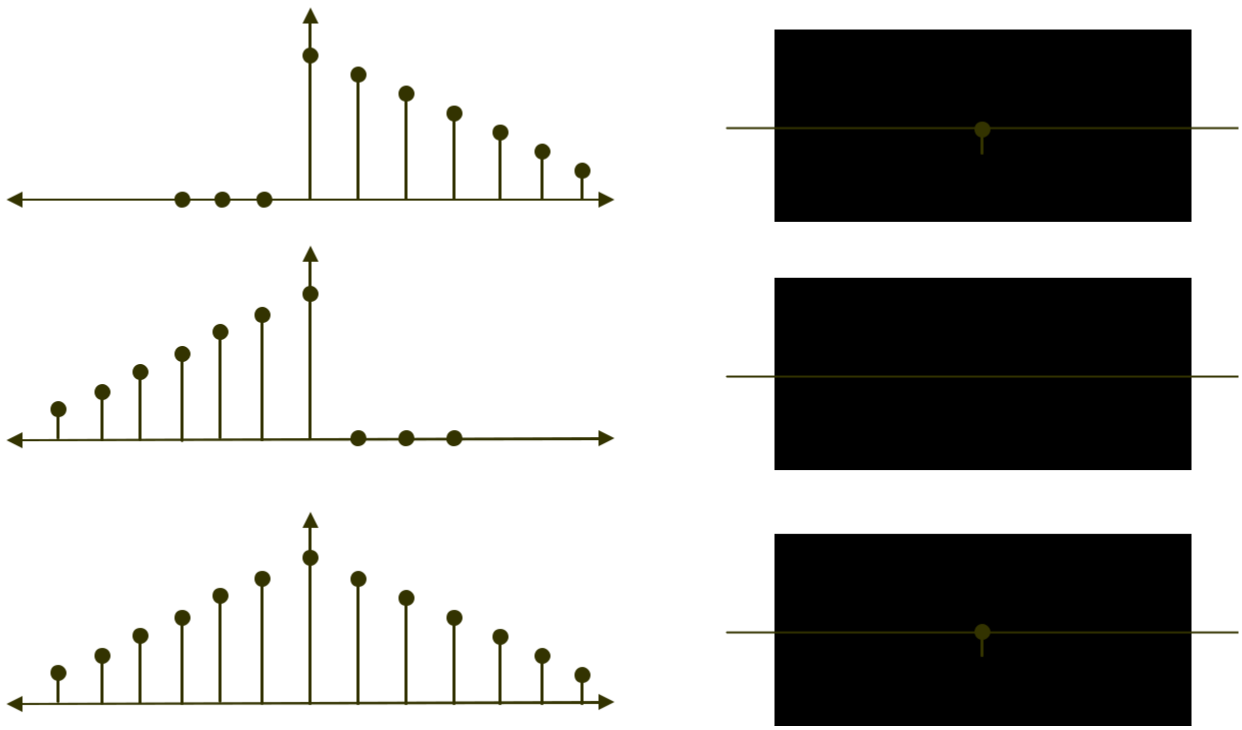
5. VIBRATION ANALYSIS

Normally machines such as motor, ball bearing etc systems vibrate depending upon the speed of their movements.

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.

This spectrum analysis can be performed by DSP system. The DSP system can also be used to monitor other parameters of the machine simultaneously.

5.6 FIGURE FOR FINITE DURATION



There are one sided(right and left sequences) and two sided sequences are present in the Z Transform pair.

It cosists of two duration they are

Finite duration

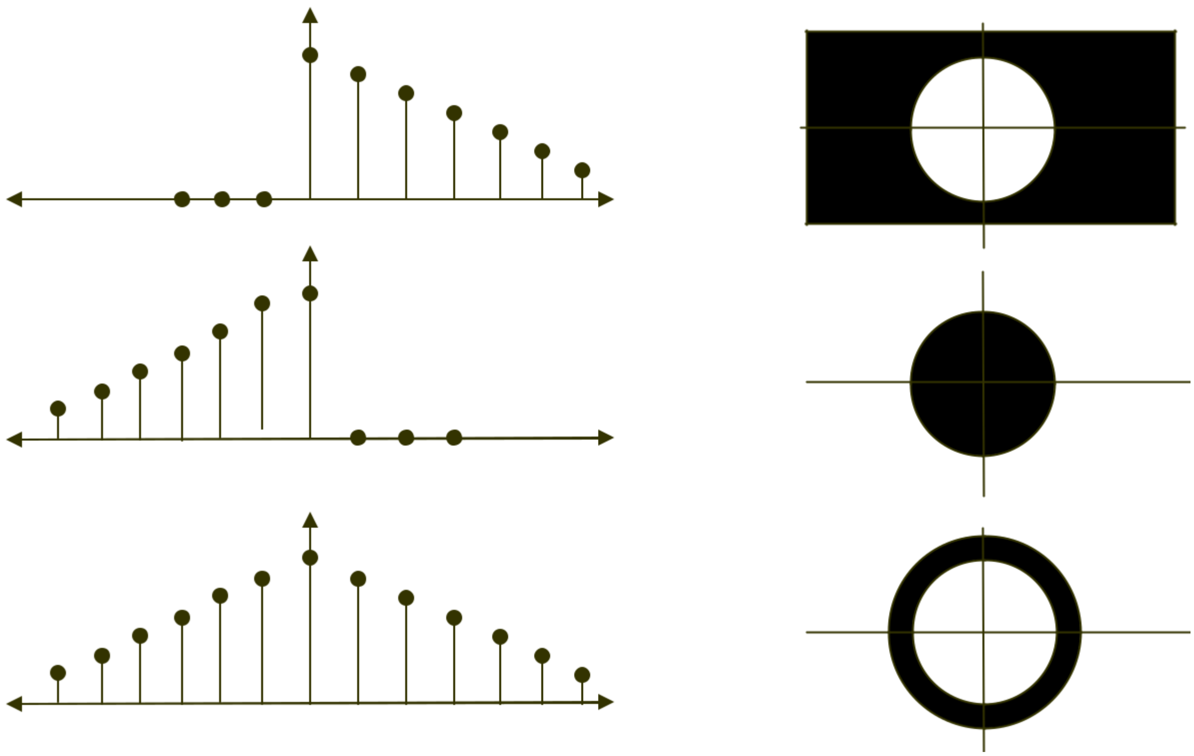
Infinite duration

IF ITS RIGHT SIDED SEQUENCE ----- ROC LIES BETWEEN Z = ∞

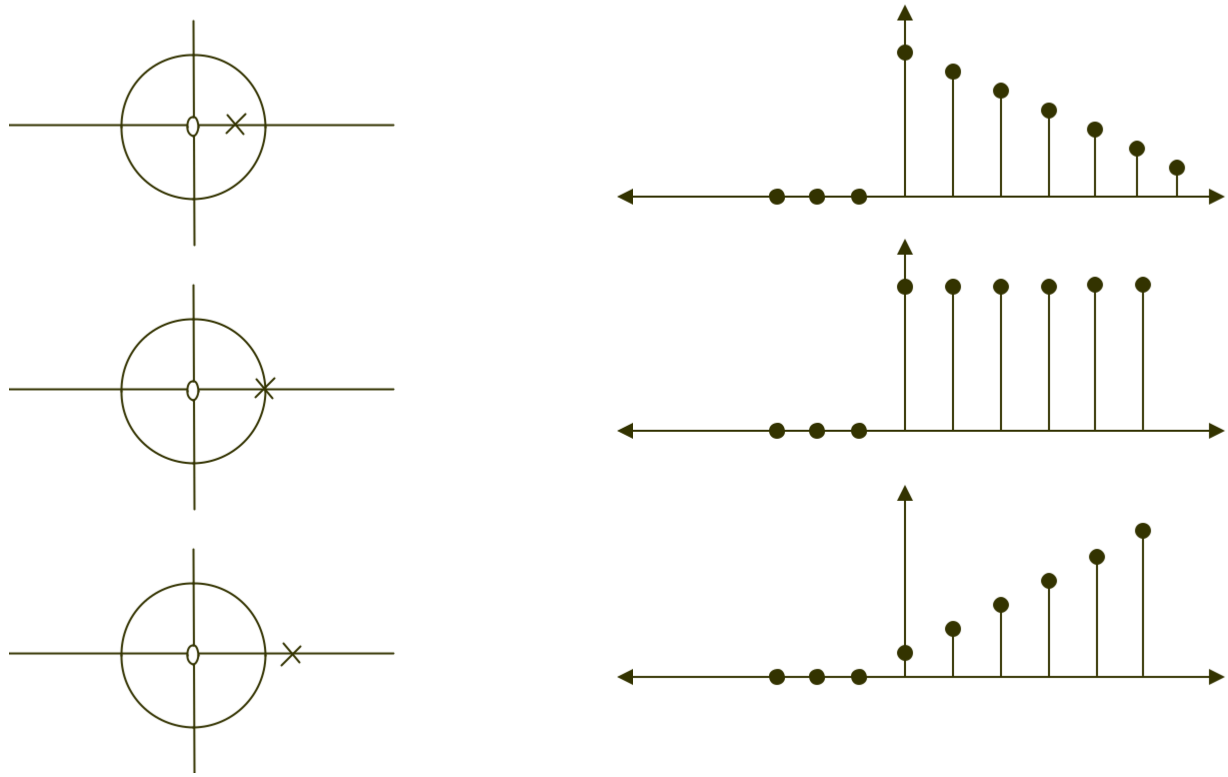
IF ITS LEFT SIDED SEQUENCES ----- ROC LIES BETWEEN Z = 0

IF ITS BOTH SIDED SEQUENCES ----- ROC LIES BETWEEN Z= 0 & Z=∞.

5.7 FIGURE FOR INFINITE DURATION



5.8 FIGURE FOR POLE ZERO PLOT



Match the Pairs.

REFERENCES:

Poorna Chandra S, Sasikala. B ,Digital Signal Processing, Vijay Nicole/TMH,2013.

B.P.Lathi, ‗Principles of Signal Processing and Linear Systems‘, Oxford

Taan S. ElAli, ‗Discrete Systems and Digital Signal Processing with Mat Lab‘,

4. Sen M.kuo, woonseng…s.gan, ―Digital Signal Processors, Architecture, Implementations &

5. Dimitris G.Manolakis, Vinay K. Ingle, applied Digital Signal Processing,Cambridge,2012

Lonnie C.Ludeman ,‖Fundamentals of Digital Signal Processing‖,Wiley,2013

R.L Lekha , Digital Signal Processing by Lakshmi publications.

GLOSSARY TERMS:

DDC:
Digital Down ConverterTuner Mix a signal down to baseband. A DDC operates in the complex domain and consists of a frequency mixer, a LPF, and a decimator. SeethedownmixerinInputDevicewindow.

Decimation:
A decimator reduces the sample rate by discarding samples. It is important to low passfilterinordertopreventaliasing.

DSP:
Digital Signal Processing. The art of manipulating signals in the discrete digital domain.

FFT:

Fast Fourier Transform. A mathematical operation that converts between the time andthefrequencydomains. O(n*logn).

FIR:

Finite Impulse Response filter. Performs convolution in the time domain. A DSP algorithmic construct made up of cascaded MAC units and filter taps. FIR filters are stable and have linear phase. Both the decimator in Input Devices and Hilbert filtersareimplementedwithFIRfilters.

Address: The logical location of program code or data stored in memory.

Addressing mode: The method by which an instruction interprets its operands to acquire the data it needs.

Address stage: The second stage of the parallel processor‘s fetch, address, execute (FAE) pipeline during which addresses are calculated and supplied to the crossbar. (TMS320C8x)

Address unit: Hardware on the parallel processor that computes a bit address during each cycle. Each parallel processor has two address units: a global address unit and a local address unit. (TMS320C8x)

Address unit arithmetic: The parallel processor‘s use of the local and global address units to perform general-purpose arithmetics in parallel with the data unit. The computed address is not used for memory access, but is stored in the destination register. (TMS320C8x)

Address visibility (AVIS) bit: A bit field that allows the internal program address to appear at the external address pins. This enables the internal program address to be traced and the interrupt vector to be decoded in conjunction with the interrupt acknowledge (IACK) signal when the interrupt vectors reside in on-chip memory. At reset, AVIS = 0. (TMS320C5x, TMS320C54x, TMS320C2xx)

Administrative privileges: Authority to set software and hardware access; includes access and privileges to install, manage, and maintain system and application software and directories on a network server or individual computer systems.

ADTR: Asynchronous data transmit and receive register. See also receive (ADTR) register.

AFB: See auxiliary register file bus.

Aggregate type: A C data type, such as a structure or an array, in which a variable is composed of multiple other variables, called members. **AIC:** See analog interface circuit.

A-Law companding: See companded.

Alias disambiguation: A technique that determines when two pointer expressions cannot point to the same location, allowing the compiler to freely optimize such expressions. (TMS320C6200)

Aliasing: 1) A method of customizing debugger commands; aliasing provides a shorthand method for entering often-used command strings. 2) A method of accessing a single data object in more than one way, as when a pointer points to a named object. The optimizer has logic to detect aliasing, but aliasing can cause problems for the optimizer. 3) Aliasing occurs when a single object can be accessed in more than one way, such as when two pointers point to a single object. It can disrupt optimization, because any indirect reference could refer to any other object.

Alignment: A process in which the linker places an output section at an address that falls on an n-byte boundary, where n is a power of 2. You can specify alignment with the SECTIONS linker directive.

Allocation: A process in which the linker calculates the final memory addresses of output sections.

Allocation node: The processor node into which an internode message is allocated.

ALU: See arithmetic logic unit.

ALU function: For the parallel processor, an action performed on the three inputs to the arithmetic logic unit (ALU), which includes any arithmetic or Boolean combination of the three inputs, as well as mixed arithmetic and Boolean functions. (TMS320C8x)

ALU function modifier: For the parallel processor, a 4-bit code that specifies modifications to the functions performed by the arithmetic logic unit (ALU) data path (such as carry-in or multiple arithmetic). These function modifiers are specified in the opcode or in the D0 register, depending on the application. (TMS320C8x)

ALU operation: For the parallel processor, an action performed by the arithmetic logic unit (ALU) data path (that is, the result of the ALU function, the operation class, and any function modifiers). (TMS320C8x) analog interface circuit (AIC): Integrated circuit that performs serial analog- to-digital (A/D) and digital-to-analog (D/A) conversions.

Analog mixing: The mixing together of two analog signals; the multiplexing of two analog signals into one.

Analog repeater: Analog Repeater is telecommunications equipment used to boost and clarify analog signals on telephone lines.

Analog-to-digital (A/D): Conversion of continuously variable electrical signals to discrete or discontinuous electrical signals.

Analog-to-digital (A/D) converter: A converter that changes real-world analog signals such as light and sound into digital code.

ANSI: American National Standards Institute. An organization that establishes standards voluntarily followed by industries.

ANSI C: A version of the C programming language that conforms to the C standards defined by the American National Standards Institute (ANSI). annul: Any instruction that is annulled does not complete its pipeline stages.

API: See application programming interface.

application programming interface (API): Used for proprietary application programs to interact with communications software or to conform to protocols from another vendor‘s product. (TMS320C8x)

Application-Specific ICs (ASICs): Application-Specific ICs (ASICs) are chips designed by our customers for specific applications by integrating cells from a TI standard library of pre-tested code. ASIC design is faster than designing a chip from scratch, and design changes can be made more easily.

AR0–AR7: Auxiliary Registers 0–7. Eight registers that are used as pointers to an address within the data space address range. The registers are operated on by the auxiliary register arithmetic unit (ARAU) and are selected by the auxiliary register pointer (ARP).

AR: See auxiliary register.

ARAU: See auxiliary register arithmetic unit.

ARB: See auxiliary register pointer buffer.

Architecture: The software or hardware structure of all or part of a computer system; includes all the detailed components of the system.

Archive library: A collection of individual files grouped into a single file by the archiver.

Archiver: A software program that collects several individual files into a single file called an archive library. With the archiver, you can add, delete, extract, or replace members of the archive library.

ARCR: See auxiliary register compare register.

Argument buffer: A memory block into which argument values are placed that accompany a command to a server parallel processor.

Arithmetic logic unit (ALU): The section of the computer that carries out all arithmetic operations (addition, subtraction, multiplication, division, or comparison) and logic functions.

ARP: See auxiliary register pointer.

ARR: See BSP address receive register.

ARSR: See asynchronous serial port receive shift register.

ASCII: American Standard Code for Information Interchange, 1968. The standard set of 7-bit coded characters (8-bit including parity check) used for information interchange among data processing systems, communications systems, and associated equipment. The ASCII set consists of control characters and graphics characters.

ASPCR: See asynchronous serial port control register.

Assemble: To prepare a machine-language program from a symbolic language program by substituting absolute operation codes for symbolic operation codes and absolute or relocatable addresses for symbolic addresses.

Assembler: A software program that creates a machine-language program from a source file that contains assembly language instructions and directives. The assembler substitutes absolute operation codes for symbolic operation codes, and absolute or relocatable addresses for symbolic addresses.

Assembly language: A low-level symbolic programming language, closely resembling machine code language and composed of groups of letters — each group representing a single instruction; allows a computer user to write a program using mnemonics instead of numeric instructions.

Assembly language instructions: The language in which computer operations are represented by mnemonics.

SHORT QUESTIONS AND ANSWERS:

UNIT-I - SIGNALS & SYSTEMS

1. Define Signal.

A Signal is defined as any physical quantity that varies with time, space or any other independent variables.

2. Define a system.

A System is a physical device (i.e., hardware) or algorithm (i.e., software) that performs an operation on the signal.

3. What are the steps involved in digital signal processing?

Converting the analog signal to digital signal, this is performed by A/D converter Processing Digital signal by digital system.

Converting the digital signal to analog signal, this is performed by D/A converter.

Give some applications of DSP?

Speech processing – Speech compression & decompression for voice storage system

Communication – Elimination of noise by filtering and echo cancellation.

Bio-Medical – Spectrum analysis of ECG,EEG etc.

Write the classifications of DT Signals.

Energy & Power signals

Periodic & Aperiodic signals

Even & Odd signals.

6. What is an Energy and Power signal?

Energy signal: A finite energy signal is periodic sequence, which has a finite energy but zero average power.**Power signal:** An Infinite energy signal with finite average power is called a power signal.

7. What is Discrete Time Systems?

The function of discrete time systems is to process a given input sequence to generate output sequence. In practical discrete time systems, all signals are digital signals, and operations on such signals also lead to digital signals. Such discrete time systems are called digital filter.

Write the Various classifications of Discrete-Time systems.

Linear & Non linear system

Causal & Non Causal system

Stable & Un stable system

Static & Dynamic systems

Define Linear system

A system is said to be linear system if it satisfies Super position principle. Let us consider x1(n) & x2(n) be the two input sequences & y1(n) & y2(n) are the responses respectively, T[ax1(n) + bx2(n)] = a y1(n) + by2(n)

10. Define Static & Dynamic systems

When the output of the system depends only upon the present input sample, then it is called static system, otherwise if the system depends past values of input then it is called dynamic system.

11. Define causal system.

When the output of the system depends only upon the present and past input sample, then it is called causal system, otherwise if the system depends on future values of input then it is called non-causal system.

12. Define Shift-Invariant system.

If y(n) is the response to an input x(n), then the response to an input X(n) = x(n-n0) then y(n) = y(n-n0) When the system satisfies above condition then it is said to shift in variant, otherwise it is variant

13. Define impulse and unit step signal.

Impulse signal (n): The impulse signal is defined as a signal having unit magnitude at n = 0 and zero for other values of n. (n) = 1; n = 0; **Unit step signal u(n):** The unit step signal is defined as a signal having unit magnitude for all values of n 0 u(n) = 1; n 0 0; n 0

14. What are FIR and IIR systems?

The impulse response of a system consist of infinite number of samples are called IIR system & the impulse response of a system consist of finite number of samples are called FIR system.

What are the basic elements used to construct the block diagram of discrete time system?

The basic elements used to construct the block diagram of discrete time Systems are Adder, Constant multiplier &Unit delay element.

What is ROC in Z-Transform?

The values of z for which z – transform converges is called region of convergence (ROC). The z-transform has an infinite power series; hence it is necessary to mention the ROC along with z-transform.

List any four properties of Z-Transform.

Linearity
Time Shifting
Frequency shift or Frequency translation
Time reversal

What are the different methods of evaluating inverse z-transform?

Partial fraction expansion
Power series expansion
Contour integration (Residue method)

Define sampling theorem.

A continuous time signal can be represented in its samples and recovered back if the sampling frequency Fs ≥ 2B. Here ‗Fs‘ is the sampling frequency and ‗B‘ is the maximum frequency present in the signal.

Check the linearity and stability of g(n),

Since square root is nonlinear, the system is nonlinear.

As long as x(n) is bounded, its square root is bounded. Hence this system is stable.

What are the properties of convolution?

Commutative property x(n) * h(n) = h(n) * x(n)

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

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

UNIT-II DISCRETE TIME SYSTEM ANALYSIS

Define DTFT. Let us consider the discrete time signal x(n).Its DTFT is denoted as X(w).It is given as X(w)= x(n)e-jwn

State the condition for existence of DTFT? The conditions are • If x(n)is absolutely summable then |x(n)|< • If x(n) is not absolutely summable then it should have finite energy for DTFT to exit.

List the properties of DTFT. Periodicity Linearity Time shift Frequency shift Scaling Differentiation in frequency domain Time reversal Convolution Multiplication in time domain Parseval's theorem

What is the DTFT of unit sample?

Define DFT. DFT is defined as X(w)= x(n)e-jwn. Here x(n) is the discrete time sequence X(w) is the fourier transform ofx(n).

Define Twiddle factor. The Twiddle factor is defined as WN=e-j2 /N

Define Zero padding. The method of appending zero in the given sequence is called as Zero padding.

Define circularly even sequence. A Sequence is said to be circularly even if it is symmetric about the point zero on the circle. x(N-n)=x(n),1<=n<=N-1.

Define circularly odd sequence. A Sequence is said to be circularly odd if it is anti symmetric about point x(0) on the circle.

Define circularly folded sequences. A circularly folded sequence is represented as x((-n))N. It is obtained by plotting x(n) in clockwise direction along the circle.

State circular convolution. This property states that multiplication of two DFT is equal to circular convolution of their sequence in time domain.

State parseval's theorem. Consider the complex valued sequences x(n) and y(n).If x(n)y*(n)=1/N X(k)Y*(k)

Define Z transform. The Z transform of a discrete time signal x(n) is denoted by X(z) and is given by X(z)= x(n)Z-n.

Define ROC. The value of Z for which the Z transform converged is called region of convergence. **15. Find Z transform of x(n)={1,2,3,4}** x(n)= {1,2,3,4} X(z)= x(n)z-n = 1+2z-1+3z-2+4z-3. = 1+2/z+3/z2+4/z3.

State the convolution property of Z transform. The convolution property states that the convolution of two sequences in time domain is equivalent to multiplication of their Z transforms.

What z transform of (n-m)? By time shifting property Z[A (n-m)]=AZ-m sin Z[(n)] =1

State initial value theorem. If x(n) is causal sequence then its initial value is given by x(0)=lim X(z).

List the methods of obtaining inverse Z transform. Inverse z transform can be obtained by using Partial fraction expansion.

Contour integration Power series expansion Convolution. **20. Obtain the inverse z transform of X(z)=1/z-a,|z|>|a|** Given X(z)=z-1/1-az-1 By time shifting property X(n)=an.u(n-1)

UNIT-III - DISCRETE FOURIER TRANSFORM AND COMPUTATION

1.What is DFT?
It is a finite duration discrete frequency sequence, which is obtained by sampling one period of Fourier transform. Sampling is done at N equally spaced points over the period extending from w=0 to 2л.

2. Define N point DFT.
The DFT of discrete sequence x(n) is denoted by X(K). It is given by, Here k=0,1,2…N-1 Since this summation is taken for N points, it is called as N-point DFT.

What is DFT of unit impulse δ(n)? The DFT of unit impulse δ(n) is unity.

List the properties of DFT.
Linearity, Periodicity, Circular symmetry, symmetry, Time shift, Frequency shift, complex conjugate, convolution, correlation and Parseval‵s theorem.

5. State Linearity property of DFT.
DFT of linear combination of two or more signals is equal to the sum of linear combination of DFT of individual signal.

6. When a sequence is called circularly even?
The N point discrete time sequence is circularly even if it is symmetric about the point zero on the circle.

7. What is the condition of a sequence to be circularly odd?
An N point sequence is called circularly odd it if is antisymmetric about point zero on the circle.

8. Why the result of circular and linear convolution is not same?
Circular convolution contains same number of samples as that of x (n) and h (n), while in linear convolution, number of samples in the result (N) are, N=L+M-1 Where L= Number of samples in x (n) M=Number of samples in h (n)

9. What is circular time shift of sequence?
Shifting the sequence in time domain by ‗1‘ samples is equivalent to multiplying the sequence in frequency domain by WNkl

What is the disadvantage of direct computation of DFT?
For the computation of N-point DFT, N2 complex multiplications and N[N-1] Complex additions are required. If the value of N is large than the number of computations will go into lakhs. This proves inefficiency of direct DFT computation.

What is the way to reduce number of arithmetic operations during DFT computation?
Number of arithmetic operations involved in the computation of DFT is greatly reduced by using different FFT algorithms as follows. 1. Radix-2 FFT algorithms. - Radix-2 Decimation in Time (DIT) algorithm.
- Radix-2 Decimation in Frequency (DIF) algorithm.

Radix-4 FFT algorithm.

What is the computational complexity using FFT algorithm?
Complex multiplications = N/2 log2N
Complex additions = N log2N

13. How linear filtering is done using FFT?

Correlation is the basic process of doing linear filtering using FFT. The correlation is nothing but the convolution with one of the sequence, folded. Thus, by folding the sequence h (n), we can compute the linear filtering using FFT.

14. What is zero padding? What are its uses?

Let the sequence x (n) has a length L. If we want to find the N point DFT (N>L) of the sequence x (n). This is known as zero padding. The uses of padding a sequence with zeros are

We can get _better display‘ of the frequency spectrum.

With zero padding, the DFT can be used in linear filtering.

15. Why FFT is needed?

The direct evaluation of the DFT using the formula requires N2 complex multiplications and N (N-1) complex additions. Thus for reasonably large values of N (inorder of 1000) direct evaluation of the DFT requires an inordinate amount of computation. By using FFT algorithms the number of computations can be reduced. For example, for an N-point DFT, The number of complex multiplications required using FFT is N/2log2N. If N=16, the number of complex multiplications required for direct evaluation of DFT is 256, whereas using DFT only 32 multiplications are required.

16. What is the speed of improvement factor in calculating 64-point DFT of a sequence using direct computation and computation and FFT algorithms? Or Calculate the number of multiplications needed in the calculation of DFT and FFT with 64-point sequence.

The number of complex multiplications required using direct computation is N2=642=4096. The number of complex multiplications required using FFT is N/2 log2N = 64/2log264=192. Speed improvement factor = 4096/192=21.33

17. What is the main advantage of FFT?

FFT reduces the computation time required to compute discrete Fourier transform.

18. Calculate the number of multiplications needed in the calculation of DFT using FFT algorithm with using FFT algorithm with 32-point sequence.

For N-point DFT the number of complex multiplications needed using FFT algorithm is N/2 log2N. For N=32, the number of the complex multiplications is equal to 32/2log232=16*5=80.

19. What is FFT?

The fast Fourier transforms (FFT) is an algorithm used to compute the DFT. It makes use of the Symmetry and periodically properties of twiddles factor WKN to effectively reduce the DFT computation time. It is based on the fundamental principle of decomposing the computation of the DFT of a sequence of length N into successively smaller discrete Fourier transforms. The FFT algorithm provides speed-increase factors, when compared with direct computation of the DFT, of approximately 64 and 205 for 256-point and 1024-point transforms, respectively.

20. How many multiplications and additions are required to compute N-point DFT using redix-2 FFT?

The number of multiplications and additions required to compute N-point DFT using redix-2 FFT are N log2N and N/2 log2N respectively.

21. What is meant by radix-2 FFT?

The FFT algorithm is most efficient in calculating N-point DFT. If the number of output points N can be expressed as a power of 2, that is, N=2M, where M is an integer, Then this algorithm is known as radix-s FFT algorithm.

22. What is a decimation-in-time algorithm?

Decimation-in-time algorithm is used to calculate the DFT of a N-point Sequence. The idea is to break the N-point sequence into two sequences, the DFTs of which can be combined to give the DFT of the original N-point sequence. Initially the N-point sequence is divided into two N/2-point sequences xe(n) and x0(n), which have the even and odd members of x(n) respectively. The N/2 point DFTs of these two sequences are evaluated and combined to give the N point DFT. Similarly the N/2 point DFTs can be expressed as a combination of N/4 point DFTs. This process is continued till we left with 2-point DFT. This algorithm is called Decimation-in-time because the sequence x(n) is often splitted into smaller sub sequences.

What are the differences and similarities between DIF and DIT algorithms?

Differences:

For DIT, the input is bit reversal while the output is in natural order, whereas for DIF, the input is in natural order while the output is bit reversed.

The DIF butterfly is slightly different from the DIT butterfly, the difference being that the complex multiplication takes place after the add-subtract operation in DIF.

Similarities: Both algorithms require same number of operations to compute the DFT. Bot algorithms can be done in place and both need to perform bit reversal at some place during the computation.

What are the applications of FFT algorithms?

Linear filtering

Correlation

Spectrum analysis

25. What is a decimation-in-frequency algorithm?

In this the output sequence X (K) is divided into two N/2 point sequences and each N/2 point sequences are in turn divided into two N/4 point sequences.

26. Distinguish between DFT and DTFT.

S.No	DFT	DTFT
1.	Obtained by performing sampling operation in both the time and frequency domains.	Sampling is performed only in time domain
2.	Discrete frequency spectrum	Continuous function of ω

27. Distinguish between Fourier series and Fourier transform.

S.No	Fourier Series	Fourier transform
1.	Gives the harmonic content of a periodic time function	Gives the frequency information for an aperiodic signal.
2.	Discrete frequency spectrum	Continuous frequency spectrum

UNIT-IV - DESIGN OF DEGITAL FILTER

1) Define IIR filter?
IIR filter has Infinite Impulse Response.

What are the various methods to design IIR filters?

Approximation of derivatives
Impulse invariance
Bilinear transformation.

Which of the methods do you prefer for designing IIR filters? Why?
Bilinear transformation is best method to design IIR filter, since there is no aliasing in it.

What is the main problem of bilinear transformation?
Frequency warping or nonlinear relationship is the main problem of bilinear transformation.

5) What is prewarping?
Prewarping is the method of introducing nonlinearly in frequency relationship to compensate warping effect.

State the frequency relationship in bilinear transformation?
= 2 tan (w/2) T

Where the j axis of s-plane is mapped in z-plane in bilinear transformation?
The j axis of s-plane is mapped on the unit circle in z-plane in bilinear transformation

Where left hand side and right hand side are mapped in z-plane in bilinear transformation?
Left hand side -- Inside unit circle Right hand side – Outside unit circle

What is the frequency response of Butterworth filter?
Butterworth filter has monotonically reducing frequency response.

Which filter approximation has ripples in its response?
Chebyshev approximation has ripples in its pass band or stop band.

Can IIR filter be designed without analog filters?

What is the advantage of designing IIR Filters using pole-zero plots?
The frequency response can be located exactly with the help of poles and zeros.

Compare the digital and analog filter.

S.No	Digital filter	Analog filter
1.	Operates on digital samples of the signal.	Operates on analog signals.
2.	It is governed by linear difference equation.	It is governed by linear difference equation.
3.	It consists of adders, multipliers and delays implemented in digital logic.	It consists of electrical components like resistors, capacitors and inductors.
4.	In digital filters the filter coefficients are designed to satisfy the desired frequency	In digital filters the approximation problem is solved to satisfy the desired

	response.	frequency response.

What are the advantages and disadvantages of digital filters?

Advantages of digital filters

High thermal stability due to absence of resistors, inductors and capacitors.

Increasing the length of the registers can enhance the performance characteristics like accuracy, dynamic range, stability and tolerance.

The digital filters are programmable.

Multiplexing and adaptive filtering are possible.

The bandwidth of the discrete signal is limited by the sampling frequency.

The performance of the digital filter depends on the hardware used to implement the filter.

What is impulse invariant transformation?

The transformation of analog filter to digital filter without modifying the impulse response of the filter is called impulse invariant transformation.

Obtain the impulse response of digital filter to correspond to an analog filter with impulse response ha(t) = 0.5 e-2t and with a sampling rate of 1.0kHz using impulse invariant method.

How analog poles are mapped to digital poles in impulse invariant transformation?

In impulse invariant transformation the mapping of analog to digital poles are as follows,

The analog poles on the left half of s-plane are mapped into the interior of unit circle in z-plane. The analog poles on the imaginary axis of s-plane are mapped into the unit circle in the z-plane.

The analog poles on the right half of s-plane are mapped into the exterior of unit circle in z-plane.

What is the importance of poles in filter design?

The stability of a filter is related to the location of the poles. For a stable analog filter the poles should lie on the left half of s-plane. For a stable digital filter the poles should lie inside the unit circle in the z-plane.

Why an impulse invariant transformation is not considered to be one-to-one?

In impulse invariant transformation any strip of width 2π/T in the s-plane for values of s-plane in the range (2k-1)/T < Ω <(2k-1) π/T is mapped into the entire z-plane. The left half of each strip in s-plane is mapped into the interior of unit circle in z-plane, right half of each strip in s-plane is mapped into the exterior of unit circle in z-plane and the imaginary axis of each strip in s-plane is mapped on the unit circle in z-plane. Hence the impulse invariant transformation is many-to-one.

What is Bilinear transformation?

The bilinear transformation is conformal mapping that transforms the s-plane to z-plane. In this mapping the imaginary axis of s-plane is mapped into the unit circle in z-plane, The left half of s-plane is mapped into interior of unit circle in z-plane and the right half of s-plane is mapped into exterior of unit circle in z-plane. The Bilinear mapping is a one-to-one mapping and it is accomplished when

How the order of the filter affects the frequency response of Butterworth filter.

The magnitude response of butterworth filter is shown in figure, from which it can be observed that the magnitude response approaches the ideal response as the order of the filter is increased.

Write the properties of Chebyshev type –1 filters.

The magnitude response is equiripple in the passband and monotonic in the stopband.
The chebyshev type-1 filters are all pole designs.
The normalized magnitude function has a value of at the cutoff frequency
c.
The magnitude response approaches the ideal response as the value of N increases.

Compare the Butterworth and Chebyshev Type-1 filters.

S.No	Butterworth	Chebyshev Type - 1
1.	All pole design.	All pole design
2.	The poles lie on a circle in s-plane	The poles lie on a ellipse in s-plane.
3.	The magnitude response is maximally flat at the origin and monotonically decreasing function of c..	. The magnitude response is equiripple in passband and monotonically decreasing in the stopband.
4.	. The normalized magnitude response has a value of $1 / \sqrt{2}$ at the cutoff frequency c.	The normalized magnitude response has a value of $1 / \sqrt{(1+\epsilon^2)}$ at the cutoff frequency c.
5.	Only few parameters has to be calculated to determine the transfer function.	A large number of parameters has to be calculated to determine the transfer function

22. What is FIR filters?

The specifications of the desired filter will be given in terms of ideal frequency response Hd(w). The impulse response hd(n) of the desired filter can be obtained by inverse fourier transform of Hd(w), which consists of infinite samples. The filters designed by selecting finite number of samples of impulse response are called FIR filters.

23. What are the different types of filters based on impulse response?

Based on impulse response the filters are of two types 1. IIR filter 2. FIR filter The IIR filters are of recursive type, whereby the present output sample depends on the present input, past input samples and output samples. The FIR filters are of non recursive type, whereby the present output sample depends on the present input, and previous output samples.

24. What are the different types of filter based on frequency response?

The filters can be classified based on frequency response. They are I) Low pass filter ii) High pass filter iii) Band pass filter iv) Band reject filter.

26. What are the techniques of designing FIR filters?

There are three well-known methods for designing FIR filters with linear phase. These are 1) windows method 2) Frequency sampling method 3) Optimal or minimax design.

27. State the condition for a digital filter to be causal and stable.

A digital filter is causal if its impulse response h(n) = 0 for n<0 A digital filter is stable if its impulse response is absolutely summable,

What is the reason that FIR filter is always stable?

FIR filter is always stable because all its poles are at origin.

What are the properties of FIR filter?

FIR filter is always stable.

A realizable filter can always be obtained.

FIR filter has a linear phase response.

30. How phase distortion and delay distortions are introduced?

The phase distortion is introduced when the phase characteristics of a filter is not linear within the desired frequency band. The delay distortion is introduced when the delay is not constant within the desired frequency range.

Write the steps involved in FIR filter design.

Choose the desired (ideal) frequency response Hd(w).
Take inverse fourier transform of Hd(w) to get hd(n).
Convert the infinite duration hd(n) to finite duration h(n).
Take Z-transform of h(n) to get the transfer function H(z) of the FIR filter.

What are the advantages of FIR filters?

Linear phase FIR filter can be easily designed.
Efficient realization of FIR filter exist as both recursive and nonrecursive structures.
FIR filters realized nonrecursively are always stable.
The roundoff noise can be made small in nonrecursive realization of FIR filters.

What are the disadvantages of FIR filters?

The duration of impulse response should be large to realize sharp cutoff filters.
The non-integral delay can lead to problems in some signal processing applications.

What is the necessary and sufficient condition for the linear phase characteristic of an FIR filter?

The necessary and sufficient condition for the linear phase characteristic of an FIR filter is that the phase function should be a linear function of w, which in turn requires constant phase and group delay.

What are the conditions to be satisfied for constant phase delay in linear phase FIR filters?

The conditions for constant phase delay ARE Phase delay, $\alpha = (N-1)/2$ (i.e., phase delay is constant) Impulse response, $h(n) = -h(N-1-n)$ (i.e., impulse response is antisymmetric)

How constant group delay & phase delay is achieved in linear phase FIR filters?

The following conditions have to be satisfied to achieve constant group delay & phase delay. Phase delay, $\alpha = (N-1)/2$ (i.e., phase delay is constant) Group delay, $\beta = \pi/2$ (i.e., group delay is constant) Impulse response, $h(n) = -h(N-1-n)$ (i.e., impulse response is antisymmetric)

What are the possible types of impulse response for linear phase FIR filters?

There are four types of impulse response for linear phase FIR filters

- Symmetric impulse response when N is odd.
- Symmetric impulse response when N is even.
- Antisymmetric impulse response when N is odd.
- Antisymmetric impulse response when N is even.

List the well-known design techniques of linear phase FIR filters.

There are three well-known design techniques of linear phase FIR filters. They are

- Fourier series method and window method
- Frequency sampling method.
- Optimal filter design methods.

What is Gibb's phenomenon (or Gibb's Oscillation)?

In FIR filter design by Fourier series method the infinite duration impulse response is truncated to finite duration impulse response. The abrupt truncation of impulse response introduces oscillations in the passband and stopband. This effect is known as Gibb‘s phenomenon (or Gibb‘s Oscillation).

40. When cascade form realization is preferred in FIR filters?

The cascade form realization is preferred when complex zeros with absolute magnitude less than one.

What are the desirable characteristics of the frequency response of window function?

The desirable characteristics of the frequency response of window function are

- The width of the mainlobe should be small and it should contain as much of the total energy as possible.
- The sidelobes should decrease in energy rapidly as w tends to π.

Write the procedure for designing FIR filter using frequency-sampling method.

- Choose the desired (ideal) frequency response Hd(w).
- Take N-samples of Hd(w) to generate the sequence
- Take inverse DFT of to get the impulse response h(n).
- The transfer function H(z) of the filter is obtained by taking z-transform of impulse response.

What are the drawback in FIR filter design using windows and frequency sampling method? How it is overcome?

The FIR filter design using windows and frequency sampling method does not have Precise control over the critical frequencies such as wp and ws. This drawback can be overcome by designing FIR filter using Chebyshev approximation technique.In this technique an error function is used to approximate the ideal frequency response, in order to satisfy the desired specifications.

Write the characteristic features of rectangular window.

The mainlobe width is equal to 4π/N.
The maximum sidelobe magnitude is –13dB.
The sidelobe magnitude does not decrease significantly with increasing w.

List the features of FIR filter designed using rectangular window.

The width of the transition region is related to the width of the mainlobe of window spectrum.
Gibb‘s oscillations are noticed in the passband and stopband.
The attenuation in the stopband is constant and cannot be varied.

Why Gibb's oscillations are developed in rectangular window and how it can be eliminated or reduced?

The Gibb‘s oscillations in rectangular window are due to the sharp transitions from to 0 at the edges of window sequence. These oscillations can be eliminated or reduced by replacing the sharp transition by gradual transition. This is the motivation for development of triangular and cosine windows.

List the characteristics of FIR filters designed using windows.

The width of the transition band depends on the type of window.
The width of the transition band can be made narrow by increasing the value of N where N is the length of the window sequence.
The attenuation in the stop band is fixed for a given window,except in case of Kaiser window where it is variable.

Compare the rectangular window and hanning window.

Rectangular window	Hanning Window
i) The width of mainlobe in window spectrum is 4π/N	i)The width of mainlobe in window spectrum is 8π/N
ii) The maximum sidelobe magnitude in window spectrum is –13dB.	ii) The maximum sidelobe magnitude in window spectrum is –31dB.
iii) In window spectrum the sidelobe magnitude slightly decreases with increasing w.	iii) In window spectrum the sidelobe magnitude decreases with increasing w.
iv) In FIR filter designed using rectangular window the minimum stopband attenuation is 22dB.	iv) In FIR filter designed using hanning window the minimum stopband attenuation is 44dB.

49. Compare the rectangular window and hamming window.

Rectangular window	Hamming Window
i) The width of mainlobe in window spectrum is 4π/N	i)The width of mainlobe in window spectrum is 8π/N
ii) The maximum sidelobe magnitude in window spectrum is –13dB.	ii) The maximum sidelobe magnitude in window spectrum is –41dB.
iii) In window spectrum the sidelobe magnitude slightly decreases with increasing w.	iii) In window spectrum the sidelobe magnitude remains constant.
iv) In FIR filter designed using rectangular window the minimum stopband attenuation is 22dB.	iv) In FIR filter designed using hamming window the minimum stopband attenuation is 44dB.

Write the characteristic features of hanning window spectrum.

The mainlobe width is equal to 8π/N.
The maximum sidelobe magnitude is –41dB.
The sidelobe magnitude remains constant for increasing w.

What is the mathematical problem involved in the design of window function?

The mathematical problem involved in the design of window function(or sequence) is that of finding a time-limited function whose Fourier Transform best approximates a band limited function. The approximation should be such that the maximum energy is confined to mainlobe for a given peak sidelobe amplitude.

List the desirable features of Kaiser Window spectrum.

The width of the mainlobe and the peak sidelobe are variable.
The parameter α in the Kaiser Window function is an independent variable that can be varied to control the sidelobe levels with respect to mainlobe peak.
The width of the mainlobe in the window spectrum can be varied by varying the length N of the window sequence.

Compare the hamming window and Kaiser window.

Hamming Window	Kaiser Window
i)The width of main lobe in window spectrum is 8π/N	i) The width of main lobe in window spectrum depends on the values of α & N.
ii) The maximum sidelobe magnitude in window spectrum is –41dB.	ii) The maximum side lobe magnitude with respect to peak of main lobe is variable using the parameter α.
iii) In window spectrum the side	iii) In window spectrum the side

lobe magnitude remains constant.	lobe magnitude decreases with increasing w.
iv) In FIR filter designed using hamming window the minimum stop band attenuation is 44dB. .	iv) In FIR filter designed using Kaiser window the minimum stop band attenuation is variable and depends on the value of α.

UNIT V - DIGITAL SIGNAL PROCESSOR

1. Write short notes on general purpose DSP processors

General-purpose digital signal processors are basically high speed microprocessors with hard ware architecture and instruction set optimized for DSP operations. These processors make extensive use of parallelism, Harvard architecture, pipelining and dedicated hardware whenever possible to perform time consuming operations .

Write notes on special purpose DSP processors.

There are two types of special; purpose hardware.

(i) Hardware designed for efficient execution of specific DSP algorithms such as digital filter, FFT.

(ii) Hardware designed for specific applications, for example telecommunication, digital audio.

Briefly explain about Harvard architecture.

The principal feature of Harvard architecture is that the program and the data memories lie in two separate spaces, permitting full overlap of instruction fetch and execution. Typically these types of instructions would involve their distinct type.

Instruction fetch

Instruction decode

Instruction execute

Briefly explain about multiplier accumulator. The way to implement the correlation and convolution is array multiplication Method. For getting such as digital filter, FFT.

(ii) Hardware designed for specific applications, for example telecommunication, digital audio.

What are the types of MAC is available? There are two types MAC‘S available

Dedicated & integrated

Separate multiplier and integrated unit

What is meant by pipeline technique?

The pipeline technique is used to allow overall instruction executions to overlap. That is where all four phases operate in parallel. By adapting this technique, execution speed is increased.

What are four phases available in pipeline technique? The four phases are

(i) Fetch (ii)

Decode

(iii)Read (iv)

Execution

In a non-pipeline machine, the instruction fetch, decode and execute take 30 ns, 45 ns and 25 ns respectively. Determine the increase in throughput if the instruction were pipelined. Assume a 5ns pipeline overhead in each stage and ignore other delays. The average instruction time is = 30 ns+45 ns + 25 ns = 100 ns Each instruction has been completed in three cycles = 45 ns * 3 = 135ns Throughput of the machine = The average instruction time/Number of M/C per instruction = 100/135 = 0.7407 But in the case of pipeline machine, the clock speed is determined by the speed of the slowest stage plus overheads. In our case is = 45 ns + 5 ns =50 ns The respective throughput is = 100/50 = 2.00 The amount of speed up the operation is = 135/50 = 2.7 times

9. **Assume a memory access time of 150 ns, multiplication time of 100 ns, addition time of 100 ns and overhead of 10 ns at each pipe stage. Determine the throughput of MAC** After getting successive addition and multiplications The total time delay is 150 + 100 + 100 + 5 = 355 ns System throughput is = 1/355 ns.

10.Write down the name of the addressing modes. Direct addressing. Indirect addressing. Bit-reversed addressing

Immediate addressing.

i. Short immediate addressing.

ii. Long immediate addressing.

Circular addressing.

11.What are the instructions used for block transfer in C5X Processors? The BLDD, BLDP and BLPD instructions use the BMAR to point at the source or destination space of a block move. The MADD and MADS also use the BMAR to address an operand in program memory for a multiply accumulator operation

12.**Briefly explain about the dedicated register addressing modes.** The dedicated-registered addressing mode operates like the long immediate addressing modes, except that the address comes from one of two special-purpose memory-mapped registers in the CPU: the block move address register (BMAR) and the dynamic bit manipulation register (DBMR). The advantage of this addressing mode is that the address of the block of memory to be acted upon can be changed during execution of the program.

Briefly explain about bit-reversed addressing mode?

In the bit-reversed addressing mode, INDX specifies one-half the size of the FFT. The value contained in the current AR must be equal to 2n-1, where n is an integer, and the FFT size is 2n. An auxiliary register points to the physical location of a data value. When we add INDX t the current AR using bit reversed addressing, addresses are generated in a bit-reversed fashion. Assume that the auxiliary registers are eight bits long, that AR2 represents the base address of the data in memory (0110 00002), and that INDX contains the value 0000 10002.

Briefly explain about circular addressing mode.

Many algorithms such as convolution, correlation, and finite impulse response (FIR) filters can use circular buffers in memory to implement a sliding window; which contains the most recent data to be processed. The _C5x supports two concurrent circular buffer operating via the ARs. The following five memory-mapped registers control the circular buffer operation.

CBSR1- Circular buffer 1 start register.

CBSR2- Circular buffer 2 start Register, CBER1- Circular buffer 1 end register

CBER2- Circular buffer 2 end register

CBCR - Circular buffer control register.

Write the name of various part of C5X hardware.

Central arithmetic logic unit (CALU)

Parallel logic unit (PLU)

Auxiliary register arithmetic unit (ARAU)

Memory-mapped registers.

Program controller.

Write short notes about arithmetic logic unit and accumulator.

The 32-bit general-purpose ALU and ACC implement a wide range of arithmetic and logical functions, the majority of which execute in a single clock cycle. Once an operation is performed in the ALU, the result is transferred to the ACC, where additional operations, such as shifting, can occur. Data that is input to the ALU can be scaled by the prescaler. The following steps occur in the implementation of a typical ALU instruction:

Data is fetched from memory on the data bus,

Data is passed through the prescaler and the ALU, where the arithmetic is performed, and

The result is moved into the ACC.

The ALU operates on 16-bit words taken from data memory or derived from immediate instructions. In addition to the usual arithmetic instructions, the ALU can perform Boolean operations, thereby facilitating the bit manipulation ability required of high-speed controller. One input to the ALU is always supplied by the ACC. The other input can be transferred from the PREG of the multiplier, the ACCB, or the output of the prescaler. After the ALU has performed the arithmetic or logical operation, the result is stored in the ACC.

17. Write short notes about parallel logic unit.

The parallel logic unit (PLU) can directly set, clear, test, or toggle multiple bits in control/status register pr any data memory location. The PLU provides a direct logic operation path to data memory values without affecting the contents of the ACC or the PREG.

18. What is meant by auxiliary register file?

The auxiliary register file contains eight memory-mapped auxiliary registers (AR0-AR7), which can be used for indirect addressing of the data memory or for temporary data storage. Indirect auxiliary register addressing allows placement of the data memory address of an instruction operand into one of the AR. The ARs are pointed to by a 3-bit auxiliary register pointer (ARP) that is loaded with a value from 0-7, designating AR0-AR7, respectively.

19. Write short notes about circular registers in C5X.

The _C5x devices support two concurrent circular buffers operating in conjunction with user-specified auxiliary register. Two 16-bit circular buffer start registers (CBSR1 and CBSR2) indicate the address where the circular buffer starts. Two 16-bit circular buffer end registers (CBER1 and CBER2) indicate the address where the circular buffer ends. The 16-bit circular buffer control register (CBCR) controls the operation of these circular buffers and identifies the auxiliary registers to be used.

What are the factors that influence selection of DSPs?

* Architectural features
* Execution speed *
Type of arithmetic *
Word length

What are the classification digital signal processors?

The digital signal processors are classified as
(i) General purpose digital signal processors.
(ii) Special purpose digital signal processors.

What are the applications of PDSPs?

Digital cell phones, automated inspection, voicemail, motor control, video conferencing, noise cancellation, medical imaging, speech synthesis, satellite communication etc.

23. Give some examples for fixed point DSPs.

TM32OC50, TMS320C54, TMS320C55, ADSP-219x, ADSP-219xx..

Give some example for floating point DSPs?

TMS320C3x, TMS320C67x, ADSP-21xxx

What is pipelining?

Pipelining a processor means breaking down its instruction into a series of discrete pipeline stages which can be completed in sequence by specialized hardware.

26. What is pipeline depth?

The number of pipeline stages is referred to as the pipeline depth.

What are the advantages of VLIW architecture?

Advantages of VLIW architecture
a. Increased performance
b. Better compiler targets
c. Potentially easier to program
d. Potentially scalable
e. Can add more execution units; allow more instructions to be packed into the VLIW instruction.

What are the disadvantages of VLIW architecture?

Disadvantages of VLIW architecture
f. New kind of programmer/compiler complexity
g. Program must keep track of instruction scheduling
h. Increased memory use
i. High power consumption

What is the pipeline depth of TMS320C50 and TMS320C54x?

TMS320C50 – 4
TMS320C54x – 6

What are the different buses of TMS320C5x?

The _C5x architecture has four buses
j. Program bus (PB)
k. Program address bus (PAB)
l. Data read bus (DB)
m. Data read address bus (DAB)

31. Give the functions of program bus?

The program bus carries the instruction code and immediate operands from program memory to the CPU.

32. Give the functions of program address bus?

The program address bus provides address to program memory space for both read and write.

33. Give the functions of data read bus?

The data read bus interconnects various elements of the CPU to data memory space.

34. Give the functions of data read address bus?

The data read address bus provides the address to access the data memory space.

What are the different stages in pipelining?

n. The fetch phase
o. The decode phase
p. Memory read phase
q. The execute phase

List the various registers used with ARAU.

Eight auxiliary registers (AR0 – AR7)
Auxiliary register pointer (ARP)
Unsigned 16-bit ALU

What are the elements that the control processing unit of 'C5x consists of the central
processing unit consists of the following elements:

r. Central arithmetic logic unit (CALU)
s. Parallel logic unit (PLU)
t. Auxiliary register arithmetic unit (ARAU)
u. Memory mapped registers
v. Program controller

What is the function of parallel logic unit?

The parallel logic unit is a second logic unit that executes logic operations on data without affecting the contents of accumulator.

39. List the on chip peripherals in 'C5x.

The on-chip peripherals interfaces connected to the ‗C5x CPU include
w. Clock generator
x. Hardware timer
y. Software programmable wait state generators
z. General purpose I/O pins
Parallel I/O ports
Serial port interface
Buffered serial port
Time-division multiplexed (TDM) serial port
Host port interface
User unmask able interrupts

40.What are the arithmetic instructions of 'C5x?

ADD, ADDB, ADDC, SUB, SUBB, MPY, MPYU

What are the shift instructions?

ROR, ROL, ROLB, RORB, BSAR.

What are the general purpose I/O pins?

Branch control input (BIO)

External flag (XF)

42. What are the logical instructions of 'C5x?

AND, ANDB, OR, ORB, XOR, XORB

43. What are load/store instructions?

LACB, LACC, LACL, LAMM, LAR, SACB, SACH, SACL, SAR, SAMM.

Mention the addressing modes available in TMS320C5X processor?

Direct addressing mode
Indirect addressing mode
Circular addressing mode
Immediate addressing
Register addressing
Memory mapped register addressing

Give the features of DSPs?

Architectural features
Execution speed
Type of arithmetic
Word length

What is function of NOP instruction?

* NOP- No operation
* Perform no operation.

What is function of ZAC instruction?

ZAC – Zero accumulator
Clear the contents of accumulator to zero.

Give the function of BIT instruction.

BIT – Test bit Copy the specified bit of the data memory value to the TC bit in ST1.

Mention the function of B instruction.

B – Branch conditionally.
Branch to the specified program memory address. Modify the current AR and ARP as specified.

What is use of ADD instruction?

ADD – Add to accumulator with shift.
Add the content of addressed data memory location or an immediate value of accumulator, if a shift is specified, left-shift the data before the add. During shifting, low order bits are Zero-filled, and high-order bits are sign extended if SXM=1.

51. Give the advantages of DSPs?

Architectural features, Execution speed, Type of arithmetic, Word length.

52. Give the applications of DSP Processors?

Digital cell phones, automated inspection, voicemail, motor control, video conferencing, noise cancellation, medical imaging, speech synthesis, satellite communication etc.

IMPORTANT QUESTIONS

UNIT – I INTRODUCTION

PART – A

What is signal and Signal processing?
List the advantages of Digital Signal processing.
Mention few applications of Digital Signal processing.
Classify discrete time signals.
What are Energy and Power signals?
What do you mean by periodic and Aperiodic signals?
When a signal is said to be symmetric and Anti symmetric?
What are deterministic and random signals?
What are the elementary signals?
What are the different types of representation of discrete time signals?
Draw the basic block diagram of digital signal processing of analog signals.
What are the basic time domain operations of discrete time signal?
What is the significance of unit sample response of a system?
Classify discrete time systems.
Whether the system defined by the impulse response h(n) = 2n u(-n) + 2-n u(n) is causal ?
Justify your answer.
Compute the energy of the signal x(n) = 2-n u(n)
Compute the energy of the signal x(n) = (0.5)n u(n)
Define convolution.
List out the properties of convolution.
What do you mean by BIBO stable?
What is linear time invariant system?
Compute the convolution of x(n) = {1,2,1,-1} and h(n) = {1,2,1,-1} using tabulation method.
Check whether the system defined by h (n) = [5 (1/2)n + 4 (1/3)n] u(n) is stable.
Differentiate between analog, discrete, quantized and digital signals.
Differentiate between analog and digital signals.
Differentiate between one dimensional and two dimensional signal with an example for each.
Name any four elementary time domain operations for discrete time signals.
For the signal f (t) = 5 cos (5000πt) + sin2 (3000πt), determine the minimum sampling rate for recovery without aliasing.

$\Omega1 = 5000\pi = 2\pi F1$ $\Omega2 = 3000\pi = 2\pi F2$
F1 = 2.5 kHz F2=1.5 kHz
Fmax = 2.5 kHz
According to sampling theorem Fs ≥ 2 Fmax
So, Fs = 5 kHz

For the signal f (t) = cos2 (4000πt) + 2 sin (6000πt), determine the minimum sampling rate
for recovery without aliasing.
Ω1 = 4000π = 2πF1 Ω2 = 6000π = 2πF2
F1 = 2 kHz F2 =3 kHz
Fmax = 3 kHz
According to sampling theorem Fs ≥ 2 Fmax
So, Fs = 6 kHz

What is sampling?

State sampling theorem and what is Nyquist frequency?

Sampling theorem - Fs ≥ 2 Fmax ,Nyquist frequency or Nyquist rate FN = 2 Fmax

What is known as aliasing?

Define the criteria to perform sampling process without aliasing.

Differentiate between anti aliasing and anti imaging filters.

What are the effects of aliasing?

What is anti aliasing filter? What is the need for it?

Draw the basic block diagram of a digital processing of an analog signal.

Draw the basic structure of linear constant difference equation.

What is sample and Hold circuit?

If a minimum signal to noise ratio (SQR) of 33 dB is desired, how many bits per code word
are required in a linearly quantized
system? SQR =1.76 +6.02b, SQR given is
33 Db 1.76+6.02b = 33 dB
6.02 b = 31.24 b
= 5.18 = 6 bits

Determine the number of bits required in computing the DFT of a 1024 – point sequence
with an SQR of 30 dB
The size of the sequence is N = 1024 = 210
SQR is σX2 / σq2 = 22b / N2 10 log [σX2 / σq2] = 10 log [22b /
N2] N2 = 22010 log [σX2/ σq2] = 10 log [22b / 220]
SNR = 10 log [22b - 20] = 30 dB 3(2b-20) =30 ,b=15 bits is the precision for both multiplication and addition

Determine the system described by the equation y (n) = n x (n) is linear or not.

What is the total energy of the discrete time signal x(n) which takes the value of unity at n
=-1,0,1?

Draw the signal x(n) = u(n) – u(n-3)

For each of the following systems, determine whether the system is static stable, causal,
linear and time invariant
a. y(n) = e x(n)
b. y(n) = ax(n) +b
c. y(n) =

d. y(n) = $_{-}^{+1}{}_{\infty}$ ( )
e. y(n) = n x2(n)
f. y(n) = x(-n+2)
g. y(n) = nx(n)
h. y(n) = x(n) +C
i. y(n) = x(n) – x(n-1)
j. y(n) = x(-n)
k. y(n) = x(n) where x(n) = [x(n+1) – x(n)]
l. y(n) = g(n) x(n)
m. y(n) = x(n2)
n. y(n) = x2(n)
o. y(n) = cos x(n)
p. y(n) = x(n) cos ω0n

Compute the linear convolution of h(n) = {**1,**2,1} and x(n) ={**1**,-3,0,2,2}

Explain the concept of Energy and Power signals and determine whether the following are energy or power signals
a. x(n) = (1/3)n u(n)
b. x(n) = sin (π / 4)n

The unit sample response h(n) of a system is represented by
h (n) = n2u(n+1) – 3 u(n) +2n u(n-1) for -5≤ n ≤5. Plot the unit sample response.

State and prove sampling theorem. How do you recover continuous signals from its samples? Discuss the various parameters involved in sampling and reconstruction.

What is the input x(n) that will generate an output sequence y(n) = {1,5,10,11,8,4,1} for a system with impulse response h(n) = {1,2,1}.

Check whether the system defined by h(n) = [5 (1/2)n +4(1/3)n] u(n) is stable?

Explain the analog to digital conversion process and reconstruction of analog signal from digital signal.

What are the advantages and disadvantages of digital signal processing compared with analog signal processing?

Classify and explain different types of signals.

Explain the various elementary discrete time signals.

Explain the different types of mathematical operations that can be performed on a discrete time signal.

Explain the different types of representation of discrete time signals.

Determine whether the systems having the following impulse responses are causal and stable
a. h(n) = 2n u(-n)
b. h(n) = sin nπ / 2
c. h(n) = sin nπ + δ (n)
d. h(n) = e2n u(n-1)

For the given discrete time signal
x (n) = { -0.5,0.5, for n = -2, -1
1, n = 0
3, 2, 0.4 n > 0}

Sketch the following a) x (n-3), b) x (3-n) c) x (2n) d) x (n/2) e) [x (n) + x (-n)] / 2

Find the convolution of x (n) = an u (n), a < 1 with h(n) = 1for 0 ≤ n ≤ N-1

Draw the analog, discrete, quantized and digital signal with an example.

Explain the properties of linearity and stability of discrete time systems with examples.

The impulse response of a linear time invariant system is h (n) = {1, 2, 1,-1}.

Determine the response of the system to the input signal x (n) = {1, 2, 3, 1}.

Determine whether or not each of the following signals are periodic. If a signal is Periodic specify its fundamental time period.

i. x(t) = 2 cos 3 πt

x(t) = sin 15 πt + sin 20 πt

x(n) = 5 sin 2n

x(n) = cos (n/8) cos (πn / 8)

****

UNIT - II DISCRETE TIME SYSTEM ANALYSIS

PART - A

1. Define Z-transform
2. Define ROC in Z-transform
3. Determine Z-transform of the sequence x(n)= {2,1,-1,0, 3}

Determine Z-transform of x(n) = - 0.5 u (-n-1)

Find Z- transform of x(n) = - bn u(-n-1) and its ROC

Find Z- transform of x(n) = an u(n) and its ROC

What are the properties of ROC in Z- transform?

State the initial value theorem of Z- transforms.

State the final value theorem of Z- transforms.

Obtain the inverse Z – transform of X(Z) = log (1 + Z-1) for Z < 1

Obtain the inverse Z – transform of X(Z) = log (1-2z) for Z < 1/2

What is the condition for stability in Z-domain?

Mention the basic factors that affect thr ROC of z- transform.

Find the z- transform of a digita limpulse signal and digital step signal

PART - B

Determine the Z-transform and ROC of

a. x(n) = rn cos ωn u(n)

b. x(n) = n2an u(n)

c. x(n) = -1/3 (-1/4)n u(n) – 4/3 (2)n u(-n-1)

d. x(n) = an u(n) + bn u(n) + cn u(-n-1) , |a | < | b| < |c|

e. x(n) = cos ωn u(n)

f. x(n) = sin ω0n . u(n)

g. x(n) = an u(n)

h. x(n) = [3 (2n) – 4 (3n)] u(n)

Find the inverse Z-transform of

a. X(z) = z (z+1) / (z-0.5)3

X(z) = 1+3z-1 / 1 + 3z-1 + 2z-2
H(z) = 1 / [1 - 3z-1 + 0.5z-2] |z | > 1
X(z) = [z (z2- 4z +5)] / [(z-3) (z-2) (z-1)] for ROC |2 | < | z| < |3|, |z| > 3, |z|< 1

3. Determine the system function and pole zero pattern for the system described by difference equation y (n) -0.6 y (n-1) +0.5 y (n-2) = x (n) – 0.7 x (n-2)
4. Determine the pole –zero plot for the system described by the difference equation y (n) – 3/4 y (n-1) +1/8 y (n-2) = x(n) – x(n-1)
5. Explain the properties of Z-transform.
6. Perform the convolution of the following two sequences using Z-transforms. x(n) = 0.2n u(n) and h(n) = (0.3)n u(n)
7. A causal LTI system has an impulse response h(n) for which the Z-transform is given by H(z) = (1+z-1) / [(1 + 1/2z-1) (1 + 1/4z-1). What is the ROC of H (z)? Is the system stable? Find the Z-transform X (z) of an input x (n) that will produce the output y(n) = -1/3 (-1/4)n u(n) – 4/3 (2)n u(-n-1).Find the impulse response (n) of the system.
8. Solve the difference equation y(n) -3y(n-1) – 4y(n-2) = 0, n ≥ 0 ,y(-1) = 5
9. Compute the response of the system y(n) = 0.7 y(n-1)-0.12y(n-2) +x(n-1)+ (n-2)to the input x(n) = n u(n)
10. What is ROC? Explain with an example.
11. A causal LTI IIR digital filter is characterized by a constant co-efficient difference equation given by y(n) = x(n-1)-1.2x(n-2)+x(n-3)+1.3 y(n-1) – 1.04 y(n-2)+0.222y(n-3),obtain its transfer function.
12. Determine the system function and impulse response of the system described by the difference equation y(n) = x(n) +2x(n-1)- 4x(n-2) + x(n-3)
13. Solve the difference equation y(n) - 4y(n-1) - +4 y(n-2) = x(n) – x(n-1) with the initial condition y(-1) = y(-2) = 1
14. Find the impulse response of the system described by the difference equation y(n) = 0.7 y(n-1) -0.1 y(n-2) +2 x(n) – x(n-2)
15. Determine the z- transform and ROC of the signal x (n) = [3 (2n) – 4 (3n)] u(n).
16. State and prove convolution theorem in z-transform.
17. Given x(n) = δ(n) + 2 δ(n-1) and y(n) = 3 δ(n+1) + δ(n)- δ(n-1). Find x(n) * y(n) and X(z).Y(z).

UNIT -3 DISCRETE TIME ANALYSIS

PART – A

Compute the DFT of x(n) = δ(n – no)
State and prove the Parseval‘s relation of DFT.
What do you mean by the term bit reversal as applied to DFT?
Define discrete Fourier series.
Draw the basic butterfly diagram of DIF –FFT algorithm.
Compute the DFT of x(n) = an
State the time shifting and frequency shifting properties of DFT.

What is twiddle factor? What are its properties?

Draw the basic butterfly diagram of DIT –FFT algorithm.

Determine the 3 point circular convolution of x(n) = {1,2,3} and h(n) = {0.5,0,1}

If an N-point sequence x(n) has N-point DFT of X(K) then what is the DFT of the following

i) x*(n) ii) x*(N-n) iii) x((n-l))N iv) x(n) ej2πln/N

What is FFT and what are its advantages?

Distinguish between DFT and DTFT (Fourier transform)

What is the basic operation of DIT –FFT algorithm?

What is zero padding? What are its uses?

State and prove Parseval‘s relation for DFT.

Draw the flow graph of radix – 2 DIF - FFT algorithm for N= 4

What do you mean by bit reversal in DFT?

Write the periodicity and symmetry property of twiddle factor.

Give the relationship between z-domain and frequency domain.

Distinguish between discrete Fourier series and discrete Fourier transform.

What is the relationship between Fourier series co-efficient of a periodic sequence and DFT?

What is the circular frequency shifting property of DFT?

Establish the relation between DFT and z-transform.

Define DFT pair.

Define overshoot.

Define Gibbs phenomenon.

How many multiplications and additions are required to compute N-point DFT using radix – 2 FFT?

Perform circular convolution of the sequence using DFT and IDFT technique x1(n) = {2, 1,2,1} x2 (n) = {0,1,2,3} (8)

Compute the DFT of the sequence x(n) = {1,1,1,1,1,1,0,0} (8)

From the first principles obtain the signal flow graph for computing 8 – point DFT using radix-2 DIT FFT algorithm. Using the above compute the DFT of sequence x(n) = {0.5,0.5,0.5,0.5,0,0,0,0} (16)

State and prove the circular convolution property of DFT.Compute the circular convolution of x(n) = {0,1,2,3,4} and h(n) = {0,1,0,0,0} (8)

Perform circular convolution of the sequence using DFT and IDFT technique x1(n) = {1,1,2,1} x2 (n) = {1,2,3,4} (8)

Compute the DFT of the sequence x(n) = {1,1,1,1,1,1,0,0} (8)

From the first principles obtain the signal flow graph for computing 8 – point DFT using radix-2 DIF-FFT algorithm. An 8 point sequence is given by x(n)={2,2,2,2,1,1,1,1} compute its 8 point DFT of x(n) by radix-2 DIF-FFT (16)

Compute 5 point circular convolution of x1(n) = δ (n) +δ (n-1)-δ (n-2) - δ (n-3) and x2(n) = δ (n) – δ (n-2)+ δ (n-4) (8)

Explain any five properties of DFT. (10)

Derive DIF – FFT algorithm. Draw its basic butterfly structure and compute the DFT
$x(n) = (-1)^n$ using radix 2 DIF – FFT algorithm. (16)

Perform circular convolution of the sequence using DFT and IDFT technique
$x_1(n) = \{0,1,2,3\}$ $x_2(n) = \{1,0,0,1\}$ (8)

Compute the DFT of the sequence $x(n) = 1/3\,\delta(n) - 1/3\,\delta(n-1) + 1/3\,\delta(n-2)$ (6)

From the first principles obtain the signal flow graph for computing 8 – point DFT using radix-2 DIT - FFT algorithm. Using the above compute the DFT of sequence $x(n) = \sin n\pi / 4$ for $0 \leq n \leq 7$ (16)

What is circular convolution? Explain the circular convolution property of DFT and compute the circular convolution of the sequence x(n)=(2,1,0,1,0) with itself (8)

Perform circular convolution of the sequence using DFT and IDFT technique $x_1(n) = \{0,1,2,3\}$ $x_2(n) = \{1,0,0,1\}$ (8)

i) Compute the DFT of the sequence $x(n) = (-1)^n$ (4)

What are the differences and similarities between DIT – FFT and DIF – FFT algorithms? (4)

17. From the first principles obtain the signal flow graph for computing 8 – point DFT using radix-2 DIT - FFT algorithm. Using the above compute the DFT of sequence $x(n) = \cos n\pi / 4$ for $0 \leq n \leq 7$ (16)
18. Compute 4-point DFT of the sequence x (n) = (0, 1, 2, 3) (6)
19. Compute 4-point DFT of the sequence x (n) = (1, 0, 0, 1) (6)
20. Explain the procedure for finding IDFT using FFT algorithm (6)
21. Compute the output using 8 point DIT – FFT algorithm for the sequence x(n) = {1,2,3,4,5,6,7,8} (16)
22. Determine the 8-point DFT of the sequence x(n) = {0,0,1,1,1,0,0,0}
23. Find the circular convolution of x(n) = 1,2,3,4} and h(n) = {4,3,2,1}
24. Determine the 8 point DFT of the signal x(n) = {1,1,1,1,1,1,0,0}. Sketch its magnitude and phase.

****

UNIT - IV DESIGN OF DIGITAL FILTERS

PART – A

An analog filter has a transfer function H(s) = 1 / s+2. Using impulse invariance method, obtain pole location for the corresponding digital filter with T = 0.1s.

What is frequency warping in bilinear transformation?

If the impulse response of the symmetric linear phase FIR filter of length 5 is h (n) = {2,3,0,x,y}, find the values of x and y.

4. What is prewarping? Why is it needed?

Find the digital transfer function H (z) by using impulse invariance method for the analog
transfer function H(s) = 1 / s+2.

What are the different structures of realization of FIR and IIR filters?

What are the methods used to transform analog to digital filters?

State the condition for linear phase in FIR filters for symmetric and anti symmetric
response.

Draw a causal FIR filter structure for length M= 5.

What is bilinear transformation? What are its advantages?

Write the equation of Barlett (or) triangular and Hamming window.

Write the equation of Rectangular and Hanning window.

Write the equation of Blackman and Kaiser window.

Write the expression for location of poles of normalized Butter worth filter.

Write the expression for location of poles of normalized Chebyshev filter.

Draw the magnitude response of 3rd order Chebyshev filter.

Draw the magnitude response of 4th order Chebyshev filter.

Draw the basic FIR filter structure.

Draw the direct form – I structure of IIR filter.

Draw the direct form – II structure of IIR filter.

Draw the cascade form realization structure of IIR filter.

Draw the parallel form realization structure of IIR filter.

When cascade form realization structure is preferred in filters?

Distinguish between FIR and IIR filters.

Compare analog and digital filters.

Why FIR filters are always stable?

State the condition for a digital filter to be causal and stable.

What are the desirable characteristics of windows?

Give the magnitude function Butterworth filter. What is the effect of varying the order of N
on magnitude and phase response?

List out the properties of Butterworth filter.

List out the properties of Chebyshev filter.

Give the Chebyshev filters transfer function and draw its magnitude response.

Give the equation for the order _N‘ and cut off frequency Ωc of Butterworth filter

Why impulse invariance method is not preferred in the design of IIR filter other than low
pass filters?

What are the advantages and disadvantages FIR filters?

What are the advantages and disadvantages IIR filters?

What is canonic structure?

If the number of delays in the structure is equal to the order of the difference equation or order of transfer function, then it is called canonic form of realization.

Compare Butterworth and Chebyshev filters.
What are the desirable and undesirable features of FIR filters?
What are the design techniques of designing FIR filters?
Fourier series method
Windowing technique
Frequency sampling method

PART –B

With suitable examples, describe the realization of linear phase FIR filters (8)

Convert the following analog transfer function H(s) = (s+0.2) / [(s+0.2)2 + 4] into equivalent digital transfer function H (z) by using impulse invariance method assuming T= 1 sec. (8)

Convert the following analog transfer function H(s) = 1 / (s+2) (s+4) into equivalent digital transfer function H (z) by using bilinear transformation with T = 0.5 sec.

Convert the following analog transfer function H(s) = (s+0.1) / [(s+0.1)2 + 9] into equivalent digital transfer function H (z) by using impulse invariance method assuming T= sec. (8)

Convert the following analog transfer function H(s) = 2/ (s+1) (s+3) into equivalent digital transfer function H (z) by using bilinear transformation with T = 0.1sec.Draw the diect form – II realization of digital filter. (8)

Design a high pass filter of length 7 samples with cut off frequency of 2 rad / sec using Hamming window. Plot its magnitude and phase response. (16)

For the constraints
$0.8 \le H(\omega) \le 1.0, 0 \le \omega \le 0.2\pi$
$H(\omega) \le 0.2, 0.6\pi \le \omega \le \pi$
With T= 1 sec determine the system function H(z) for a Butterworth filter using bilinear transformation. (16)

Describe the effects of quantization in IIR filter. Consider a first order filter with difference equation y (n) = x (n) + 0.5 y (n-1).Assume that the data register length is bits plus a sign bit. The input x (n) = 0.875δ (n). Explain the limit cycle oscillations in the above filter, if quantization is preferred by means of rounding and signed magnitude representation is used. (16)

With a neat sketch explain the architecture of TMS 320 C54 processor. (16)

For the constraints
$0.7 \le H(\omega) \le 1.0, 0 \le \omega \le \pi/2$
$H(\omega) \le 0.2, 3\pi/4 \le \omega \le \pi$
With T= 1 sec, design a Butterworth filter. (16)

Explain the quantization effects in design of digital filters. (16)
Discuss about the window functions used in design of FIR filters (8)

Obtain the cascade and parallel realization of system described by difference equation
y(n) = -0.1 y(n-1) + 0.2 y(n-2) + 3x(n) +3.6 x(n-1) + 0.6 x(n-2) (10)

Design a digital Butterworth filter satisfying the following constraints with T= 1 sec,
using Bilinear transformation.
$0.707 \leq H(\omega) \leq 1.0, 0 \leq \omega \leq \pi/2$
$H(\omega) \leq 0.2, 3\pi/4 \leq \omega \leq \pi$ (16)

Design a digital Chebyshev filter satisfying the following constraints with T= 1 sec,
using Bilinear transformation.
$0.707 \leq H(\omega) \leq 1.0, 0 \leq \omega \leq \pi/2$
$H(\omega) \leq 0.2, 3\pi/4 \leq \omega \leq \pi$ (16)

Draw and explain cascade form structure for a 6th order FIR filter. (6)

Explain impulse invariance method of digital filter design. (10)

Derive an expression between s- domain and z- domain using bilinear transformation.
Explain frequency warping. (10)

Draw the structure for IIR filter in direct form – I and II for the following transfer Function
$H(z) = (2 + 3 z^{-1}) (4+ 2 z^{-1} +3 z^{-2}) / (1+0.6 z^{-1}) (1+ z^{-1}+0.5 z^{-2})$ (10)

Design a filter with

0 $\pi/4 \leq \omega \leq \pi$ Using a Hamming window with N= 7 (16)

Discuss about frequency transformations in detail. (8)

Design a LPF with

0 $3\pi/4 \leq \omega \leq \pi$ Using a Hamming window with N= 7 (16)

Using the bilinear transformation and a low pass analog Butterworth prototype, design a low pass digital filter operating at a rate of 20 KHz and having pass band extending to a 4 KHz with a maximum pass band attenuation of 0.5 dB and stop band
starting at 5KHzwith a minimum stop band attenuation of 10 dB. (16)

Using the bilinear transformation and a low pass analog Chebyshev type I prototype,
design a low pass digital filter operating at a rate of 20 KHz and having pass band extending to a 4 KHz with a maximum pass band attenuation of 0.5 dB and stop band starting at 5KHzwith a minimum stop band attenuation of 10 dB. (16)

Obtain the cascade realization of linear phase FIR filter having system function H(z) =
$(1+1/2 z^{-1} + z^{-2}) (2 + ¼ z^{-1} +2z^{-2})$ using minimum number of multipliers.(8)

Design an ideal Hilbert transformer having frequency response
$H(e^{j\omega}) = j$ for $-\pi \leq \omega \leq 0$
$= -j$ for $0\leq \omega \leq \pi$ for N=11, using i. rectangular window ii. Blackmann window

Obtain the direct form – I, direct form – II, cascade and parallel form of realization
for the system y(n) = -0.1 y9n-1) + 0.2 y(n-2) + 3 x(n) + 3.6 x (n-1) + 0.6 x(n-2)

Using Bilinear transformation and a low pass analog Butterworth prototype, design a low pass digital filter operating at the rate of 20k Hz and having pass band extending to 4 kHz with maximum pass band attenuation of 10 dB and stop band starting at 5 kHz with a minimum stop band attenuation of 0.5 dB

Using Bilinear transformation and a low pass analog Chebyshev type I prototype,
design a low pass digital filter operating at the rate of 20k Hz and having pass band extending to 4 kHz with maximum pass band attenuation of 10 dB and stop band starting at 5 kHz with a minimum stop band attenuation of 0.5 dB

Design a low pass filter using Hamming window for N=7 for the desired frequency
Response D (ω) = ej3ω for -3π / 4 ≤ ω ≤3π / 4
0 for 3π / 4 ≤ ω ≤ π

Design an ideal differentiator for N=9 using Hanning and triangular window

UNIT - V DSP HARDWARE

PART - A

Compare fixed point arithmetic and floating point arithmetic.
What is product quantization error or product round off error in DSP?
What are the quantization methods?
What is truncation and what is the error that arises due to truncation in floating point numbers?
What is meant by rounding? Discuss its effects?
What are the two kinds of limit cycle oscillations in DSP?
Why is rounding preferred to truncation in realizing digital filters?
What are the 3 quantization errors due to finite word length registers in digital filters?
List out the features of TMS 320 C54 processors.
What are the various interrupt types supported by TMS 320 C54?
Mention the function of program controller of DSP processor TMS 320 C54.
List the elements in program controller of TMS 320C54.
What do you mean by limit cycle oscillations?
What is pipelining? What is the pipeline depth of TMS 320 C54 processor?
What are the different buses of TMS 320 C54 processor?
What are quantization errors due to finite word length registers in digital filters?
Differentiate between Von Neumann and Harvard architecture.
Define limit cycle oscillations in recursive systems.
How to prevent overflow in digital filters?

Describe the function of on chip peripherals of TMS 320 C54 processor. (12)
What are the different buses of TMS 320 C54 processor? Give their functions. (4)
Explain the function of auxiliary registers in the indirect addressing mode to point the data
memory location. (8)
Explain about the MAC unit. (8)
What is meant by instruction pipelining? Explain with an example how pipelining increases through put efficiency. (8)

Explain the operation of TDM serial ports in P-DSPs (8)

Explain the characteristics of a limit cycle oscillation with respect to the system described by the equation y (n) = 0.95 y (n-1) + x (n).Determine the dead band of the filter. (10)

Draw the product quantization noise model of second order IIR filter. (6)

In a cascaded realization of the first order digital filter, the system function of the individual section are H19z) = 1 / (1-0.9 z-1) and H2(z) = 1 / (1-0.8z-1). Draw the product quantization noise model of the system and determine the output noise power. (16)

Explain the statistical characterization of quantization effects in fixed point realization of digital filter. (16)

Give a detailed note on Direct memory Access controller in TMS 320 C54x processor.

Find the effect of quantization on the pole locations of the second order IIR filter Given by H(z) = 1 / (1-0.5z-1) (1- 0.45 z-1) when it is realized in direct form – I and in cascade form. Assume a word length of 3 bits.

Determine the variance of the round off noise at the output of the two cascade realizations of the filters with system functions H1 (z) = 1 / 1-0.5 z-1 ,H2 (z) = 1 / 1-0.25 z-1

Cascade I, H (z) = H1 (Z) H2 (z).

Cascade II, H (z) = H2 (z) H1 (z).

Question Paper Code: 11295

B.E./B.Tech. DEGREE EXAMINATION, APRIL/MAY 2011
Fifth Semester
Electrical and Electronics Engineering
EC 2314 — DIGITAL SIGNAL PROCESSING
(Regulation 2008)

Time : Three hours **Maximum : 100 marks**

Answer ALL questions

PART A — (10 × 2 = 20 marks)

Define sampling theorem.
What is known as Aliasing?
What is meant by ROC?
Obtain the Discrete Fourier series coefficients of $x(n) = \cos wn$.
What is the relation between DFT and Z transform?
Draw the butterfly diagram for DITFFT.
What are the special features of FIR filters?
What is meant by prewarping?
Mention the features of DSP processor.
What is the condition for linear phase in FIR filters?

PART B — (5 × 16 = 80 marks)

11. (a) (i) Check whether the following is linear, time invariant, casual and stable $y(n) = x(n) + nx(n + 1)$. (8) (ii) Check whether the following are energy or power signals

$= (\frac{1}{2}) ()$

$=$ (8)

Or

(b) (i) Describe in detail the process of sampling and quantization. Also determine the expression for quantization liner. (10)

(ii) Check whether the following are periodic

(1) $x(n) = \cos(3 \; n)$

(2) $x(n) = \sin(3n)$. (6)

12. (a) (i) Determine the Z transform of

(1) $=0.$ (5)

(2) $x(n) = 3\,u(n)$ (3)

(ii) Obtain $x(n)$ for the following :

$= \frac{1}{1-1.5^{-1}+0.5^{-2}}$

for ROC : $|z|>1$, $|z| < 0.5$, $0.5 < |z| < 1$. (8)

Or

(b) (i) Determine the linear convolution of the following sequences

1 = 1,2,3,1 2 = 1,2,1, −1 . (6)

↑

(ii) Obtain the system function and impulse response of the following

system $y(n) - 5y(n-1) = x(n) + x(n-1)$. (10)

(a) (i) Explain the following properties of DFT.

Convolution.

Time shifting

Conjugate Symmetry. (10)

Compute the 4 point DFT of $x(n) = \{0,1, 2,3\}$. (6)

Or

(i) Explain the Radix 2 DIFFFT algorithm for 8 point DFT. (8)

Obtain the 8 point DFT using DITFFT algorithm for

$x(n) = \{\underset{\uparrow}{1},1,1,1,1,1,1,1\}$. (8)

14. (a) (i) Realize the following using cascade and parallel form .

$$= \frac{3+3.6^{-1}+0.6^{-2}}{1+0.1^{-1}-0.2^{-2}}$$ (12)

Explain how an analog filter maps into a digital filter in Impulse Invariant transformation. (4)

Or

(b) (i) Using Hanning window, design a filter with

− = − − 4 ≤ ≤ 4

0 + 4 ≤ ≤

Assume $N = 7$. (12)

Write a note on need and choice on windows. (4)

(a) Explain in detail the architectural features of a DSP processor.

(16) Or

Explain the addressing formats and functional modes of a DSP processor. (16)

Question Paper Code: 53129

B.E./B.Tech. DEGREE EXAMINATION, NOV/DEC 2010
Fifth Semester
Electrical and Electronics Engineering
EC 2314 — DIGITAL SIGNAL PROCESSING
(Regulation 2008)

Time : Three hours **Maximum : 100 marks**

Answer ALL questions

PART A — (10 × 2 = 20 marks)

What are even and odd signals?
What is known as Aliasing effect?
What is the relation between FT and Z transform?
Give any 2 properties of linear convolution?
5. Calculate DFT of $= 1,1,-2,-2$.
Differentiate between DIF and DIT.
What are the advantages of FIR filter?
Mention the significance of Chebyshev‘s approximation.
Define warping.
What is BSAR instruction? Give an example.

PART B — (5 × 16 = 80 marks)

11. (a) (i) What you mean by Nyquist rate? Give its significance. (6)
(ii) Explain the classifications of the discrete signals. (10)

Or

(b) (i) Explain in detail the quantization of digitals signals. (8)
Describe the different types of sampling methods can be used.(8)

(a) (i) Explain the properties of Z transform. (8)
Find the impulse response given by Differential equations.
$y(n) - 3y(n-1) - 4y(n-2) = x(n) + 2x(n-1)$. (8)

Or

(b) (i) Test the stability of given systems.
1) = cos(( )) 2) ( ) = (− − 2) 3) ( ) = ( ) (8)
(ii) Find the convolution of the following systems
. 1) $= -1,1,2,-2$, ↑ $= 0.5,1,-1,2,0.75$ ↑ 8

13. (a) An 8- point sequence is given by $= 2,2,2,2,1,1,1,1$ Compute 8 point DFT OF x(n) by radix DIT------FFT Method, Sketch the Magnitude and Phase. (16)

Or

(b) Determine the response of LTI system when the input sequence is $= -1,1,2,1,-1$ radix 2 DIF FFT. The impulse response is $= -1,1-,1,1$.(16)

(a) (i) Design a low pass filter using rectangular window by taking 1 samples of w(n) with cut off sequence of 1.2 randians/sec also draw the filter.(16)

Or

The specifications of defined low pass filter is : Using Hanning window, design a filter with

$0.8 \leq \;\leq 0.2;\; ;\; 0.32 \leq \;\leq 1.0; 0 \leq \;\leq 0.2$

Design Chebyshev‘s digital filter using bilinear transformations.(16)

(a) (i) Explain in detail the Von Neumann architecture with a neat diagram.(8)

What is MAC unit? Explain its functions?(8)

Or

(b) Explain in detail the architectural of TMS320C50 with a neat diagram. (16)

Question Paper Code: 21368

B.E./B.Tech. DEGREE EXAMINATION,MAY/JUN 2013

Fifth Semester

Electrical and Electronics Engineering

EC 2361/EC2314/EE502 — DIGITAL SIGNAL PROCESSING

(Regulation 2008)

Time : Three hours **Maximum : 100 marks**

Answer ALL questions

PART A — (10 × 2 = 20 marks)

Given continuous time signal $= 2\cos 500$. what is the nyquist rate and fundamental frequency of the signal?

Determine = whether a power signal or an energy signal?

Give the difference equation $= + 3 - 1 + 2 - 1$. determine the system function H(Z).

4. Find the stability of the system whose impulse response $= (\frac{1}{2})$

Find the discrete Fourier transform of

Draw the basic butterfly diagram for DIF algorithm.

Name two methods for digitiling the transfer function of an analog filter.

Mention the properties of Chebyshev‘s filter.

Mention one important features of hardware architecture?

What is the advantage of pipelining?

PART B — (5 × 16 = 80 marks)

11. (a) (i) Given $y(n) =$ ( )² ,Determine whether the system is linear ,time invariant ,memoryless and causal (8)

(ii) Determine whether the following is an energy signal or power signal. (8)

(1) 1= 6 cos( ).

2 = 3(0.5) ( ). Or

Starting from first priniciples, state and explain the sampling theorem both in time domain and frequency domain (16)

(a) (i) Find the Z transform and its ROC for the following discrete time signal

= (−1/5)+ 5(1/2)− − −1 . (8)

Evaluate the frequency response of the system described by system function

= ($\frac{1}{1-0.5^{-1}}$). (8)

Or

(b) Using the Z transform determine the response y(n) for n≥ 0 if

= $\frac{1}{2}$ y n − 1 + x n , x n = $\frac{1}{3}$ [n] u n ; y −1 = 1 (16)

13. (a) Find the output y(n) of a filter whose impulse response = 1,1,1 and the input signal = 3, −1,0,1,3,2,0,1,2,1 using overlap save method.(16)

Or

(b) Find the DFT of a sequence = 1,2,3,4,4,3,2,1, using DIT algorithm. (16)

(a) Design and realize a digital filter using bilinear transformations for the following specifications. (16)

Monotonic pass band and stop bamnd -3.01dB cut off at 0.5 radian magnitude down atleast 15dB at = 0.75 .

Or

(b) (i) Consider the causal linear time shift in variant filter with system function

= $\frac{1+0.875^{-1}}{1+0.2^{-1}+0.9^{-2}(1-0.75^{-1})}$. Draw the structure using a parallel interconnection of first and second order systems. (8)

(ii)Consider the following interconnection of a linear shift invariant system.where = ,2=−1

= 1 ≤ $\frac{\pi}{2}$; 0 $\frac{}{2}$ < ≤ (8)

Find the overall impulse response of the system.

15. (a) Explain the addressing formats of a DSP processor. (16)

Or

(b)Draw the functional block diagram of a DSP processor and explain. (16)

Question Paper Code: 10304

B.E./B.Tech. DEGREE EXAMINATION,MAY/JUN 2012
Fifth Semester
Electrical and Electronics Engineering
EC 2361/EC2314/EE502 — DIGITAL SIGNAL PROCESSING
(Regulation 2008)

Time : Three hours **Maximum : 100 marks**

Answer ALL questions

PART A — (10 × 2 = 20 marks)

What is meant by nyquist rate?
State and prove the time reversal property of FT.
3. Consider the signal = 1 − 1 ≤ ≤ 10 for all other values of n,
sketch the magnitude and phase spectrum.
4. Find the convolution of = 0,1,0,2 and h(n)= 2,0,1 .
Differentiate IIR and FIR filter.
Give relationship between DTFT and Z transformation. What is meant by quantization error?
State warping and give the necessity of prewarping.
Define the condition for stability of digital filter.
Define periodogram.
Define Gibbs phenomena.

PART B — (5 × 16 = 80 marks)

(a) Check the following systems are linear, causal, time in variant, stable, static.
y(n) = x(1/2n)
y(n) = sin(x(n))
y(n) = x(n)cos(x(n))
y(n) = x(-n+5)
(v) y(n) = x(n) + nx(n+2) (16)

Or

(b)Compute linear and circular convolution of the two sequence
x1(n) = 1,2,2,2 and x2(n) = 1,2,3,4 (16)

12. (a) (i) Determine the system function and the unit sample response of the system described by the difference equation y(n) +1/2y(n-1)=2x(n). (8)
(ii) Determine the step response of the system y(n)- (− 1)+x(n), -1< < 1,when the initial condition is y(-1) = 1. (8)

Or

(b) An filter system is described by the difference equation y(n) = x(n0+x(n-10).

(i) Compute and sketch its magnitude phase response

(ii)Determine its response to the input x(n) = cos(10) + 3sine(3 + 10).

(16)

(a) (i) Derive the computational equation for the 8-point FFT DIT. (8)

State and prove any five properties of DFT. (8)

Or

(b) Find the X(K) for given sequence = 1,2,3,4,1,2,3,4 . (16)

14. (a) For the analog transfer function H(s) = 2/(s+1)(s+3) determine H(z) using bilinear transformation. With T=0.1 sec. (16)

Or

(b) Design an ideal high pass filter with Hd( ) = 0≤/4 /4 ≤≤ π .Using Hamming window with N = 11.

.

(a)(i) Explain the addressing formats of a DSP processor. (8)

Explain in detail the architecture of the DSP processor. (8)

Or

(b)(i) Explain the functional modes present in the DSP processor. (8)

(ii) Explain about pipelining in DSP. (8)

Question Paper Code: 66288

B.E./B.Tech. DEGREE EXAMINATION,NOV/DEC 2011
Fifth Semester
Electrical and Electronics Engineering
EC 2361/EC2314/EE502 — DIGITAL SIGNAL PROCESSING
(Regulation 2008)

Time : Three hours **Maximum : 100 marks**

Answer ALL questions

PART A — (10 × 2 = 20 marks)

What is a linear time invariant system?
. State sampling theorem.
What is meant by ROC in z-transforms?
Write the commutative and distributive properties of convolution.
Draw the basic butterfly diagram for Radix 2 DITFFT.
6.Write the DTFT for (a) $x(n) = a^n u(n)$ (b) $x(n) = 4\delta(n) = 3\delta(n-1)$.
What is meant by linear phase response of a filter?
Compare bilinear transformation and Impulse invariant method of IIR filter design.
Give the special features of DSP processors.
What is pipelining?

PART B — (5 × 16 = 80 marks)

11. (a) (i) Discuss whether the following are energy or power signals

(1) $x(n) = \frac{3}{2}^N u(n)$

(2) $x(n) = Ae^{jw_0 n}$. (10)

(ii) Explain the concept of quantization. (6)

Or

Check whether following are linear, time invariant, causal and stable.

(i) $y(n) = x(n) + n\,x(n+1)$. (8)

(ii) $y(n) = \cos x(n)$. (8)

12. (a) (i) Obtain the linear convolution of

$x(n) = \{3,2,1,2\}$ $h(n) = \{1,2,1,2\}$. (6)

↑

(ii) A discrete time system is described by the following equation :

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

Determine its impulse response. (10)

Or

(b) (i) Obtain the discrete Fourier series coefficients of $x(n) = \cos w_0 n$. (4)

Determine x(n) for the given x(2) with ROC

$|z| > 2$

$|z| < 2$

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

. (12)

13. (a) (i) Explain 8 pt DIFFFT algorithm with signal flow diagram. (10)

(ii) Compute the DFT of $x(n) = \{1, 1, 0, 0\}$. (6)

Or

(b) (i) Describe the following properties of DFT.

1. Time reversal.
2. Circular convolution.

(ii) Obtain the circular convolution of

$x_1(n)] = \{1, 2, 2, 1\}$ $x_2(n) = \{1, 2, 3, 1\}$. (6)

(a) Design a butterworth filter using the Impulse invariance method for the following specifications.

$0.8 \leq |H(e^{jw})| \leq 1 \quad 0 \leq W \leq 0.2\pi$

$|H(e^{jw})| \leq 0.2 \quad 0.6\pi \leq w \leq \pi$

Or

Design a filter with desired frequency response

$$Hd(e^{jw}) = e^{-j3w}$$

for

$$= 0 \quad \text{for} \quad -\frac{3\pi}{4} \le w \le \frac{3\pi}{4}$$

$$\frac{3\pi}{4} \le |w| \le \pi$$

Use Hanning window for $N = 7$.

15.(a) Explain the addressing modes of a DSP processor.

Or

(b)Describe the Architectural details of a DSP processor.