Vidisha Narang
161011423

Digital Signal Processing
EED

IEEE Signal Processing Magazines
↳ Implementation Difficulties
→ Describing concepts in a better way

Quizzes 10%

Lab 10% → completion of weekly lab
15% → lab exam
project - 10%

Midsem - 20%
Endsem - 35%

Midsem Topics covered in class

→ DC Blocker

→ Introduction
→ System Modelling
→ Convolution
→ FIR
→ IIR
→ Block Convolution
→ Deconvolution
→ ~~[illegible]~~
→ Difference Equation
→ Cascaded and Parallel Forms
→ Similarity Measure → Auto correlation
Cross correlation
→ Z Transform
→ Eigen values and Eigen vectors
→ Frequency Response (DTFT)

Engineering ⟷ Sets

Signals ⟷ Sequence] ⟶ Discrete

Systems ⟷ Functions

Stegnography

Voice (Speech) — Numbers — MP3 players — Numbers

Image, Video (Light Intensity variation) — "

Text (numbers) — "

Sensors o/p — "

Different types of signals converting into numbers will help in transmitting them together at the same time.

Numbers ⟶ Integers set ⟶ I

Real set R

Complex C

Real ⟶ 1 (Write as 1.0 in decimals)

It is integers not decimal

Therefore signal is a sequence of numbers.

Order is important

default ⟶ Indexed number sequence ⟶ Integers (Discrete signal) $x[n]$

⟶ Real number (continuous signal) $x(t)$

We will use Integer indexed sequence.

System ⟶ manipulator

input $x[n]$ ⟶ [System] ⟶ o/p $y[n]$

System is uniquely mapping using any manipulation

Mapping

Sample by Sample mapping

$x(0) \longrightarrow y(0)$

$x(1) \longrightarrow y(1)$

$$\begin{bmatrix} x(0) \\ x(1) \\ x(2) \end{bmatrix} \longrightarrow \begin{bmatrix} y(0) \\ y(1) \\ y(2) \end{bmatrix}$$

Block by Block

$x(n) \;—\; \boxed{\quad\quad} \;—\; y(n)$

i/p   o/p

$y(n) = T(\,x(n))$

↳ Functional

One to one → reversible, you can get input from output

one to many → unidirectional

We will be looking at one to one, reversible mapping and linear systems

$T \begin{cases} \to \text{linear} \\ \to \text{non linear} \end{cases}$

Input Sequence $(x(0), x(1), x(2))$

Output $(y(0), y(1), y(2), y(3))$

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix}_{4\times1} \qquad \begin{bmatrix} x(0) \\ x(1) \\ x(2) \end{bmatrix}_{3\times1}$$

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix}_{4\times1} = \begin{bmatrix} & & \\ & & \\ & & \\ & & \end{bmatrix}_{4\times3} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \end{bmatrix}_{3\times1}$$

$\frac{10\times30}{10+30} = \frac{300}{40}$

57

$4\times1 \rightarrow$ output

$3\times1 \rightarrow$ input

Hence system $\rightarrow$ 4×3 matrix

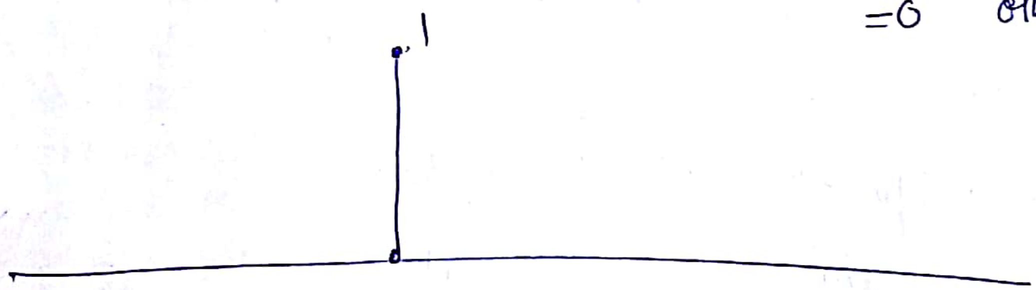
If we change the input to the system, we may or may not get the same system response,

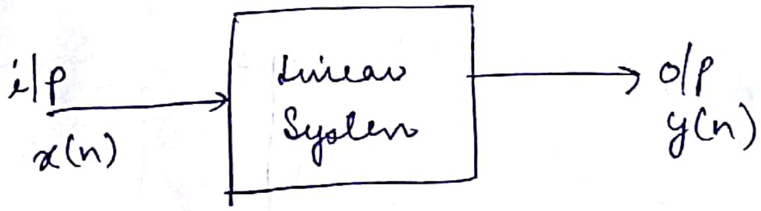
System Modelling $\rightarrow$ If we find a unique method to represent linear system which gives a is constant for all types of input then it is called system modelling.

Shift operation should exist,

Converting a system into numbers.

$$\delta(n) = \begin{cases} 1 & n=0 \\ 0 & \text{otherwise} \end{cases}$$



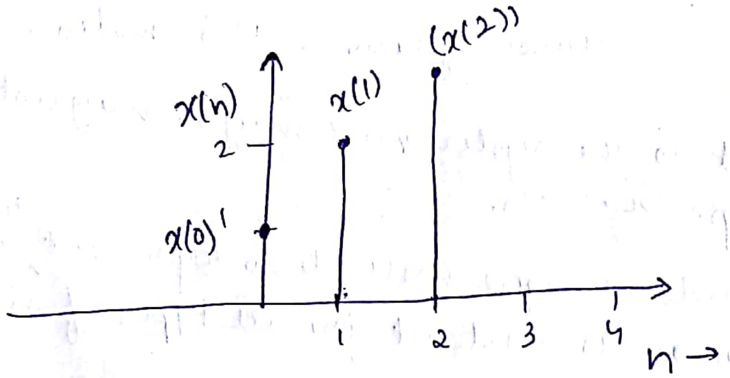


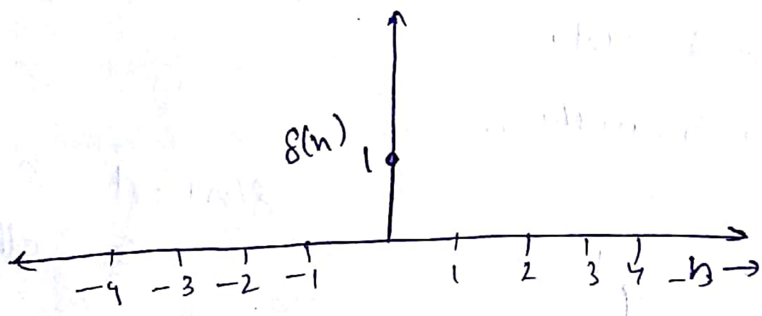
We want express input into some basic signals.

Ex:

$$\delta(n) = \begin{cases} 1 & n=0 \\ 0 & \text{otherwise} \end{cases}$$

$$x(n) = \{ \overset{(1)}{x(0)}, \overset{(2)}{x(1)}, \overset{(3)}{x(2)} \}$$

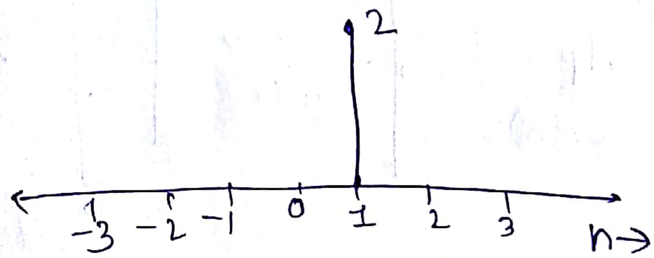




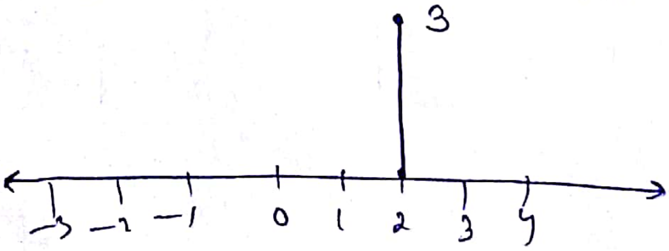
Split x(n) into three sub sequences.

1. δ(n)

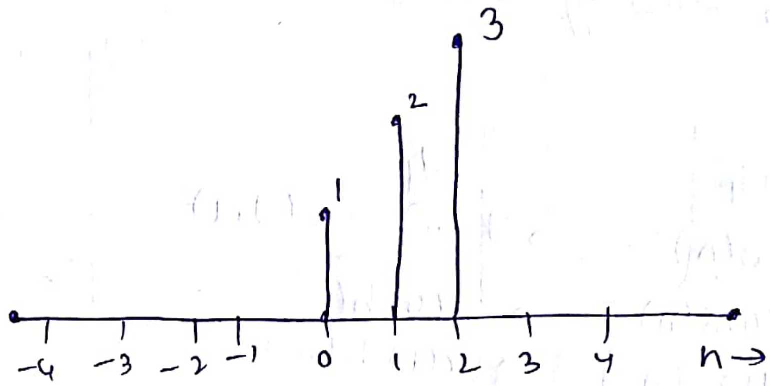
2. δ(n−1)



3. δ(n−2)



$1.\delta(n) + 2.\delta(n-1) + 3.\delta(n-2) = x(n)$



$x(n) = x(0)\,\delta(n) + x(1)\,\delta(n-1) + x(2)\,\delta(n-2)$

↳ weighted sum of delta sequence and its shifts

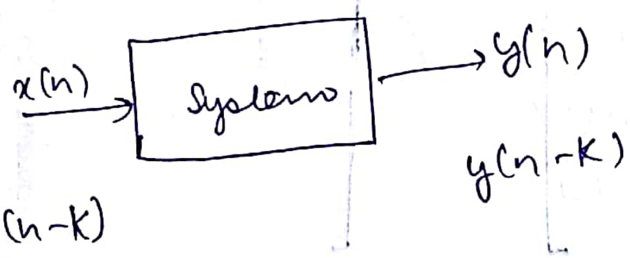
$$\boxed{x(n) = \sum_{k=0}^{k=\infty} x(K)\,\delta(n-K)}$$

$$\boxed{x(n) = \sum_{k=-\infty}^{\infty} x(K)\,\delta(n-K)}$$

Any input signal is expressible in terms of delta sequence and its shift in range $[-\infty, \infty)$.

For a linear system, the system response will be same for both individual parts of inputs and entire sequence of input i.e it satisfies superposition and homogenity.

Time invariant → Any system which does not vary with time, For ex if we delay the input, output is also delayed.



Combination of Linearity + Time invariance → LTI systems

Linear Time Invariant

$$
\begin{array}{ccc}
\text{I/p} & & \text{O/p} \\
\delta(n) & \longrightarrow & h(n) \quad (\text{Let}) \\
x(0)\delta(n) & \longrightarrow & x(0)\,h(n) \\
x(1)\delta(n-1) & \longrightarrow & x(1)\,h(n-1) \\
\vdots & & \vdots
\end{array}
$$

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

Advantage: We need to find $h(n)$ once, then we can find all the outputs. i.e the system is unique.

To see this, we give specific input → $\delta(n)$. height is 1

$\delta(n) \to h(n)$ delta or impulse (discrete) response.

↳ response

dirac (continuous) response

height is undefined

impulse response → Linear Time Invariant represent model

$$
\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} \; \\ \; \\ \; \\ \; \end{bmatrix} \begin{bmatrix} x(1) \\ x(2) \\ x(3) \end{bmatrix}
$$

$$x(n) \longrightarrow \boxed{\begin{array}{c}\text{LTI} \\ h(n)\end{array}} \longrightarrow y(n)$$

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

If $h(n) = (h(0))$ $\rightarrow$ infinite possibilities

~~Well causale~~

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)\, h(n-k)$$ $\rightarrow$ linear convolution sum

Causality of system

↳ depends on present and past values, not on future.

Real time $\rightarrow$ instantaneous
For a given input we get a output.

For a causal systems.

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

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

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

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

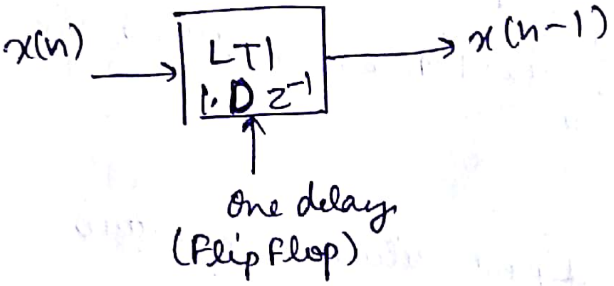
$$= \underbrace{h(0)x(n) + h(1)x(n-1) + h(2)x(n-2)}$$ ↳ weighted sum of input and its shifts

$$h(n) = (h(0), h(1), h(2))$$

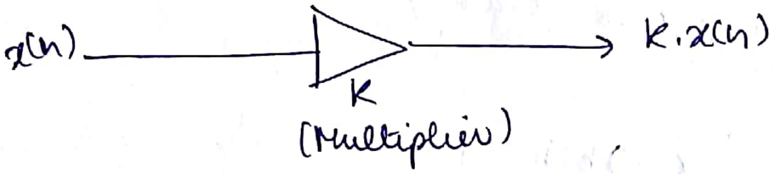
Impulse response is scaling the different shifts of input

Hardware requirement: ① Shift operator
② Scale operator
③ Adder/subtractor

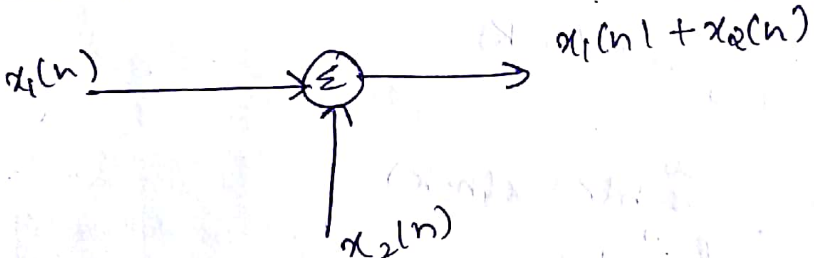
Shift operator



Scale operator:

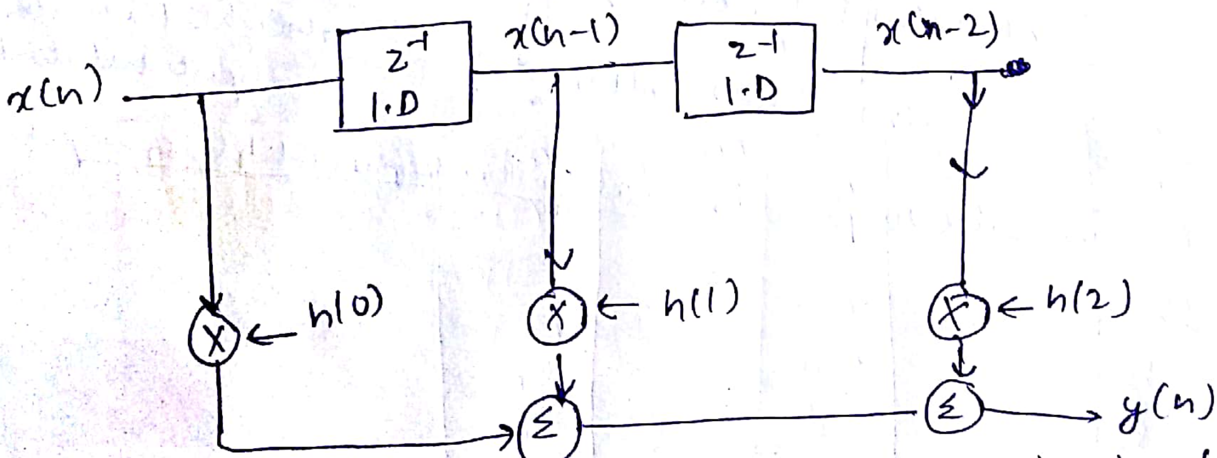


Adder/Subtractor:



(FIR)

Eg. $y(n) = h(0)x(n)+h(1)\,x(n)+h(2)\,x(n-2)$
$h(0)x(n)+h(1)x(n-1)+h(2)x(n-2)$



FIR System $\underbrace{(h'(0),\ h'(1),\ h'(2))}_{\text{characterisation}}$

If $h'(n) = (h'(0), h'(1), h'(2))$

we just need to change the three multipliers

Any LTI system can be of two types

1. Finite length Impulse Response (FIR System)
2. Infinite length Impulse Response (IIR System)

Transient output (~~before~~ the outputs we see before the final output)

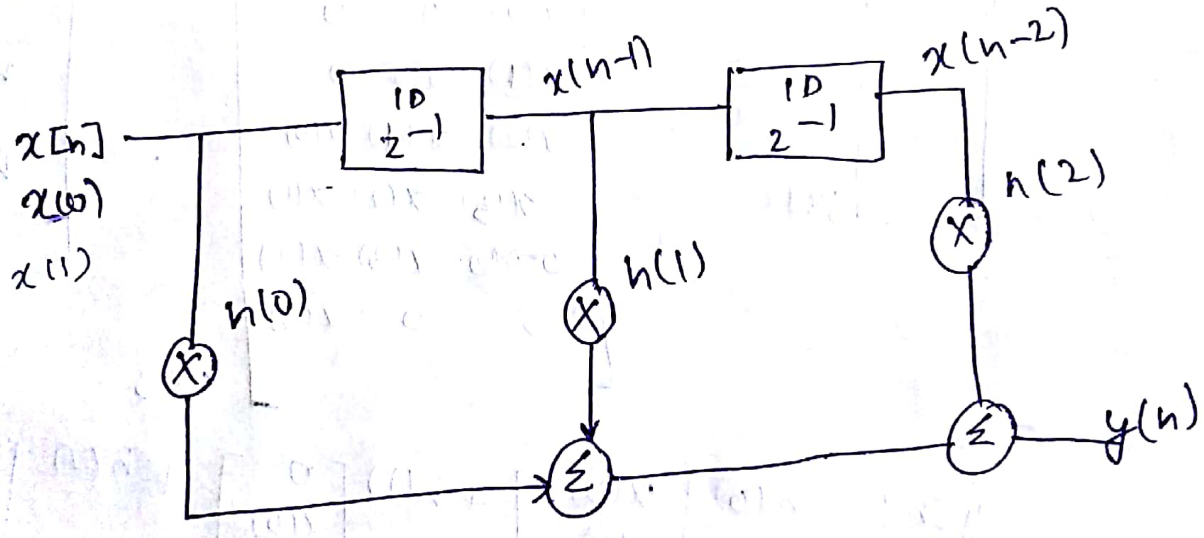
For Ex-1

Till the three inputs are given, we won't be able to see the complete output.

Finite Impulse Response :

FIR $\quad h(n) = [h(0), h(1), h(2)]$

IIR $\quad x(n) = [x(0), x(1), x(2), x(3)]$



Input or Transient
$$\left[\begin{array}{l} y(0) = x(0)h(0) \\ y(1) = x(1)h(0) + x(0)h(1) + 0 \end{array}\right.$$

No. of delays = on transient length

Steady State Output
$$y(2) = x(2)h(0) + x(1)h(1) + x(0)h(2)$$
$$y(3) = x(3)h(0) + x(2)h(1) + x(1)h(2)$$

Off Transient
$$y(4) = x(3)h(1) + x(2)h(2)$$
$$y(5) = 0 + x(3)h(2)$$
$$y(6) = 0$$

Off transient length (2) = No. of delays

No. of delays = length of Impulse Response −

1. Output length of linear convolution sum is equal to input length + impulse response length − 1.
2. Identifying input on Transient output, steady state output, ~~Input~~ off transient output..

$$y[n] = \sum_{k=0}^{\infty} h(k)x(n-k)$$
$$= h(0)x(n) + h(1)x(n-1) + h(2)x(n-2) + \alpha x(n-3)$$

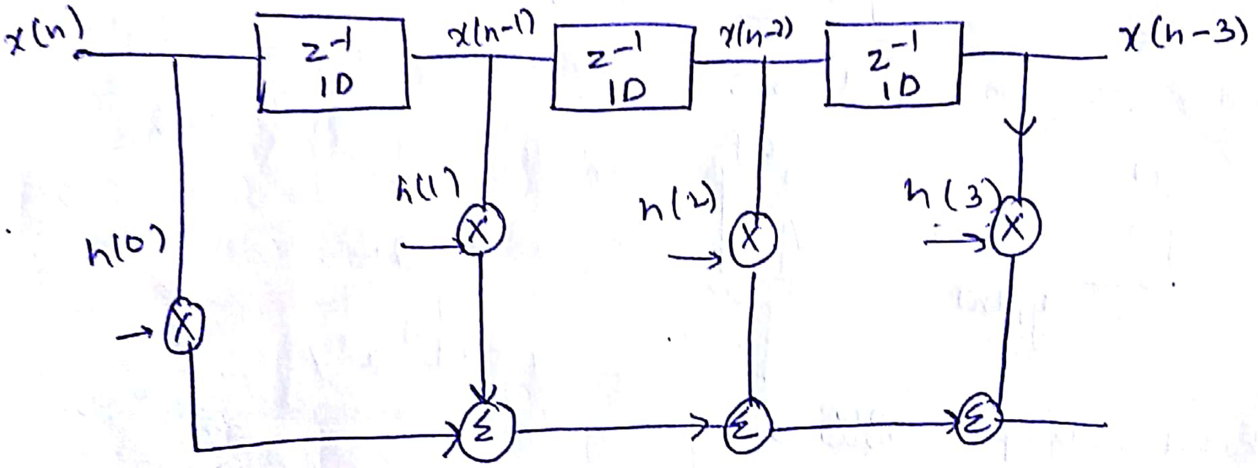
$$\underset{B}{\begin{bmatrix} \\ \\ \\ \\ \\ \\ \end{bmatrix}}_{6\times 1} = \underset{A}{\begin{array}{ccc} x(n) & x(n-1) & x(n-2) \\ \hline \end{array}} \begin{bmatrix} x(0) & 0 & 0 \\ x(1) & x(0) & 0 \\ x(2) & x(1) & x(0) \\ x(3) & x(2) & x(1) \\ 0 & x(3) & x(2) \\ 0 & 0 & x(3) \end{bmatrix}_{6\times 3} \underset{X}{\begin{bmatrix} h(0) \\ h(1) \\ h(2) \end{bmatrix}}_{3\times 1}$$

$$y[n] = h(0)\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ 0 \\ 0 \end{bmatrix} + h(1)\begin{bmatrix} 0 \\ x(0) \\ x(1) \\ x(2) \\ x(3) \\ 0 \end{bmatrix} + h(2)\begin{bmatrix} 0 \\ 0 \\ x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \\ y(4) \\ y(5) \end{bmatrix} = \begin{bmatrix} h(0) & 0 & 0 & 0 \\ h(1) & h(0) & 0 & 0 \\ h(2) & h(1) & h(0) & 0 \\ 0 & h(2) & h(1) & h(0) \\ 0 & 0 & h(2) & h(1) \\ 0 & 0 & 0 & h(2) \end{bmatrix}_{6\times 4} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}_{4\times 1}$$

$h(n) = (h(0), h(1), h(2), h(3))$ FIR

$x(n) = (x(0), x(1))$



$\boxed{\text{No steady state}}$ as no output has effect of all components. Every output has only two inputs and corresponding response.

Minimum Input length required is $\boxed{\text{On transient } +1}$ so that we can get steady state.

$h(n) = (h(0), h(1), h(2), h(3))$

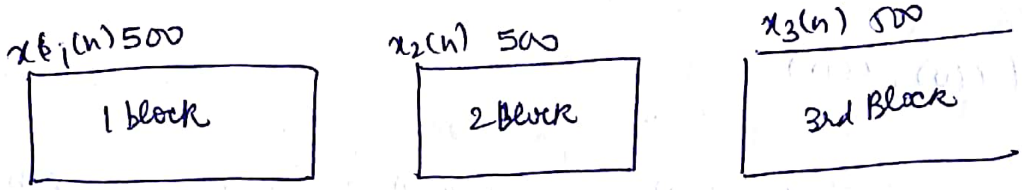
$x(n) = (x(0), x(1) \ldots\ldots\ldots)$

Real time processing → Output before completion of input

Off line processing / stored → Output after completion of input

If we have infinite length input, divide it into blocks of length equal to tolerance limit

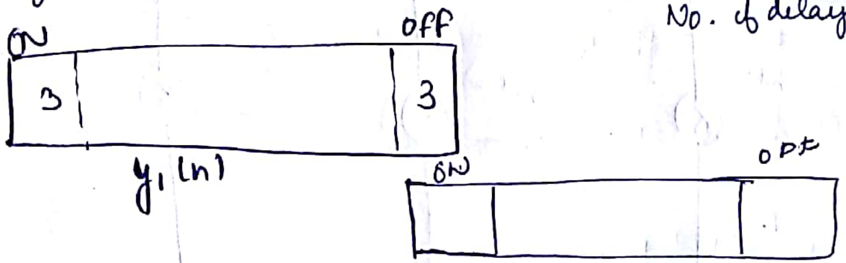
Let us, say, we divide input into length of 500



Output length $500 + 4 - 1 = 503$

length of impulse response $= 4$

No. of delays $= 4 - 1 = 3$



First 3 samples → ON transient

last 3 samples – OFF transient

Since we forced input to stop after 500 samples there is an extra OFF transient for $(y_1(n))$ and ON transient for $(y_2(n))$.

If. OFF transient first terms that goes away is $h(0)$

ON transient First terms to come is $h(0)$

If we add these two we will not see any break in output and it will behave as a stored processing

This strategy is known as $\boxed{\text{over lap add method} \rightarrow \text{Block Convolution}}$

Explore the other possibilities.

Block Convolutions.

$$h(n) = (h(0),\ h(1),\ h(2)) = h(0)\delta(n) + h(1)\delta(n-1) + h(2)\delta(n-2)$$

$$x(n) = (x(0),\ x(1),\ x(2),\ x(3))$$

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

$$Y = \overset{H}{\left[\qquad\right]}\overset{x}{\left[\quad\right]}$$

If $y(n)$ and $h(n)$ is given, finding $x(n)$, → Source Identification Problem ~~system~~
For ex. speech, image processing.

Given $y(n)$ $x(n)$, identify $h(n)$ → System Identification Problem

Given $x(n)h(n)$, identify $y(n)$ → Blind Convolution Problem.

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

$Y = HX$ → linear convolution

for above question

$$\overset{Y}{\left[\quad\right]}_{6\times1} = \overset{H}{\left[\qquad\right]}_{6\times4}\overset{X}{\left[\quad\right]}_{4\times1}$$

$$\left[\quad\right]_{6\times1} = \overset{X}{\left[\qquad\right]}_{6\times3}\overset{H}{\left[\quad\right]}_{3\times1}$$

Least Square Solution : (Linear Deconvolution) Finding X from Y and H

$4\times 6 \quad 6\times 4$

$$Y = HX$$

multiplying by H^T

$$H^TY = H^THX$$

assuming H is a real number matrix

If H is a complex number matrix then multiply by $H^H = (H^*)^T$ ↳ (conjugate of H)^T

$$\underset{4\times 6\ \ 6\times 1}{H^T Y} = \underset{4\times 6\ \ 6\times 4}{H^T H}\ \underset{4\times 1}{X}$$

$$(H^TY)_{4\times 1} = (H^TH)_{4\times 4} X_{4\times 1}$$

$$(H^TH)^{-1}(H^TY) = (H^TH)^{-1}(H^TH)X$$

determinant of $|H^TH| \neq 0$

$$\boxed{\hat{X} = (H^TH)^{-1}H^TY}$$

Least Square solution

Now if x(n) y(n) are given,

$$\underset{6\times 1}{Y} = \overset{X}{\begin{bmatrix} & \\ & \end{bmatrix}}_{6\times 3} \overset{H}{\begin{bmatrix} \\ \end{bmatrix}}_{3\times 1}$$

$$(X^TY) = \overset{X^T}{\begin{bmatrix} & \\ & \end{bmatrix}}_{3\times 6} \overset{X}{\begin{bmatrix} & \\ & \end{bmatrix}}_{6\times 3} \overset{H}{\begin{bmatrix} \\ \end{bmatrix}}_{3\times 1}$$

$$h = (X^TX)^{-1}(X^TY)$$

Preambles → known information sent to the receiver.

Types of LTI System: Finite Impulse Response (FIR)
Infinite Impulse Response (IIR)

Infinite Impulse Response (IIR)

$h(n) = (h(0), \ldots \ldots)$ → gives infinite degree of freedom

→ infinite delays
→ No steady state
→ always in ON Transient State

if the coefficients of impulse response are not independent
For ex
$h(n) = (1, a, a^2, a^3 \ldots \ldots)$

→ GP should converge
→ only one delay required

independent terms of impulse response implies more degree of freedom

$h(n) = a^n \quad n \geq 0$ → causal system
$0 \quad$ otherwise

Sequence can converge → to finite value
→ to infinite value (no use)

$|a|$ should be between 0 to 1
a can be 0
not 1

$0 \leq |a| < 1$

$(0.9)^2 \quad (0.1)(0.1)$
$(0.81)^2 \quad 0.01$
$(0.3)^3 \quad (0.001)$
$0.729 \quad (1.0001)$
$(10)^{-4}$

If $|a| = 0.9 \quad |a| = 0.1$
they both will converge to 0,
0.1 will Reach faster to 0 for less n
0.9 will reach 0 for more n

Impulse Response will change according to the value of a.

→ Such Representation gives variable finite impulse response.

→

$h_{effective}$

We can stop impulse response after it reaches 1% of its initial value.

$$a^{n_{eff}} = 0.01$$

→ it will help ~~....~~ us to find steady state by choosing appropriate value of n effective

2. $h(n) = a^n + b^n$ is the following sequence possible

both are convergent

convergence will depend on larger of a and b

→ a and b needs not be dependent.

→ 2 degree of freedom

→ we can take any sum of GP but it should be of finite length

→ Poles of a system

For ex: for $a^n + b^n$

Poles $= a, b$

dominated poles determines the length of system.

→ $h(n) = a^n \quad n \geq 0$

$= 0$

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

$h(-1) = 0$

$h(n) = ah(n-1) + \delta(n)$

↳ Recursive Formula

$h(0) = 1$

$\delta(n) = 1 \quad n = 0$

$\quad\quad\;\; 0 \quad \text{otherwise}$

$h(1) = a + 0$

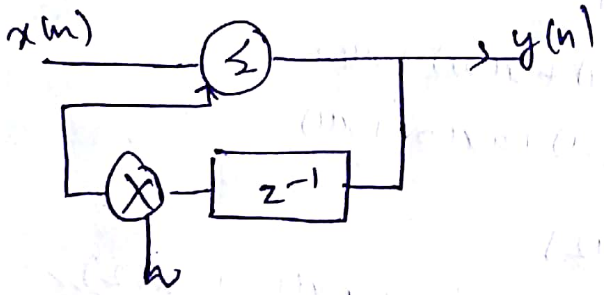
$h(2) = a(h(1)) + 0$

$h(2) = a^2$

If $x(n) = \delta(n)$

$y(n) = h(n)$

→ Choose appropriate value of a



effective length of h can be changed by changing a.

Hardware Implementation reduced to 1 delay

Steady state $= n^{eff} - 1$

IIR

→ dependent ~~sequence~~ series

→ convergent

→ recursive

Accumulation effect:

$|a| = 1$

↳ Creation of Accumulation

(Overflow Condition)

due to

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

recursions ⟶ Difference Equation ⟶ (Ist Order Constant Coefficient Linear Difference Equation)

→ as series is not convergent.

FIR → Non Recursive

IIR → Recursive Form

If $y(-1) \neq 0$

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

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

$$= a(a y(-1) + x(0)) + x(1)$$

$$= a^2 y(-1) + a x(0) + x(1)$$

$$y(2) = a\,y(1) + x(2)$$

$$= a^3 y(-1) + a^2 x(0) + a x(1) + x(2)$$

$$y(n) = a^{n+1} y(-1) + a^n x(0) + a^{n-1} x(1) + \dots + x(n)$$

$$y(n) = \underbrace{a^{n+1} y(-1)}_{\uparrow} + \underbrace{\Sigma a^n x(0) + a^{n-1} x(1) + \dots + x(n)}_{\downarrow \text{ Inputs}}$$

Initial Condition of Difference Equation ↑ Zero Input Response
(Output even when input is zero)
(Natural Response of systems)

When initial conditions are zero, the response of input at that time is zero state response (Forced system response)

If initial conditions are zero, we can characterise recursive response and nonrecursive form in a same way.

So in previous eq^n put $y(-1)=0$

$$y(n) = \sum a^k x(n-k)$$

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

$$h(n) = a^k u(n)$$

Z - Transform

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

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

$z \rightarrow$ complex

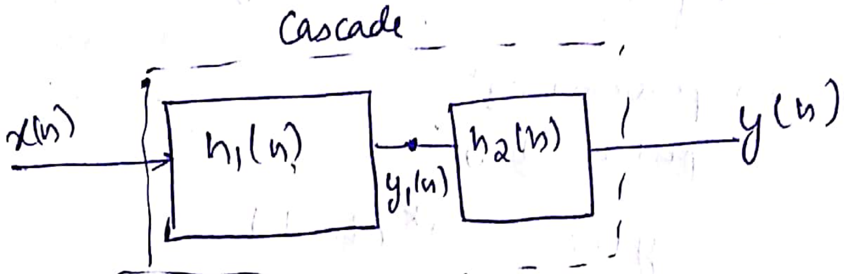
Z has two variable
real imaginary

So $X(z) \rightarrow$ should be represented in 3-D graph but to avoid that situation we draw two separate graphs
→ 1. to represent magnitude
→ 2. to represent phase.

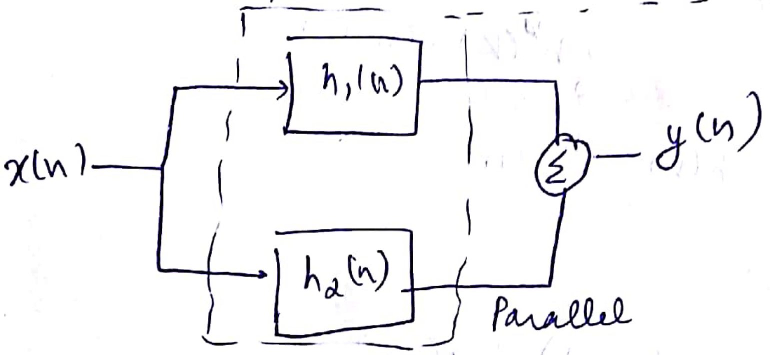
$y = a^n + b^n$

$a\,y(n-1) + b\,y(n-2)$

Cascaded Systems



Parallel forms



Effective Impulse response

$y_1(n) = x(n) * h_1(n)$

$y(n) = y_1(n) * h_2(n)$

$= (x(n) * h_1(n)) * h_2(n)$

$= x(n) * (h_1(n) * h_2(n))$

$= x(n) * h_3(n)$

$\boxed{h_3(n) = h_1(n) * h_2(n)}$

Effective Impulse Response

If systems are cascaded, their effective impulse responses will be convoluted.

$h_N(n) = h_1(n) * h_2(n) * \cdots * h_N(n)$

Vice Versa, we can split inverse into n sums

For parallel form

$y_1(n) = x(n) \circledast h_1(n)$

$y_2(n) = x(n) * h_2(n)$

$y(n) = y_1(n) + y_2(n)$
$= x(n) * h_1(n) + x(n) * h_2(n)$

$y(n) = x(n) * (h_1(n) + h_2(n))$

$\boxed{h_3(n) = h_1(n) + h_2(n)}$

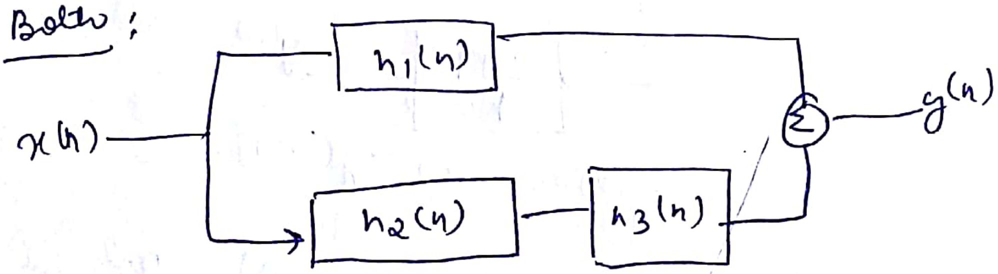
$\uparrow$

Effective Impulse Response.

$y(n) = x(n) * [h_1(n) + h_2(n) + \cdots + h_N(n)]$

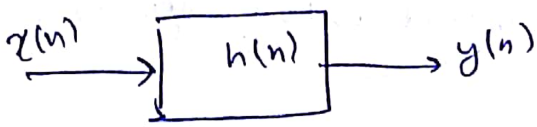
In parallel form, the impulse responses of different systems are added to form a single effective Impulse Response.

Combination of Both :



$h_{eff} = h_1(n) + (h_2(n) * h_3(n))$

Similarity Measure → Correlation function



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

Systems cannot tolerate ambiguity.

$R_{xy}(l) = x(l) * y(-l)$ → Linear convolution

Cross correlation

$r_{xy}(l) = x(l) * y(-l)$ → Machine learning

Auto correlation

$r_{xx}(l) = x(l) * x(-l)$ → extensively used in communication

Relation b/w

$r_{yy}(l)$ $r_{xx}(l)$

Correlation → Auto Correlation function
Correlation → Cross correlation function

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

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

$r_{yx}(l) = y(n) * x(-n)$ $n=l$

$r_{xx}(l) = x(n) * x(-n)$ $n=l$

$r_{yy}(l) = y(n) * y(-n)$ $n=l$

$r_{yx} = y(n) * x(-n) \quad n = l$

$r_{yx} = x(n) * h(n) * x(-n) \quad n = l$

$r_{yx}(l) = h(n) * x(n) * x(-n) \quad n = l$

$$\boxed{r_{yx}(l) = h(l) * r_{xx}(l)}$$

$r_{yy}(l) = y(n) * y(-n)$

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

$= \cancel{x * x}$

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

$$\boxed{r_{yy}(l) = r_{xx}(l) * r_{hh}(l)} \rightarrow \text{Imp}$$

(used in Random process)

Normalised Auto correlation function $= \dfrac{r_{xx}(l)}{|r_{xx}(0)|}$

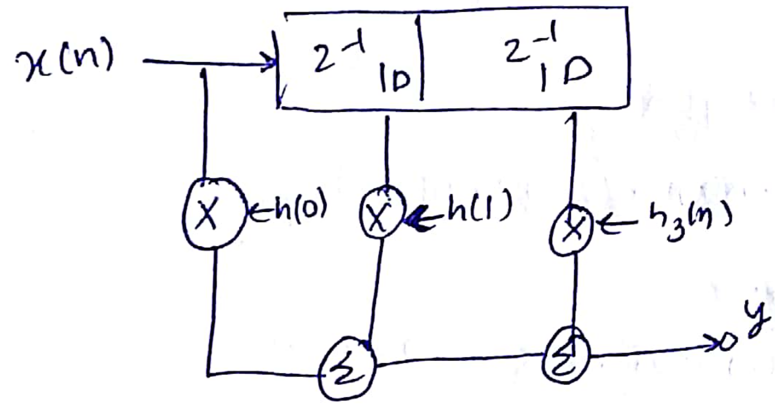
Normalised Cross correlation function

$$= \frac{r_{xy}(l)}{\sqrt{|r_{xy}(0) \cdot r_{yx}(0)|}}$$

This will ensure that function lies b/w -1 and 1.

$r_{xx}(0) \Rightarrow$ first element, source of energy of signal,

Solutions of Linear Constant Coefficient Difference equation.



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

↑

Second Order Non Recursive Linear constant coefficient Difference equation,

$$y = f(x(n),\ x(n-1),\ x(n-2))$$

$$y(n) = f(x(n), x(n-1) \ldots, y(n-1)\ y(n-2))$$

↳ Recursive

$$\boxed{\begin{array}{ll} y(n) = a\,x(n-1) + x(n) & \rightarrow \text{Non Recursive} \\ y(n) = a\,y(n-1) + x(n) & \rightarrow \text{Recursive} \end{array}}$$

Z transform

$z = re^{j\theta}$ $\quad r, \theta$

$s = \sigma + j\omega$

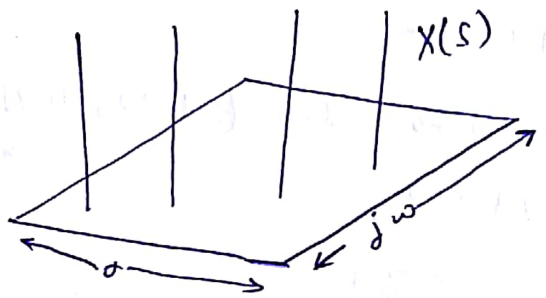
$-\infty \leq \sigma \leq \infty$

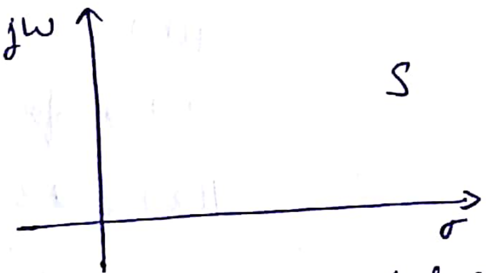
$-\infty \leq \omega \leq \infty$

$S = \sigma + j\omega$

$X(s) = X(\sigma + j\omega)$

↳ Requires 3 dimension to represent





Cartesian Representation

We can have a mapping b/w the two as they are representing the same plane.

Domain of r and θ

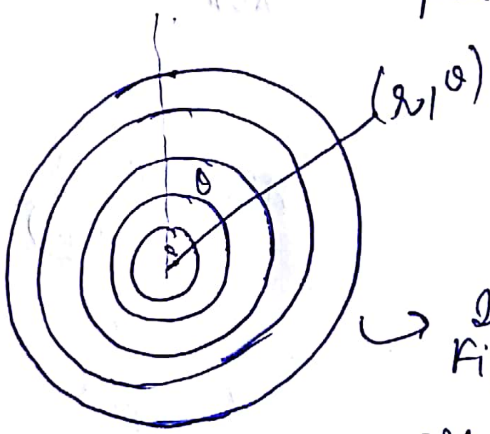
$0 \leq r \leq \infty$

$0 \leq \theta \leq 2\pi$

$-\pi \leq \theta \leq \pi$

$r = 0$

$\theta = 2\pi$



↳ It cannot have fixed references

→ If it is fixed it means its capabilities are limited

Scanned by CamScanner

$$Y(z) = \sum_{n=-\infty}^{\infty} y(n) z^{-n}$$

→ weighted sum of z^{-n}

$$x(n) \xleftrightarrow{Z} X(z)$$

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

$$x(n) * y(n) \longleftrightarrow X(z) Y(z)$$

Linear constant coefficient Difference eqⁿ → FIR (Recursive) / IIR (Non Recursive)

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

Z transform on both sides ($X(z)$, $Y(z)$ should exist)

$$Y(z) = a z^{-1} Y(z) + X(z)$$

$$Y(z) - a z^{-1} Y(z) = X(z)$$

$$Y(z) = \frac{X(z)}{(1 - a z^{-1})}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - a z^{-1}} = \frac{1}{1 - \frac{a}{z}} = \frac{z}{z-a}$$

Rational Polynomial

$$H(z) = \frac{B(z)}{A(z)}$$

$B(z)$ → If we put roots of this polynomial over $H(z)$ will converge to zero

$A(z)$ → If we put roots of this polynomial over $H(z)$ will go to infinity.

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + x(n)$$

$$Y(z) = a_1 z^{-1} Y(z^{-1}) + a_2 z^{-2} Y(z^{-1}) + X(z)$$

$$\frac{Y(z)}{X(z)} = \frac{1}{1 - a_1 z^{-1} - a_2 z^{-2}}$$

Any recursive LCDE will result in ~~as~~ rational polynomial.

Non recursive

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

$$\frac{Y(z)}{X(z)} = \frac{1 + a z^{-1}}{1}$$

For non recursive $A(z) = 1$

only one polynomial

so only one root

Roots of $B(z)$

$H(z) = 0 \Rightarrow$ Zeros of the Function $H(z)$

Roots of $A(z)$

$H(z) = \infty \rightarrow$ Poles of the function $H(z)$

$$H(z) = \frac{1}{1 - a^{-1} z}$$

$H(z$

Z- ~~tra~~transform

$$z = re^{j\theta}$$

$$h(n) = a^n u(n)$$

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

$$= \sum_{n=0}^{\infty} a^n z^{-n}$$

$$= \sum_{n=0}^{\infty} (az^{-1})^n$$

$$= \cancel{\frac{az^{-1}}{1-az^{-1}}}$$

$$|az^{-1}| < 1$$

$$|z| > |a|$$

$$= \frac{1}{1-az^{-1}}$$

Region of Convergence (ROC)

System function

$$H(z) = \frac{B(z)}{A(z)}$$

$$= \frac{(z-z_1)(z-z_2)\cdots(z-z_n)}{(z-z_1')\cdots\cdots(z-z_n')}$$

→ Splitting into partial fraction

→ finding roots.

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

$$\frac{1}{(z-1)(z-2)} = \frac{A}{(z-1)} + \frac{B}{(z-2)} = \frac{A(z-2)+B(z-1)}{(z-1)(z-2)}$$

$$A(z-2) + B(z-1) = 1$$

$$z = 2 \qquad z = 1$$

$$B = 1 \qquad A = -1$$

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

$$= \frac{1}{1-z} + \frac{1}{2}\left(\frac{-1}{1-z/2}\right)$$

$a = 1$

$=$

$= \cancel{y(n) + -\frac{1}{2}\left(\frac{1}{2}\right)}$

eigen function of LTI system → $x(n) = z^n$

$$x(n) \longrightarrow \boxed{LTI}_{h(n)} \longrightarrow y(n)$$

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

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

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

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

→ let us say it converges to K

$=$

$$\underbrace{y(n) = x^n (K)}$$

↓ Output is scalar multiple of input

If we change system response we will get different scaling factor,

can be compared with → $\boxed{A x = \lambda x}$

A → Transformation matrix

λ → eigen value

x → eigenvector

$A_{3\times3}$ Matrix λ_1 λ_2 λ_3 are eigen values

v_1 v_2 v_3 are ~~corresponding~~ corresponding eigen vectors

$$Av_1 = \lambda_1 v_1$$

$$Av_2 = \lambda_2 v_2$$

$$Av_3 = \lambda_3 v_3$$

We can use eigen values and vectors to get A

$$A = \underbrace{\begin{bmatrix} v_1 & v_2 & v_3 \\ | & | & | \end{bmatrix}}_{3\times3}^{U} \underbrace{\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}}_{3\times3}^{\Lambda} \underbrace{\begin{bmatrix} - v_1 - \\ - v_2 - \\ - v_3 - \end{bmatrix}}_{3\times3}^{U^T}$$

$$A = U \Lambda U^T$$

We can get infinite A matrices because eigen vectors/vectors can have infinite values.

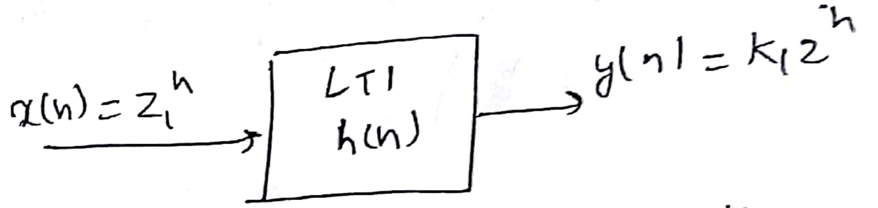
Three possibilities of eigen values 1. changing eigen vector const 2. constant changing 3. both changing

$$A = \underbrace{v_1 \lambda_1 v_1'}_{3\times3} + \underbrace{v_2 \lambda_2 v_2^T}_{3\times3} + \underbrace{v_3 \lambda_3 v_3^T}_{3\times3}$$

$\downarrow$ Eigen Transformations

If we remove any one of eigen vector and eigenvalue, it will be impossible to get back A.

Completeness of Transformation → All eigen vectors and value should be there.



If $K_1 = \sum_{k=-\infty}^{\infty} h(k) z_1^{-k}$ is finite then

$$y(n) = K_1 z^n$$

We need all such K and z_1 whose sum is finite, to transform into A.

$$\left.\begin{matrix} \cdots & K_2 & K_3 \\ & \vdots & \vdots \\ \cdots & K_2 & K_n \end{matrix}\right\}$$

If we do not have completeness, we cannot built that system exp

$$\left.\begin{matrix} (z_1^n, z_2^n \ldots z_k^n) \\ (K_1, K_2 \ldots K_z') \end{matrix}\right] \to$$ gives freedom of designing ed

$$z_1^n = r_1^n e^{j\omega_1}$$

$$|z_1| = 1$$

$$x(n) = 1 \cdot \underbrace{e^{j\theta_1 n}}_{z_1^n}$$

$$\sum_{k=-\infty}^{\infty} h(k) e^{-j\theta_1 k} \to \text{Finite}, \ \text{Infinite}$$

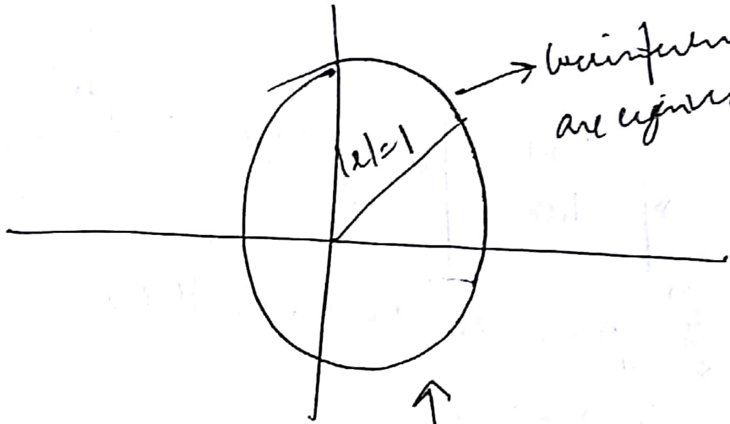
$$y(n) = e^{j\theta_1 n} \underbrace{\sum_{k=-\infty}^{\infty} h(k) e^{-j\theta_1 k}}$$

[illegible]

[illegible]

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

BIBO ↳ Bounded Input Bounded Output



→ circumference of circle are eigenvectors for x.

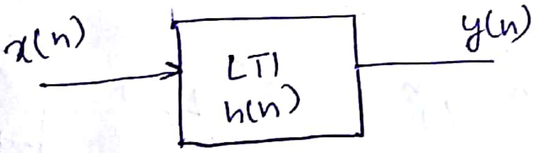
Frequency Response

→ For any LTI systems if $x(n) = z_1^n - k_1$ for all these we will get scaled version of impulse response

$z_2^n - k_2$

⋮

$z_K^n - k_K$ ↳ degenerative type



$$z = r e^{j\theta}$$

$$x = e^{j\theta_1 n}$$

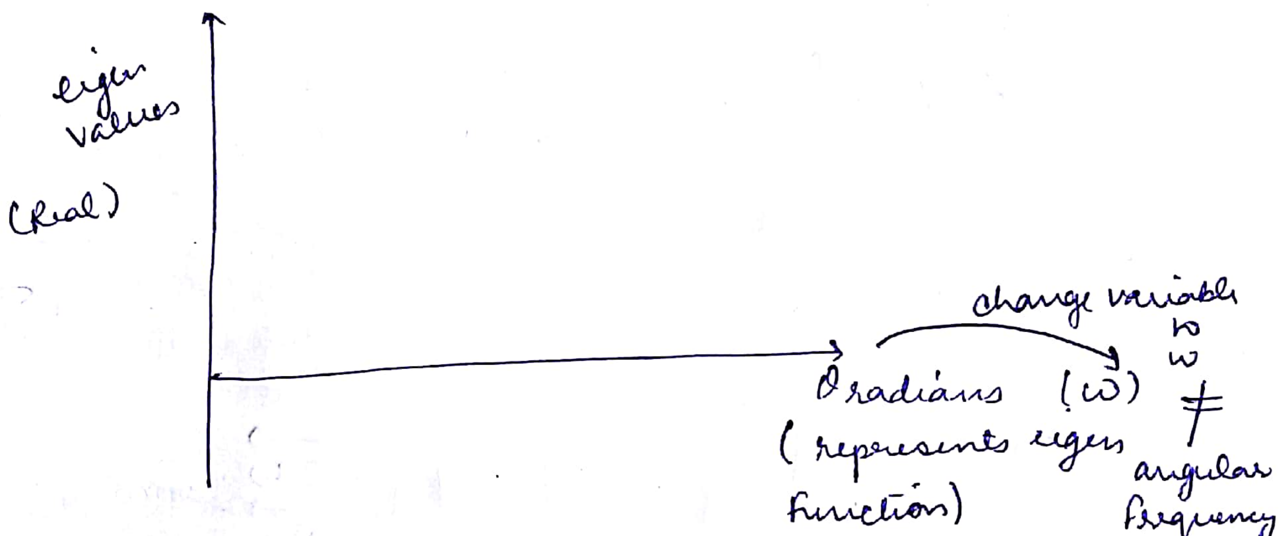
There are multiple sets of eigen functions possible of any LTI systems. Ex → z^n and $e^{j\theta_1 n}$ both gives scaled version of LTI systems.

If we get eigen values and eigen vectors we can build transformation matrise from here.

If we specific ROC of z_1^n . . . . z_k^n . . . . sequence then we ~~cannot~~ can find the eigen vectors / functions.

$$x(n) = e^{j\theta_1 n}$$

They are not convergent continous repetitive nature



If eigen values real → one plane is enough.

complex → two graphs → one magnitude
→ one phase

radian per second

$$-\infty \leq \Omega \leq \infty$$

Scanned by CamScanner

Frequency Representation: Discrete Time Fourier Transform

eigen values →

→ Eigen Function

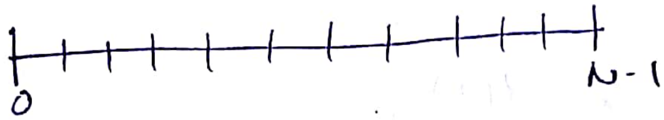
Limitation → Time Resolution as $-\infty \leq n \leq \infty$

they are non converging sequency with length 1.

Temporal Resolution does not exist.

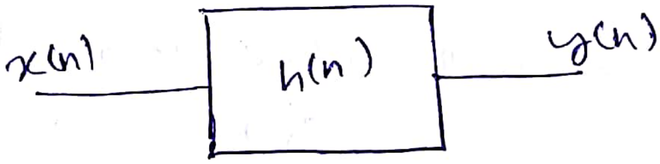
Finite length Impulse

Sequence →



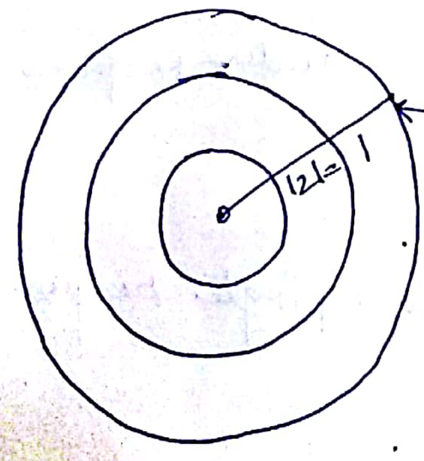
Simplest eigen functions for these
→ N dimensional representations
→ max only N functions

Frequency Response of an LTI System



$x(n) = e^{j\omega_i n}$

If we choose this input we will scaled functions of input as output.



This circle denotes all the values of ω (non decaying function)

Frequency Response is the set eigen vectors on $|z| = 1$

Example:

$$H(z) = 1 - z^{-1}$$

$$\frac{Y(z)}{X(z)} = 1 - z^{-1}$$

$$Y(z) = (1-z^{-1})\, X(z)$$
$$= X(z) - X(z) z^{-1}$$

Inverse Z transform on both sides

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

Non recursive as output is not a function of its delayed version.

ROC is entire Z plane ~~with exp~~ except $z=0$

We are interested on the values that lie of $|z|=1$ circle.
i.e Subset of ROC
for that we take $z = e^{j\omega}$.

$$H(z = e^{j\omega})$$

$$H(z=e^{j\omega}) = 1 - e^{-j\omega} = 2j e^{-j\omega/2} \frac{[e^{j\omega/2} - e^{-j\omega/2}]}{2j} = 2j e^{\omega/2 j} \sin\omega/2$$

$$|H(z=e^{j\omega})| = |2 \sin\omega/2|$$

↳ Magnitude Response

$$\angle H(z=e^{j\omega}) = \frac{\pi}{2} - \frac{\omega}{2}$$

↳ Phase | Angle response

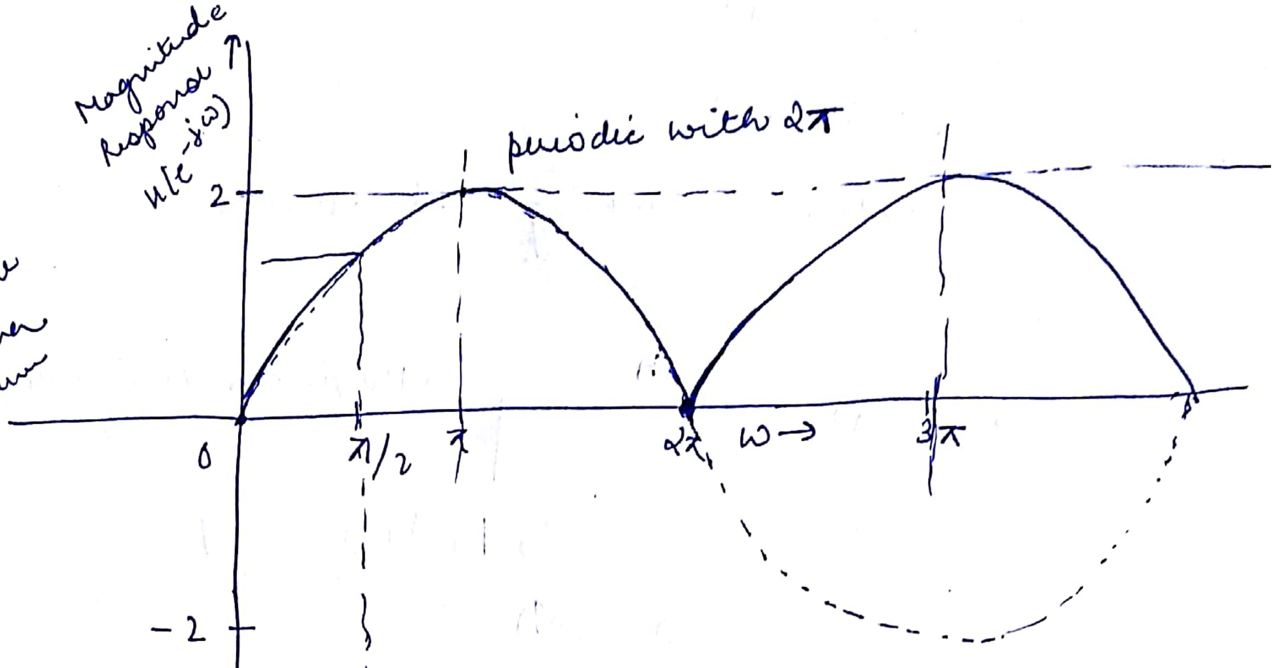
$$2j\, e^{\omega/2 j} \sin\omega/2$$

$$\cancel{2j\,\cos} = 1 - e^{-j\omega}$$

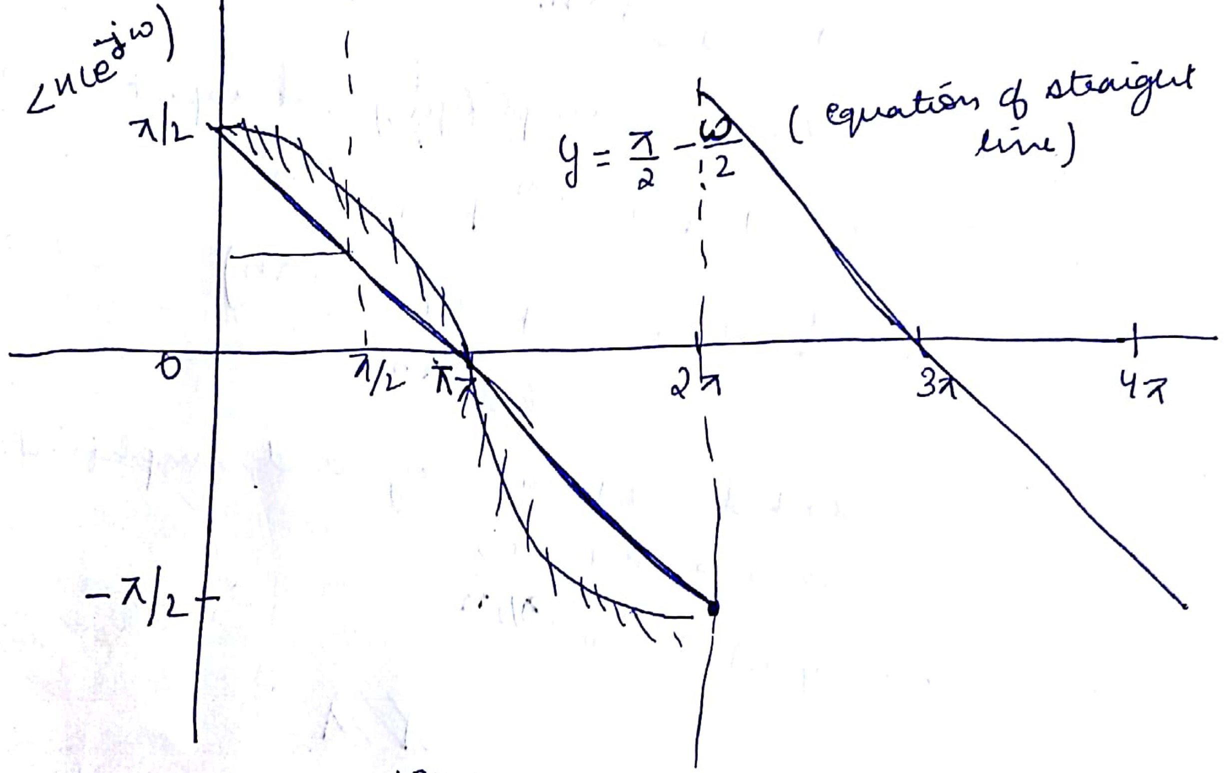
$$j = e^{j\pi/2}$$

$$2\, e^{j\pi/2}\, e^{-j\omega/2} \sin\frac{\omega}{2}$$

Frequency Response = eigenvalue plot for $x(n) = e^{j\omega n}$



* It has only one maxima
* 2 minimum



$x(n) = e^{j0n}$
↳ sequence of 1's

$x(n) = \{\cdots 1, \cdots\}$

eigenvalue for $\omega = 0 = 0$

$y(n) = 0 \cdot e^{j\pi/2} = 0$

$\longrightarrow$ $H(n) = \delta(n) - \delta(n-1)$

$\longrightarrow$ $x(n) = e^{j0n}$

$y(n) = 0$

System is blocking that sequence

$\longrightarrow$ $x(n) = e^{j\pi n}$ → Maximum variation

$x(n) = \{ \ldots -1, 1, -1, 1, \ldots \}$

$y(n) = 2 e^{j0} \cdot e^{j\pi n}$

$y(n) = 2 e^{j\pi n}$

System is amplifying the input

$\longrightarrow$ $x_3(n) = 2 e^{j0n} + 3 e^{j\pi n}$

$y_3(n) = 0 + 3(2 e^{j\pi n})$

$= 6 e^{j\pi n}$

Blocked one frequency and amplified the other

This is a filter

$\longrightarrow$ $x_4(n) = e^{j\pi/2 n}$

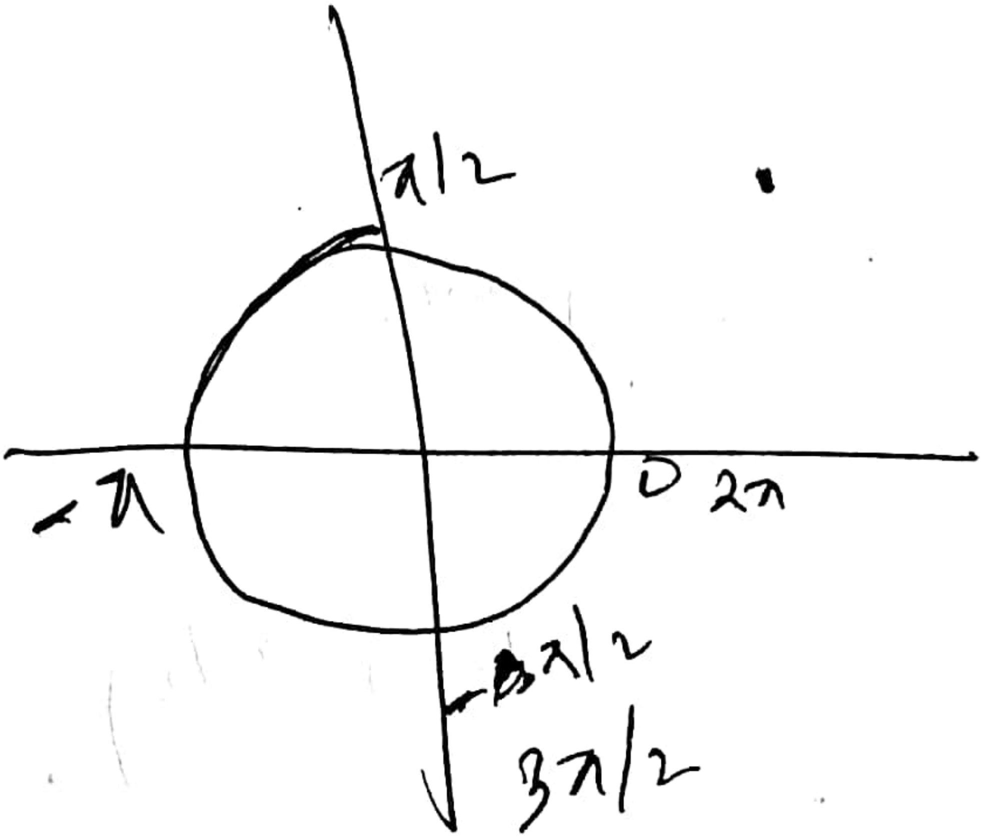
N/A

=

* No two eigen values are same.
* For every value of $0 \le \omega \le 2\pi$ the eigen value is different.
* eigen functions remain same only eigen values change.

DC BLOCKER

↑ Input is not varying


π/2
-π
0
2π
-3π/2
3π/2

12/09/18

(Causal System)

$H(z) = 1 - z^{-m}$ m is an integer

$z = e^{j\omega}$

~~$H(e^{j\omega})$~~ =

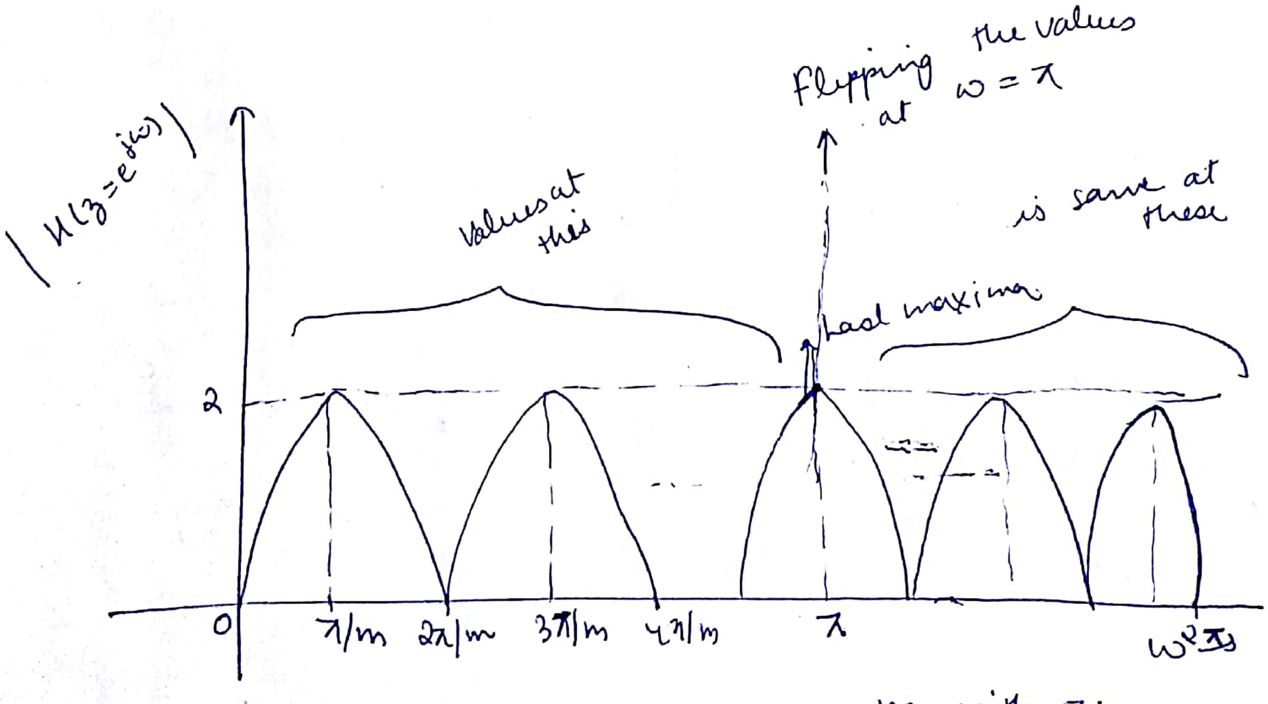
$$H(z=e^{j\omega}) = 1 - (e^{-j\omega})^m$$

$$= e^{-j\omega m/2}\frac{\left[e^{j\omega m/2} - e^{-j\omega m/2}\right]}{2j} \times 2j$$

$$= 2j\, e^{-j\omega m/2}\left(\sin\frac{\omega m}{2}\right)$$

$$= e^{j(\pi/2 - j\omega m/2)} \cdot 2\sin\left(\frac{\omega m}{2}\right)$$

$$|H(z=e^{j\omega})| = |2\sin(\omega m/2)|$$



for maximum equate $\frac{\omega m}{2}$ with $\pi/2$

$$\frac{\omega m}{2} = \frac{\pi}{2}$$

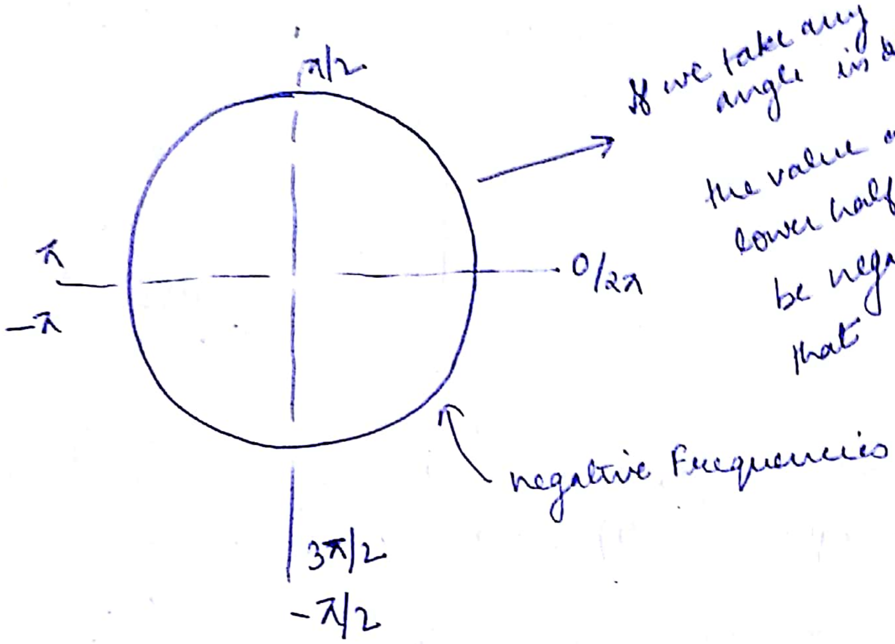
$$\boxed{\omega = \frac{\pi}{m}}$$

Scanned by CamScanner

Minimum value

$\frac{\omega m}{2} = 2\pi$ or $\frac{\omega m}{2} = \pi$

$\omega = \frac{4\pi}{m}$. $\omega = \frac{2\pi}{m}$



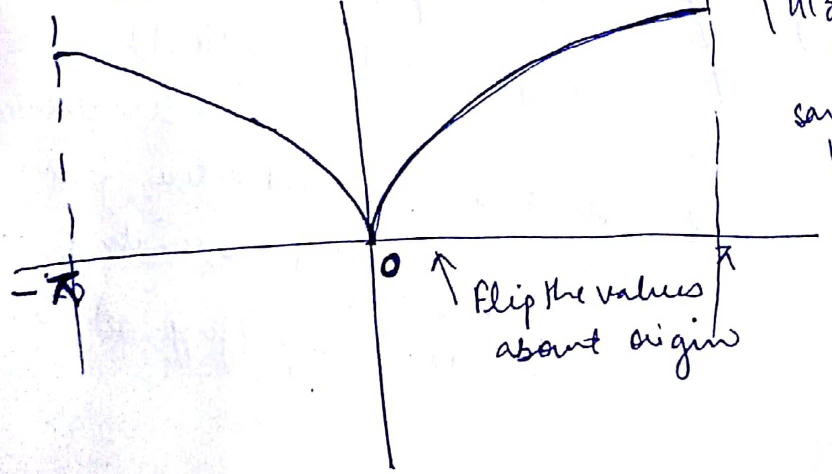
If we take any angle in upper half the value at the lower half will be negative of that

The maximum unique values available $= m/2$

Max. Frequency of Representation $= \pi$
$(2\pi = 0)$

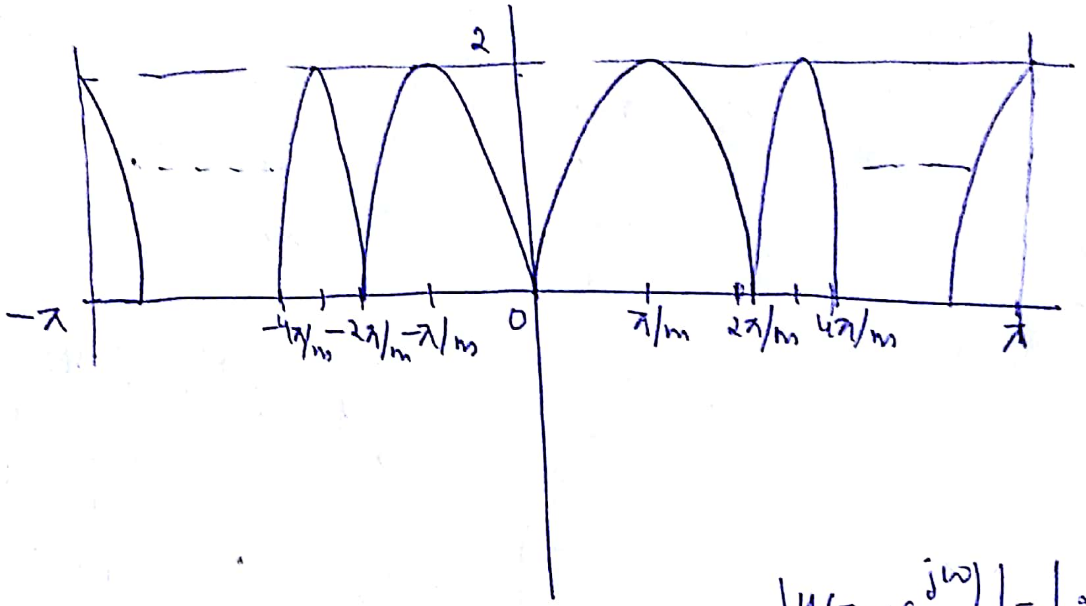
Smallest Frequency $= 0$

Since we are writing magnitude value at $\frac{\pi}{2}$ will be equal $\frac{3\pi}{2}$



For $|H(z)| = |2 \sin\frac{\omega}{2}|$

same for the previous graph



$-\pi - \frac{\pi}{m}$

$\pi\left(\frac{m-1}{m}\right)$

$$\left|H(z=e^{j\omega})\right| = \left|2\sin\left(\omega m/2\right)\right|$$

In first example, we could only eliminate only one frequency (0). Now, in second we could eliminate

$0, 2\pi/m, 4\pi/m, 6\pi/m$ ---

Same goes for max. values

$\pi/m, 3\pi/m, 5\pi/m$ ---

Which means all even harmonics of π/m from 0 to π are eliminated

all odd harmonics of π/m are amplified from 0 to π.

$y(n) = x(n) - x(n-m)$ → only ~~each~~ one substraction is required

last example we had

$y(n) = x(n) - x(n-1)$

↳ successive differences

Subtracting the current value from m^{th} sample

so this is capable of eliminating all even harmonics not only 0

same for amplifying odd harmonics

$H(z) = 1 - z^{-1} \longrightarrow$ coefficient is 1

↳ Infinite eigen value b/w 0 to π

but degree of freedom is 1 (only one 0)

$1 - z^{-m}$

↳ mth degree of polynomial

Find the magnitude response of the following system

$$H(z) = 1 + z^{-1} + z^{-2} + \dots + z^{-(m-1)}$$ where m is an integer

$$= \frac{1-z^{-m}}{1-z^{-1}}$$

Plot $|H(e^{j\omega})|$ Vs ω

$$H(z=e^{j\omega}) = \cancel{E+}\, e^{0j\omega} + e^{-j\omega} + e^{-2j\omega} + \dots + e^{-j\omega(m-1)}$$

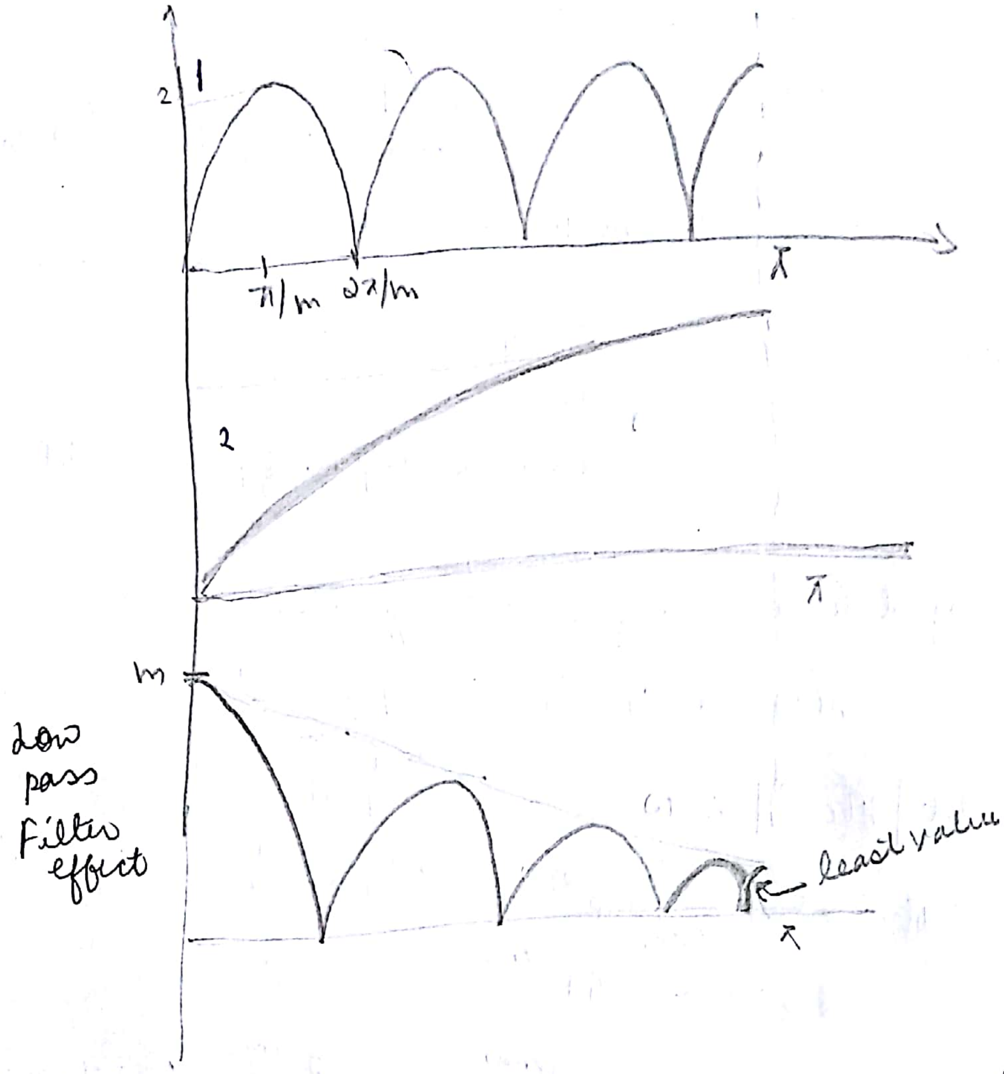
Infinite series G.P

$$H(z=e^{j\omega}) = \frac{1-e^{-jm\omega}}{1-e^{-j\omega}} = \frac{\cancel{2j}\, e^{-j\frac{\omega m}{2}} (\cancel{2}\sin(m\omega/2))}{\cancel{2j}\, e^{-j\omega/2}\, \cancel{2}\sin(\omega/2)}$$

$$|H(z=e^{j\omega})| = \left| \frac{\cancel{2}\sin(m\omega/2)}{2\sin(\omega/2)} \right|$$

$$\angle H(e^{j\omega}) = \cancel{\tan^{-1}(\ldots)}$$

$$= -\frac{\omega}{2}(m-1)$$



For ①/② at 0, the value of ①/② is not defined

till ~~π/3 = w/2~~

$$\sin\frac{w}{2} = \frac{1}{2}$$

$$\sin\frac{w}{2} = \sin\frac{\pi}{6}$$

$$\frac{w}{2} = \frac{\pi}{6}$$

because $w = \pi/3$ increases
after $w = \pi/3$ decreases

$$\boxed{w = \frac{\pi}{3}}$$

0 to $\pi/3$ → amplifies
$\pi/3$ to π → decreases
} → creates a low pass Filter

Using L Hospital's Rule

$$|H(e^{j0})| = \frac{m\omega/2}{\omega/2} = m$$

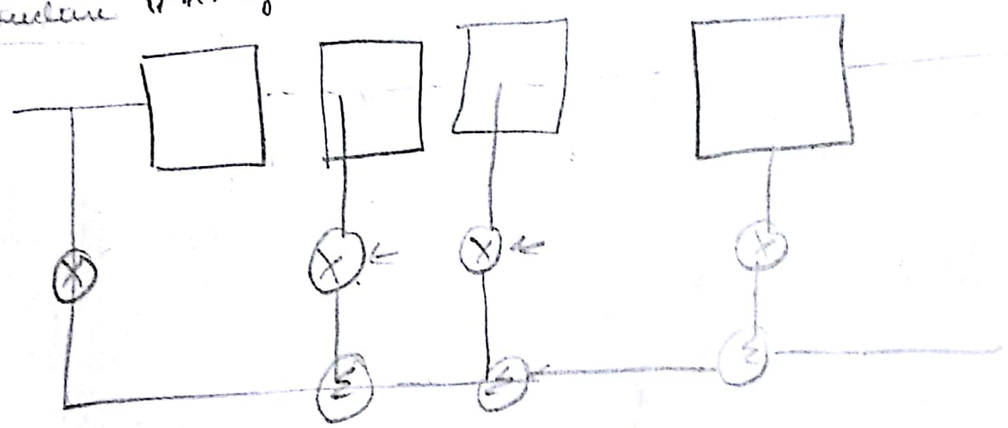
$$Y(z) = (1+z^{-1}+z^{-2}+ \dots \dots + z^{-(m-1)})\, X(z)$$

$$y(n) = x(n) + x(n-1) + \dots + x(n-m+1)$$

We are picking up all the previous m samples to get the output

FIR

FIR structure (FIR) System



Till $(m-1)^{th}$ → ON transient

m the sample - steady state

Moving Sum filter → Frequency Domain

↓

Low Pass Filter

$$H(z) = 1+z^{-1}+z^{-2}+ \dots + z^{-(m-1)}$$

$$= \frac{1-z^{-m}}{1-z^{-1}}$$

$$\frac{Y(z)}{X(z)} = \frac{1-z^{-m}}{1-z^{-1}} = Y(z)(1-z^{-1}) = X(z)(1-z^{-m})$$

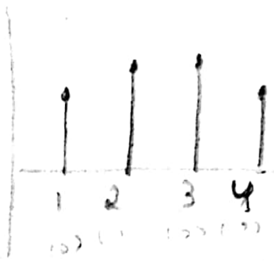
$$Y(z)\,(1-z^{-1}) = (1-z^{-m})\,X(z)$$

$$y(n) - y(n-1) = x(n) - x(n-m)$$

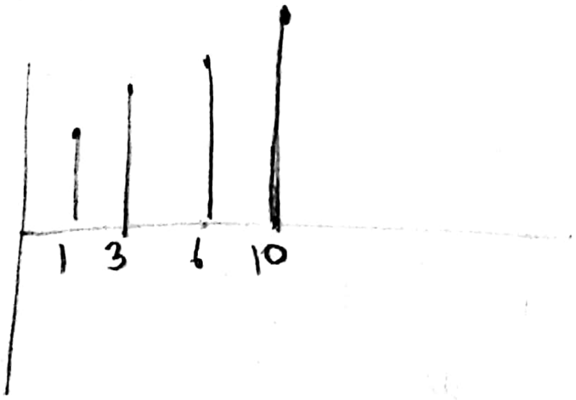
$$\boxed{y(n) = y(n-1) + x(n) - x(n-m)}$$ → Code for program

↓ IIR System

An FIR system is expressible in IIR system



— x(n-



We represented all the points from 0 to 2π in terms of polynomial or we took a polynomial and represented it within an finite range of 0 to π. (the ~~that~~ graph repeats after this interval)

$$|H(e^{j\omega})| = |1 - e^{-j\omega}| \rightarrow \text{first degree}$$

$$H(e^{j\omega}) = 1 - e^{-j\omega m} \rightarrow m\text{th degree polynomial}$$

We can represent an uncountable set of numbers (polynomials) by ~~a~~ countable set using Frequency representation

Polynomial can be represented using → coefficients / → roots

$$H(e^{j\omega}) = 1 - e^{-j\omega m} \quad (m+1) \text{ coefficients}$$

different coefficients

coefficients = 1, (m-1) zeroes , -1

$$H(e^{j\omega}) = 1 + e^{-j\omega} + e^{-2j\omega} + \dots + e^{-j(m-1)\omega}$$

no non zero coefficients

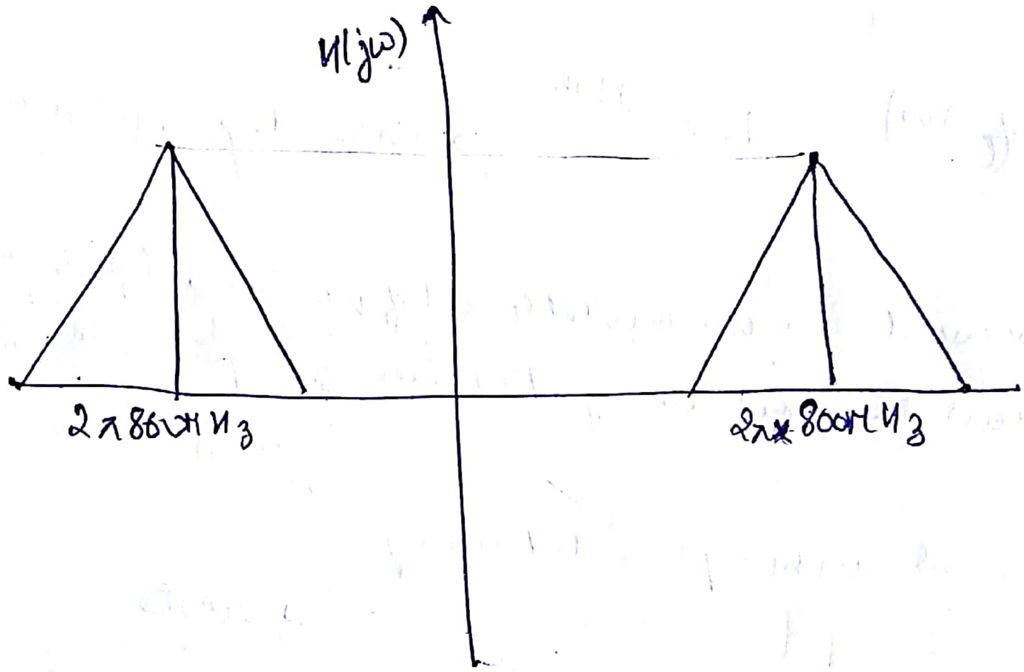
Eigenvalues → $H(e^{j\omega})$

↳ Alternative characterisation other than impulse response $h(n)$.

$H(z)$ → system function

Certain systems used in real life have representations in transform domain.

If we want to transmit at a carrier frequency of 800MHz then maximum should be at the Frequency not at zero ;



degree of polynomial = roots of the system ,

Build a System for the above with 10 degree polynomial

↳ (10+1) coefficients

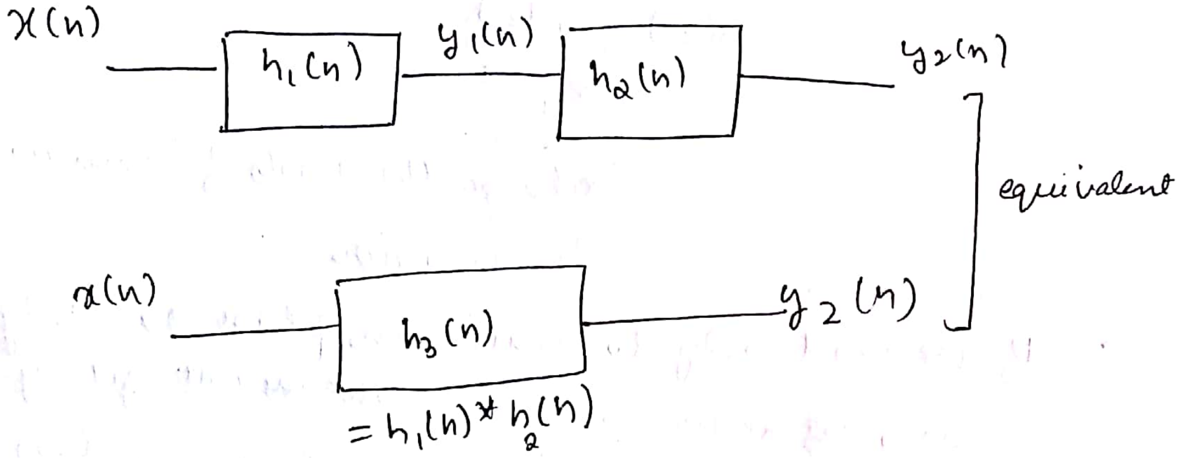
10 roots

Disadvantages of taking 100 degree (large degree polynomial) polynomial

→ more delay ~~because~~ ~~due~~

transient time increases

Two ways to represent → polynomials
→ sample function



Is it possible ~~(......)~~ to get $y_2(n)$ in some of form $\hat{x}(n)$

$$y_2(n) = \tilde{x}(n)$$

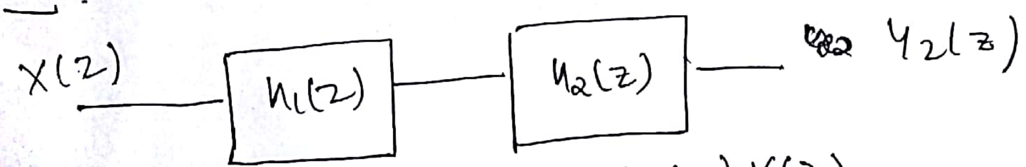
$h_2(n)$ should nullify the effect of $h_1(n)$
which means $h_3(n)$ should be some scaled function of del function.

This is called invertibility.

We can deconvolute the output to get the input back.

For this we should define a new impulse response that nullifies the effect of actual impulse response.

In Z Domain:



$$Y_2'(z) = H_1(z) H_2(z) X(z)$$

$$Y_2(e^{j\omega}) = H_1(e^{j\omega}) H_2(e^{j\omega}) X(e^{j\omega})$$

To get $Y_2(z) = X(z)$

$H_1(z) \cdot H_2(z) = 1$

$$\frac{B(z)}{A(z)} \cdot \underbrace{\frac{A(z)}{B(z)}}_{\downarrow} = 1$$

Exchange the roots of numerator and denominator

If we need only to maintain phase or only magnitude then ~~so~~ we use this → we cannot get by least square method

$$Y_2(e^{j\omega}) = H_1(e^{j\omega}) H_2(e^{j\omega}) . X(e^{j\omega})$$

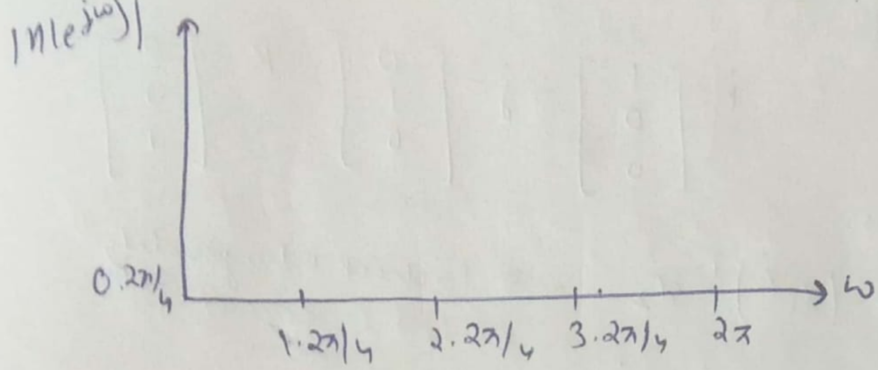
For magnitude $\quad |H_1(e^{j\omega}) . H_2(e^{j\omega})| = 1$

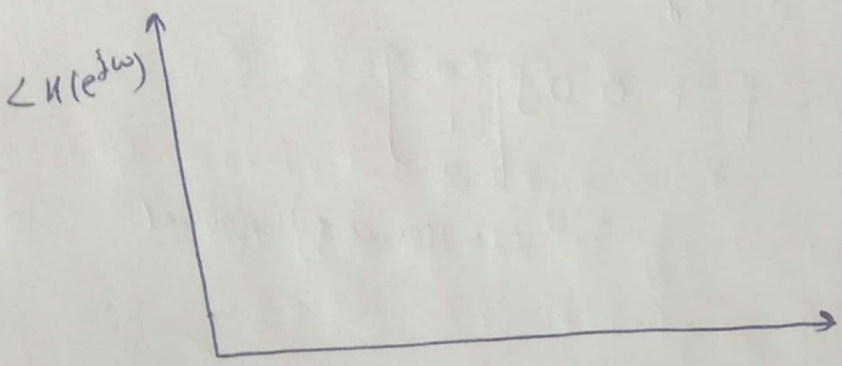
For phase $\quad \angle H_1 e^{j\omega} + \angle H_2 e^{j\omega} = 1$

↳ used in processing audio.

Sampling → Picking up N numbers with equal spacing

↳ equal part condition is being imposed.





$0.\frac{2\pi}{4}, 1.\frac{2\pi}{4}$ ..... are the eigen functions

But for a system to be defined, we need eigen values corresponding to eigen vectors

$\cancel{e^{j\omega k(\frac{2\pi}{N})n}}$ $\quad e^{j(K)(\omega)n}$ $\quad \omega = \frac{2\pi}{N}$

$K = 0, 1, 2, 3$ $\quad$ for $N = 4$

Eigen function $= e^{j\,0\cdot\left(\frac{2\pi}{4}\right)n}$ $\quad e^{j(1)\left(\frac{2\pi}{4}\right)n}$ $\quad e^{j2\cdot\left(\frac{2\pi}{4}\right)n}$ $\quad e^{j3\left(\frac{2\pi}{4}\right)n}$

$S_0(n)$ $\quad S_1(n)$ $\quad S_2(n)$ $\quad S_3(n)$

$S_N(n)$

↳ dimensions

$S_K(n)$

↑

Represents eigen function set

$(\hat{i}, \hat{j}, \hat{k}) \rightarrow$ Using this any vector can be represented in 3D.

$i \cdot j = 0$ $\quad i = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad j = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad K = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

Hermetian conjugate response:

$\langle x_1, x_2 \rangle = (x_1^*)^T x_2 \rightarrow$ To check orthogonality

$= (i^T) j$

$= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

$= 0 \hookrightarrow$ Hence orthogonal

$i \cdot i = 1$

$\alpha_1 \hat{i} + \alpha_2 \hat{j} + \alpha_3 \hat{k}$

$\alpha_1, \alpha_2, \alpha_3 \rightarrow$ Scale factors

N-dimensional : $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$

$x_1 = [1\ 0\ 0\ 0 \ldots 0]^T$

$x_2 = [0\ 1\ 0 \ldots 0]$

$x_n = [0\ 0 \ldots 1]^T$

For $n = 0$

$S_0(0) = e^{j \cdot 0 \cdot \frac{2\pi}{4} \cdot 0} = 1$

$S_1(1) = e^{j\, 0 \cdot \frac{2\pi}{4} \cdot 1} = 1$

$S_0(2) = 1$ (Similarly)

$S_0(3) = 1$

$S_0(n) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

$$S_1(0) = e^{j \cdot 1 \cdot \frac{2\pi}{4}(0)} = 1$$

$$S_1(1) = e^{j \cdot 1 \cdot \frac{2\pi}{4}(1)} = j$$

$$S_1(2) = e^{j(1)(\frac{2\pi}{4})(2)} = -1$$

$$S_1(3) = e^{j(1)(\frac{2\pi}{4})(3)} = -j$$

$$S_1(n) = \begin{bmatrix} 1 \\ j \\ -1 \\ -j \end{bmatrix}$$

$$\langle x_1 \cdot x_1 \rangle = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} = 4$$

$$\langle x_2 \cdot x_2 \rangle = 4$$

$$\langle x_1 \cdot x_2 \rangle = 0$$

$$\frac{1}{4} \langle x_2 \cdot x_2 \rangle = 1$$

$$i \cdot j = 0 \text{ but } i \cdot i \neq 1$$

Δ We are restricting that n is the dimensions length of the function

Unit norm vectors $\begin{cases} \hat{i}\cdot\hat{i} = 1 \\ \hat{j}\cdot\hat{j} = 1 \\ \hat{k}\cdot\hat{k} = 1 \end{cases}$

Here, i, j, k forms the basis of representing any vector with $\alpha_1, \alpha_2, \alpha_3$ as scaled factors of the basis.

3 dimensional vector → 3 orthogonal vectors

There can be any set of three vectors not necessarily (i, j, k) but they should satisfy ~~the~~ orthogonality ie each vector should be orthogonal to other two vectors.

$N=4$

$\hookrightarrow \omega$

To get a 4 length vector $0 \le n \le 3$

$S_0(n)$ $S_1(n)$ $S_2(n)$ $S_3(n)$ $\longrightarrow$ They are the orthogonal to each other

$\omega_K = 0, \pi/2, \pi, 3\pi/2$

To Prove that the given eigen set is orthogonal,
we have to prove that each element ($S_0(n), S_1(n), S_2(n), S_3(n)$)
one is orthogonal to other three elements,

4 equal divisions

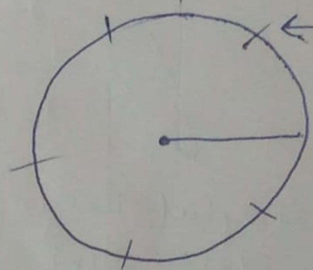
$S_0(n) = e^{j0\left(\frac{2\pi}{4}\right)n}$ $\longrightarrow$ Fundamental Frequency $\left(\frac{2\pi}{N}\right)$

$S_1(n) = e^{j1\left(\frac{2\pi}{4}\right)n}$

$S_2(n) = e^{j2\left(\frac{2\pi}{4}\right)n}$

$S_3(n) = e^{j3\left(\frac{2\pi}{4}\right)n}$

harmonics



$\leftarrow$ equal parts

$n=0$

$e^{j0\left(\frac{2\pi}{4}\right)(0)} = 1$

$e^{j1\left(\frac{2\pi}{4}\right)(0)} = 1$

$\vdots$

$e^{j3(2\pi/4)(0)} = 1$

$n=1$

$e^{j0\left(\frac{2\pi}{4}\right)(1)} = 1$

$e^{j1(2\pi/4)(1)} = e^{j2\pi/4}$

$e^{j2(2\pi/4)(1)} = e^{j2\cdot 2\pi/4}$

$e^{j3(2\pi/4)(1)} = e^{j3\cdot 2\pi/4}$

$\longleftrightarrow$

$n=2$

$e^{j0\left(\frac{2\pi}{4}\right)(2)} = 1$

$e^{j(1)\left(\frac{2\pi}{4}\right)(2)}$

$e^{j(2)(2\pi/4)(2)}$

$e^{j(3)(2\pi/4)(2)}$

$n=2$ is square of $n=1$

Similarly we can see, $n=3$ is cube of $n=1$

$\underline{n=3}$

$e^{j(0)(2\pi/4)(3)} \quad e^{j(1)(2\pi/4)(3)} \quad e^{j(2)(2\pi/4)(3)} \quad e^{j(3)(2\pi/4)(3)}$

They are cube of elements of n=1

Hence, we can say that these sequences have a structure

$$\begin{array}{c|cccc}
i=1 & 1 & 1 & 1 & 1 \\
i=2 & e\,1 & e^{j2\pi/4} & e^{j(2)(2\pi/4)} & e^{j(3)(2\pi/4)} \\
i=3 & 1 & e^{j2(2\pi/4)1} & e^{j(2)(2\pi/4)(2)} & e^{j(3)(2\pi/4)(2)} \\
i=4 & 1 & e^{j(3)(2\pi/4)} & e^{j(2)(2\pi/4)(2)} & e^{j(3)(2\pi/4)(3)}
\end{array}$$

Simplied Matrix

$$\begin{array}{c|cccc}
 & S_0(n) & S_1(n) & S_2(n) & S_3(n) \\
\hline
i=1 & 1 & 1 & 1 & 1 \\
i=2 & 1 & j & -1 & -j \\
i=3 & 1 & -1 & 1 & -1 \\
i=4 & 1 & -j & -1 & j
\end{array}$$

Fundamental row,
Once we write this row we can get the other rows from this only

To check orthogonality for $S_0(n)$

Take $S_0(n)$ as a vector

$\langle S_0(n), S_0(n) \rangle = [S_0^*(n)]^H . S_0(n)$

dot product

$= [S_0^*(n)]^T (S_0(n))$

Since $S_0(n)$ is a real set it is equal to its transpose

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$= 4$

$\langle S_1(n), S_0(n) \rangle = [S_1^*(n)]^T S_0(n)$

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

$= \cancel{1} + \cancel{1} + \cancel{1} \ 1 - j - 1 + j$

$= 0$

$\langle S_2(n), S_0(n) \rangle = [S_2^*(n)]^T S_0(n)$

$$= \begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$= 1 - 1 + 1 - 1$

$= 0$

$\langle S_3(n), S_0(n) \rangle = [S_3^*(n)]^T S_0(n)$

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

$= 1 + j - 1 - j$

$= 0$

By this we get $S_0(n)$ is orthogonal to $S_1(n)$, $S_2(n)$, $S_3(n)$

$$
\begin{array}{c} S_0^H(n) \\ S_1^H(n) \\ S_2^H(n) \\ S_3^H(n) \end{array}
\frac{1}{\sqrt{N}}\underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}}_{\text{then this } D^H}
\frac{1}{\sqrt{N}}\begin{array}{c} \begin{array}{cccc} S_0(n) & S_1(n) & S_2(n) & S_3(n) \end{array} \\ \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & j \end{bmatrix}}_{D \text{ (DFT)}} \end{array}
$$

If the resultant matrix is diagonal matrix then we can conclude that they are othogonal to each other.

$$
= \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}
$$

↳ $4I$ ↳ Identity Matrix

but we wanted it to be a unit norm

↓

it helps to show Parseval's equation (all transformations preserve the energy)

but right not this is not, So we have a problem with scaling factor

So ~~then~~ we multiply both matrices by 1/2

$$\left(\frac{1}{2}\right)\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \quad \left(\frac{1}{2}\right)\begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & j \end{bmatrix}$$

We extend it to n, then we get orthogonal sample points till n

For any general N,

the scaling factor will $\frac{1}{\sqrt{N}}$

Example:

$X = 3i + 4j + 7K \rightarrow$ vector

$\langle X, i\rangle = \langle (3i+4j+7K), i\rangle \rightarrow$ projection on i

Given a 4 length sequence

we need to find component of $x(n)$ in direction

$\hookrightarrow (s_0(n), (s_1(n), (s_2(n), (s_3(n))$

Ex:

$$x(n) = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

~~Coeffe~~ Components in direction of $S_0(n), S_1(n), S_2(n), S_3(n)$

$\langle S_0(n), x(n)\rangle \quad [\ S_0^4(n)\ ] \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$

$\langle S_1(n), x(n)\rangle \quad [\ S_1^4(n)\ ] \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$

$$\begin{bmatrix} S_0^H(n) \\ S_1^H(n) \\ S_2^H(n) \\ S_3^H(n) \end{bmatrix}_{4\times4} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}_{4\times1}$$

DFT coefficient $\rightarrow \begin{bmatrix} \\ \\ \end{bmatrix}_{4\times1}$ ↳ projection values

~~DFT Transformation Matrix~~

D^H = DFT transformation Matrix

D = Inverse DFT transformation Matrix

As we know

$$A \cdot B = I$$

and $D^H \cdot D = I$

$$D^H = D^{-1}$$

Analysis Equation

Synthesis Equation

$X = D^H x(n)$ → Multiply by D on both sides → $x = DX$

X → DFT Coefficient set

D → Inverse DFT Transformation Matrix

$N = 4, 8, 16.$

$\uparrow$

4, 8, 16 $\rightarrow$ equal divisions

28/09/18

$\langle \frac{1}{\sqrt{N}} S_K(n) \cdot x(n) \rangle$ $\hookrightarrow$ ~~a~~ gives coefficient of K^{th} component.

$\downarrow$

For dot ~~pot~~ product, we take hermation as ~~ad~~ $S_K(n)$ is not real.

$S_K^H(n) \; x(n)$

$\uparrow$ column vector

$$\begin{bmatrix} S_0^H(n) \\ S_1^H(n) \\ S_2^H(n) \\ S_3^H(n) \end{bmatrix} \underset{x(n)}{\begin{bmatrix} \\ \\ \\ \end{bmatrix}}$$

column vector $\rightarrow S_0(n) = e^{j\left(\frac{2\pi}{N}\right)0.n}$

conjugate and transpose

$S_0^H(n) = e^{-j\left(\frac{2\pi}{N}\right)0.n}$

$\uparrow$ row vector

$$\begin{bmatrix} e^{-j\left(\frac{2\pi}{4}\right)0.n} \\ e^{-j\left(\frac{2\pi}{4}\right)1.n} \\ e^{-j\left(\frac{2\pi}{4}\right)2.n} \\ e^{-j\left(\frac{2\pi}{4}\right)3.n} \end{bmatrix} \begin{bmatrix} x(n) \\ x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

For $0 \le n \le 3$, If we multiply $x(n)$ with $e^{j\left(\frac{2\pi}{4}\right)0.n}$ we will get

$e^{-j\left(\frac{2\pi}{4}\right)0(0)} x(0) + e^{-j\left(\frac{2\pi}{4}\right)0(1)} x(1) + e^{-j\left(\frac{2\pi}{4}\right)0(2)} x(2) + e^{-j(2\pi/4)0(3)}$

$$= \sum_{n=0}^{N-1} x(n) \, e^{\left(-j\frac{2\pi}{N}k\right)n} \quad -①$$

Since we want $S_K^{a}(n)$ to be a unit norm vector we will divide the matrix by $\frac{1}{\sqrt{N}}$ and similarly multiply the summation by $\frac{1}{\sqrt{N}}$ (4)

$X = D^H x \longrightarrow$ Analysis Equation

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{N}} S_0^H(n) \\ \frac{1}{\sqrt{N}} S_1^H(n) \\ \frac{1}{\sqrt{N}} S_2^H(n) \\ \frac{1}{\sqrt{N}} S_3^H(n) \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

$$X(0) = \frac{1}{\sqrt{N}} \sum_{n=0}^{n=N-1} x(n)\, e^{\left(-j\frac{2\pi}{N}\right)0 \cdot n}$$

$$X(1) = \frac{1}{\sqrt{N}} \sum_{n=0}^{n=N-1} x(n)\, e^{\left(-j\frac{2\pi}{N}\cdot 1\right)n}$$

$$X(2) = \frac{1}{\sqrt{N}} \sum_{n=0}^{n=N-1} x(n)\, e^{\left(-j\frac{2\pi}{N}\right)\cdot 2n}$$

$$X(3) = \frac{1}{\sqrt{N}} \sum_{n=0}^{n=N-1} x(n)\, e^{\left(-j\frac{2\pi}{N}\cdot 3\right)n}$$

$$\boxed{X(K) = \frac{1}{\sqrt{N}} \sum_{n=0}^{n=N-1} x(n)\, e^{\left(-j\frac{2\pi}{N}K\right)n}} \longrightarrow B$$

Coefficient of K^{th} term

DFT equation

where $\boxed{0 \le K \le N-1}$

and $\boxed{0 \le n \le N-1}$

$3i + 4j + 7k \longrightarrow k^{th}$ component of input

$\uparrow$ i^{th} component of input

$\uparrow$ j^{th} component

It is equivalent to

$$3\begin{bmatrix}1\\0\\0\end{bmatrix} + 4\begin{bmatrix}0\\1\\0\end{bmatrix} + 7\begin{bmatrix}0\\0\\1\end{bmatrix} = \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}3\\4\\7\end{bmatrix} \quad \text{— (2)}$$

Similarly,

$$X(0)\cdot\left(\frac{1}{\sqrt{N}} S_0(n)\right) + X(1)\left(\frac{1}{\sqrt{N}} S_1(n)\right) + X(2)\left(\frac{1}{\sqrt{N}} S_2(n)\right) + X(3)\left(\frac{1}{\sqrt{N}} S_3(n)\right)$$

$$\frac{1}{\sqrt{N}} \sum_{K=0}^{N-1} X(K)\, e^{j\left(\frac{2\pi}{N}\right)Kn}$$

$$\boxed{x(n) = \frac{1}{\sqrt{N}} \sum_{K=0}^{N-1} X(K)\, e^{\left(j\frac{2\pi}{N}n\right)K}} \longrightarrow \text{IDFT}$$

This is different in book

We can represent in matrix form similar to (2)

column vector

$$x(n) = \underbrace{\frac{1}{\sqrt{N}}\begin{bmatrix} S_0(n) & S_1(n) & S_2(n) & S_3(n) \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} x(0)\\ x(1)\\ x(2)\\ x(3)\end{bmatrix}}_{X}$$

$$\boxed{x = DX}$$

$$D = \frac{1}{\sqrt{N}}\left[S_0(n)\ S_1(n)\ S_2(n)\ S_3(n) \right.$$

We have $X = D^H x$

$$DX = D \cdot D^H x$$

$$\boxed{x = DX}$$

$$D \cdot D^H = I = D^H \cdot D$$

$$x[n] = \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \qquad \begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix}$$

These two ~~vectors~~ are mapped to each other

They are not the only unique 4 vectors to represent a vector in 4 dimensional plane, there can be many as long as they are orthogonal to each other.

It is not that we have only fourier transformation, there are many. We can represent a given into many other transformations.

Why Transformation ?

$$\underset{N \times N}{D^H} \; \underset{N \times 1}{x}$$

↳ we do N^2 multiplication to get this transformation

The number of nonzero elements in transformed domain is less that is it gives us a sparse vectors.

Sparse representation.

In the above example best transformation is when only one non zero is element is present in transformed domain.

1/10/18

$$\begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ \vdots \end{bmatrix} = \frac{1}{\sqrt{N}} \begin{bmatrix} s_0(n) & s_1(n) & s_2(n) & \cdots \\ & & & \end{bmatrix} \begin{bmatrix} X(0) \\ \vdots \\ \vdots \\ X(N-1) \end{bmatrix}$$

$$= \frac{1}{\sqrt{N}}\left(X(0)\, s_0(n) + X_1(n)\, s_1(n) + X_2(n)\, s_2(n) \cdots + X(n-1)\, s_{N-1}(n)\right)$$

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

$$W_N = e^{-j\frac{2\pi}{N}}$$

↑
Standard Notation

$$X(K) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) \left(e^{-j\frac{2\pi}{N}}\right)^{kn}$$

Here we can write,

$$X(K) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n)\, W_n^{kn}$$

Similarly,

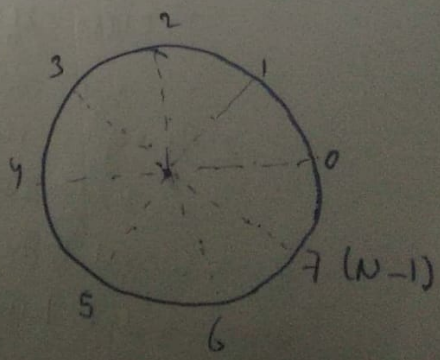
$$x(n) = \frac{1}{\sqrt{N}} \sum_{K=0}^{N-1} X(K)\, e^{j\frac{2\pi}{N}kn}$$

$$x(n) = \frac{1}{\sqrt{N}} \sum_{K=0}^{N-1} X(K)\, (W_n)^{-kn}$$

$S_K(n)$ lies on the unit circle.

Any point in n dimensional plane can be written ~~as the~~ in the Form of these n eigen functions ($s_K(n)$).

(n+1) dimensional cannot be represented with n vectors



8 equal parts

In DTFT we got infinite eigen vectors b/w 0 to 2π. Now, in DFT we got N eigen vectors only which divide the region 0 to 2π in N equal parts.

DFT also has a periodicity of 0 to 2π, it repeats after N

Let us take all coefficients from $X(1) \ldots$ to $X(N-1) = 0$

$$x(n) = X(0) S_0(n) + X(1) S_1(n) + \ldots + X(N-1)$$

$$X = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$X(0) = 4$

$$x(n) = \overset{(4)}{X(0)}\, S_0(n)$$

$$= (4) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$= \frac{4}{\sqrt{4}} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$x(n) = \frac{1}{2} \begin{bmatrix} 4 \\ 4 \\ 4 \\ 4 \end{bmatrix}$$

Only one term is nonzero

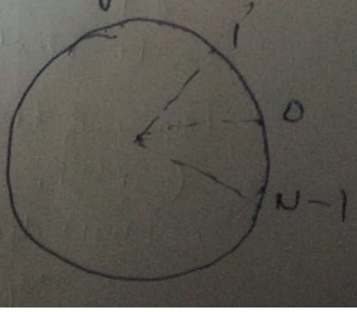
Hence it is representing only one angle $S_0(n)$

We want to take appropriate coefficients to get the desired value of $x(n)$

~~If we take $X(0) = 4$~~ ~~$X(1) = 3$~~

We can get the corresponding frequency wrt to any eigen vector $S_k(n)$ by taking only that vector and rest zero.

If we want 1, (N-1) this will us a sinusoidal function



$$x(n) = \begin{bmatrix} 1 \\ 5 \\ 2 \\ \vdots \\ \vdots \end{bmatrix}_{10\times 1}$$

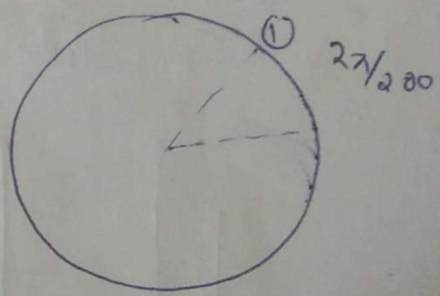
↳ 10 basis vectors to represent

Bases Vectors $S_0(n)$ $\vdots$ $S_9(n)$

$$X = \begin{bmatrix} \\ \\ \\ \end{bmatrix}_{10\times 1}$$

If we have 100 input length and we have 200 eigen vectors i.e 200 length DFT sequence

$$\begin{bmatrix} \\ \\ \end{bmatrix}_{200\times 100} \begin{bmatrix} \\ \\ \end{bmatrix}_{100}$$



Is it possible to get 100 X 1 input from here.

Modulo Operator:

* delay is the fundamental operation.

$(K)_N = \boxed{K \text{ modulo } N}$

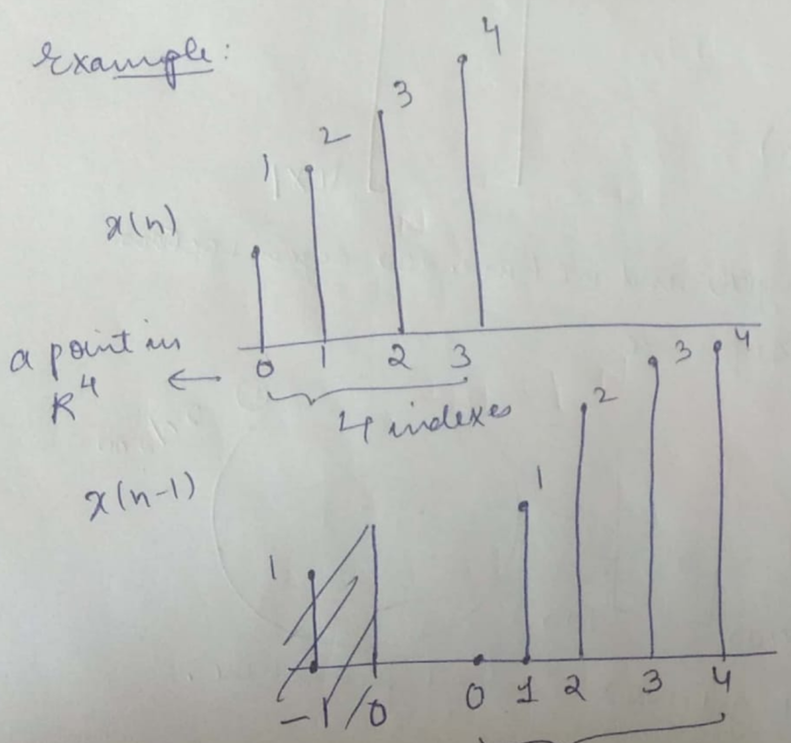
gives the remainder between 0 to N − 1

$\boxed{K = r_1 N + r}$

x(n) is finite length sequence of length 'N'

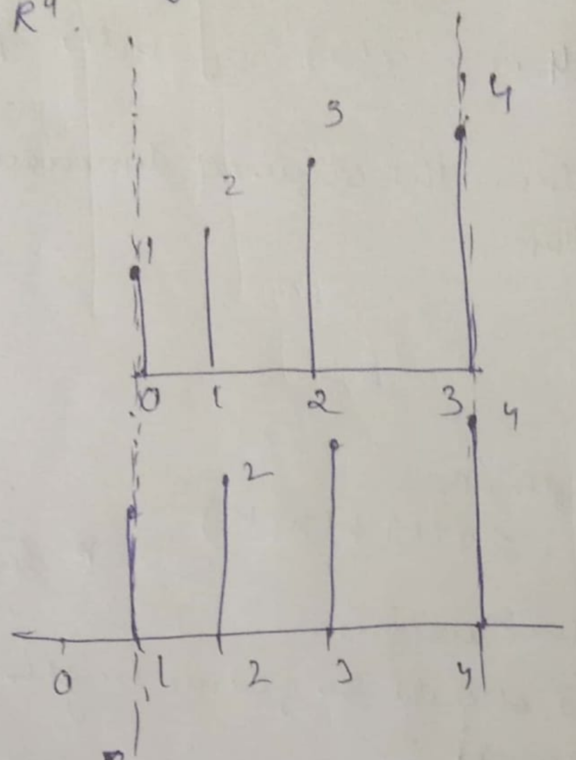
~~length of x(n−1) is (N+1)~~

example:



Increasing dimensions ~~should~~ is not acceptable.
If x(n) is in R^4 then all the manipulations on x(n) should result in a point in R^4 only.

Let us fix the starting and ending point to get a shifted sequence in R^4.



Fixed dimension

$$x\left[(n-1)_N\right] = x\left((n-1) \text{ modulo } N\right)$$

Notation of Modulo Operator in fixed dimension

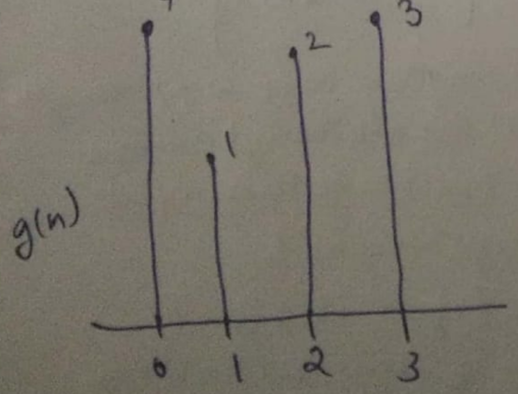
$$x\left((n-1)_4\right) = g(n)$$

$$g(0) = x\left((0-1)_4\right)$$

$$= x\left((-1)_4\right)$$

$K = r_1 N + r$

$-1 = (-1).4 + 3$



$$\boxed{g(0) = x(3)}$$

$$g(1) = x\left((0)_4\right)$$

$K = r_1 N + r$

$0 = (0)(4) + 0$

$$\boxed{g(1) = x(0)}$$

$K = r_1 N + r$

$1 = 0(4) + 1$

$$g(2) = x\left((1)_4\right)$$

$$\boxed{g(2) = x(1)}$$

$$g(3) = x\left((2)_4\right)$$

$$\boxed{g(3) = x(2)}$$

$$g(4) = x((4-1)_4) = x(3)$$
$$g(5) = x((4)_4) = x(0)$$
Periodic sequence with $N=4$

Modulo operation maintains the same ~~dif~~ dimension even after the shift operation.

Effect on convolution:

$x(n)$, $h(n)$

length $= n_1$, length $= n_2$

$$y(n) = \sum x(K)\, h(n-K)$$

length of $y = n_1+n_2-1$

Now for fixed dimensions

$x(n)$, $h(n)$ should be of same length.

length $= N$, length $= N$

$$x(n) * h(n) = \sum_{K=0}^{N-1} x(K)\, h\left[(n-K)_N\right]$$

length of $y(n)$ = ~~...~~ N

$$x(n) \; N \; h(n) = \sum_{K=0}^{N-1} x(K)\, h\left[(n-K)_4\right]$$

correct representation

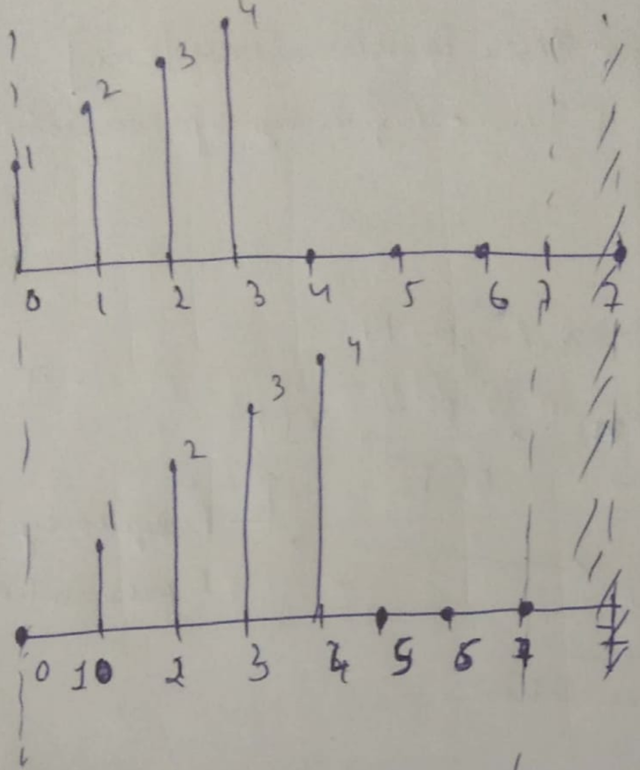
N- circular convolution

$$x(n) \; 2N \; h(n) = \sum_{K=0}^{2N-1} x(K)\, h\left[(n-K)_{2N}\right]$$

We are required to extend $x(n)$ in this case to $2N$ length ~~...~~ for that we have 3 options → Put arbitary numbers (it changes the input signal)

→ Repeat it after N (decreases the energy of the signal)

→ append zeros ~~...~~ (best option)



converting to
For ~~lin~~ linear convolution, we will use

$x(n)$ (2N-1) $h(n)$

↑

For a simple linear convolution of length 4
we use 3 delays

For same in this case
we will require 2N - 1 delays.

Circular Convolution

* $x(n) * h(n)$ — linear convolution

$x(n)$ Ⓝ $h(n)$ → circular convolution of N length

The shift operation is a modulo N operator

$$\sum_{K=0}^{N-1} x(K)\, h\left((n-K)_N\right)$$

Linear convolution

$$\underset{(2N-1)\times 1}{\begin{bmatrix} \\ \\ y(n) \\ \\ \\ \end{bmatrix}} = \underset{\substack{(2N-1)\times N \\ (N+N-1)}}{\begin{bmatrix} h(0) & & \\ h(1) & & \\ \vdots & & \\ h(N-1) & & \\ 0 & & \\ \vdots & & \\ 0 & & \end{bmatrix}} \underset{N\times 1}{\begin{bmatrix} x(0) \\ \vdots \\ x(N-1) \end{bmatrix}}$$

For circular convolution

$$\underset{N\times 1}{\begin{bmatrix} \\ y(n) \\ \\ \end{bmatrix}} = \underset{N\times N}{\begin{bmatrix} h(0) & h(N-1) & h(N-2) & h(1) \\ \vdots & h(0) & & h(2) \\ & & & h(3) \\ \vdots & \vdots & & \vdots \\ & & & h(N-1) \\ h(N-1) & h(N-2) & & h(0) \end{bmatrix}} \underset{N\times 1}{\begin{bmatrix} \\ x(n) \\ \\ \end{bmatrix}}$$

This is circulant matrix

Operation is to be done by circular convolution but resultant is desired as linear convolution.

How can the circular convolution operator give the same result as linear convolution?

Scanned by CamScanner

$x(n)$ (N) $h(n)$

$\downarrow$ use

$2N-1$

(because output length will be $2N-1$)
we want $x(2N-1)$ to be equal come at 0 index

DTFT $x(n) \xleftrightarrow{DTFT} X(e^{j\omega})$

$h(n) \xleftrightarrow{DTFT} H(e^{j\omega})$

$x(n) * h(n) \xleftrightarrow{DTFT} X(e^{j\omega})\, H(e^{j\omega})$

$x(n)$ and $h(n)$ are of finite length 'N'

$x(n) \xleftrightarrow{N\text{point DFT}} X(K)$

$h(n) \xleftrightarrow{N\text{point DFT}} H(K)$

$0 \le K \le N-1$

$x(n)$ (N) $h(n) \xleftrightarrow{N\text{point DFT}} X(K) \cdot H(K)$

$(2N-1)\,N$ Multiplications required ?

$(2N-1)\,(N-1)$ Additions Required ?

Alternatives for reducing the number $2N-1$

$x(n) * h(n) \xleftrightarrow{DTFT} X(e^{j\omega})\, H(e^{j\omega})$

We need to apply DTFT on $x(n)$ and $h(n)$ which will give $X(e^{j\omega})$ $H(e^{j\omega})$. Take product and then apply inverse. This is the alternative path for getting $y(n)$. but this method won't reduce the number $2N-1$

Scanned by CamScanner

Complexity → no. of multiplication / division

No. of addition / Subtraction

Complexity of linear convolution

$$\begin{bmatrix} \\ \\ \\ \end{bmatrix}_{2N-1} = \begin{bmatrix} & h & \end{bmatrix}_{2N-1 \times N} \begin{bmatrix} x \end{bmatrix}_{N \times 1}$$

for one row

N multiplications

N-1 additions

For total rows = $(N)(2N-1)$ multiplication

$= 2N^2 - N \qquad O(N^2)$

Addion $= (2N-1)(N-1)$

$= 2N^2 - N - 2N + 1 \qquad O(N^2)$

Total multiplication per sample = $\dfrac{(2N-1) \times N}{2N-1}$

↳ Total no. of samples in y (output)

For circular convolution:

$$\begin{bmatrix} \\ \\ \\ \end{bmatrix}_{N \times 1} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}_{N \times N} \begin{bmatrix} \\ \\ \end{bmatrix}_{N \times 1}$$

Multiplication $= N^2 \qquad O(N^2)$

Addition $= N(N-1) \qquad O(N^2)$

$$x(n) \xleftrightarrow{\text{N-point DFT}} X(K)$$

$$X = D^H x$$

$$\underbrace{\begin{bmatrix} \\ X \\ \\ \end{bmatrix}}_{N\times 1} = \underbrace{\begin{bmatrix} \\ D^H \\ \\ \end{bmatrix}}_{N\times N} \underbrace{\begin{bmatrix} \\ x \\ \\ \end{bmatrix}}_{N\times 1}$$

Complexity $= O(N^2)$

Requires $= N^2$ complex multiplications

$(N-1)N$ complex additions

1 complex Multiplication $\approx$ 4 Real Multiplications

$(a+ib)(a+ib)$

1 complex addition $=$ 2 Real Additions

$(a+ib)+(a+ib)$

To get the convolution using N-point ~~Fourier~~ DFT transforms

$$h(n) \xleftrightarrow{(2N-1)\text{ point DFT}} H(K)$$

$$x(n) \xleftrightarrow{(2N-1)\text{ point DFT}} X(K)$$

DFT and Inverse DFT have same complexity.

$$y(n) = x(n) \, (N) \, h(n) \xleftrightarrow[\text{point DFT}]{2N-1} X(K)H(K)$$

Total 3 DFT communications

Fast Fourier Transform

Algorithm used to reduce the number of complexities

It takes only $\underbrace{\frac{N}{2}\log_2 N}_{\text{complexity of FFT}}$ multiplication.

$$x(n) \longrightarrow \boxed{h(n)} \longrightarrow y(n)$$

If $N = 8$ use FFT

FFT → An algorithm to compute DFT more efficiently and with lesser multiplication.

Parseval's Relation of DFT Transform

$$X = D^H x$$

IDFT

$$x = DX$$

$$\sum_{n=0}^{N-1} |x(n)|^2 = \sum_{k=0}^{N-1} |X(k)|^2$$ → Conservation of Energy

(because length of $x(n)$ is of length 'N')

In books: $\cancel{\sum_{n=0}^{N-1} = \sum_{k=0}^{N-1}}$

$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |X(k)|^2$$

If they have not included $\frac{1}{\sqrt{N}}$ in D and D^H matrices)

$$|x(1)|^2 \neq |X(1)|^2$$

Energy compaction should happen.

$$x_1(n) \xleftrightarrow[\text{DFT}]{\text{N-point}} X_1(K)$$

$$x_2(n) \xleftrightarrow[\text{DFT}]{\text{N-point}} X_2(K)$$

Linearity

$$a_1 x_1(n) + a_2 x_2(n) \xleftrightarrow[\text{DFT}]{\text{N-point}} a_1 X_1(K) + a_2 X_2(K)$$

$$\cancel{x((n-n_0)_N) \longleftrightarrow e^{j 2\pi \omega n_0} X(K)}$$

$$x(\underbrace{(n-n_0)_N}_{\text{Modulo Operation}}) \xleftrightarrow[\text{DFT}]{\text{N point}} W_N^{K n_0} X(K)$$

$$W_N = e^{-j\frac{2\pi}{N}}$$

$$x((n-n_0)_N) \xleftrightarrow[\text{DFT}]{\text{N point}} \underbrace{e^{-j\frac{2\pi}{N} K n_0}}_{} X(K)$$

changing the phase of coefficients

$$0 \leq K \leq N-1$$

$$e^{-j\frac{2\pi}{N} K n_0} X(K)$$

For $K=0$

$(1) X(0)$

For $K=1$ $\quad X(1) e^{-j\frac{2\pi}{N} 1 \cdot n_0}$

For $K=2$

$$X(2).\,e^{-j\left(\frac{2\pi}{N}.n_0\right)2} \quad (\theta)$$

All the coefficients will ~~be~~ have an added extra term 'θ' which is dependent on n_0 and it is not arbitary.

$n_0 \rightarrow \theta$

$$W_N^{-K_0 n}\, x(n) \xleftrightarrow[\text{DFT}]{N\text{-Point}} X((K-K_0)_N)$$

$$e^{j\frac{2\pi}{N}K_0 n}\, x(n) \xleftrightarrow[\text{DFT}]{N\text{-point}} X((K-K_0)_N)$$

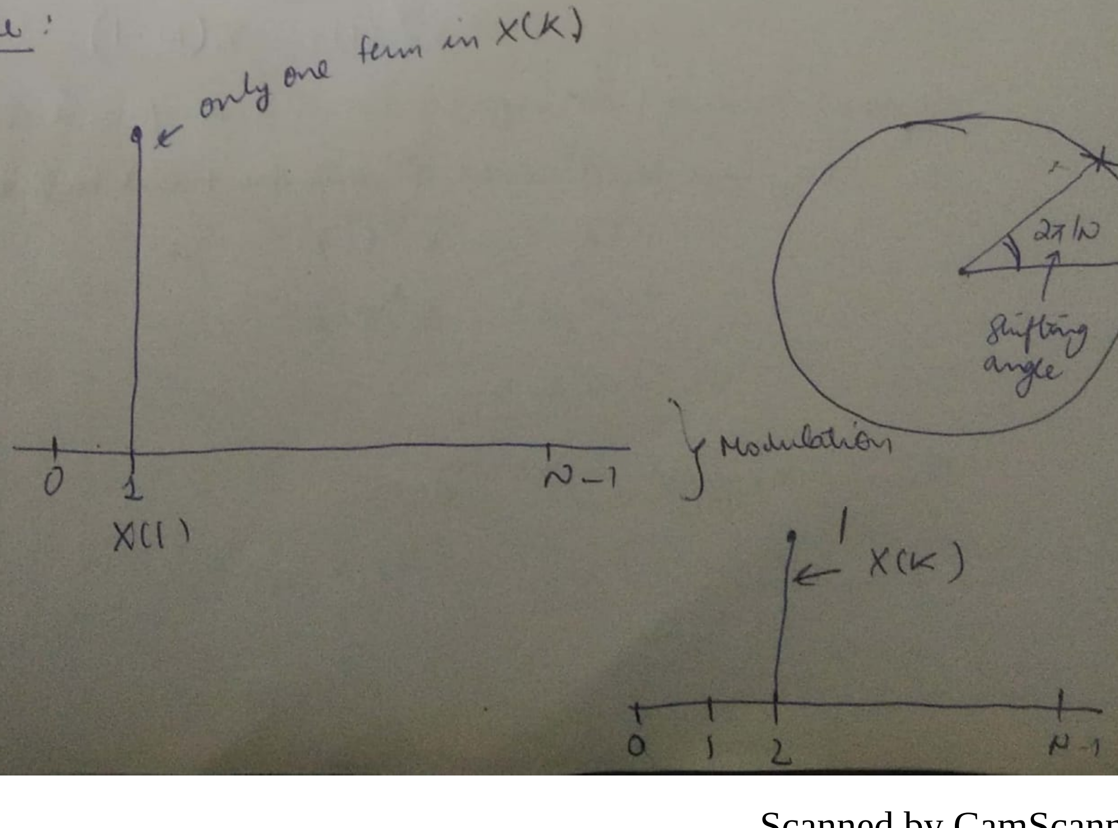
For $n=0$

$(1)\;x(0)$

$n=1$

$$e^{j\left(\frac{2\pi}{N}K_0\right)}\,x(1)$$

Example:



Symmetry Property of ~~for~~ Real Sequence $x(n)$ with length N.

Real $x(n)$ with length N

Values of $x(n)$ are always real

$$x(n) \xleftrightarrow[\text{DFT}]{\text{N-point}} X(K)$$

$$\cancel{X(K) = X^*(K)}$$

$$X(K) = X^*((N-K)_N)$$

$$X(0) = X^*((N-0)_N)$$

$$\boxed{X(0) = X^*(0)}$$

N-1

For any real $x(n)$
$X(0)$ will always be real

$$X(1) = X^*((N-1)_N)$$

$$X(1) = X^*(N-1)$$

3

To find $X(N-1)$ from $X(1)$
Take conjugate on both sides

$$X^*(1) = X(N-1)$$

We do not need to find all coefficients in transform domain as ~~the~~ they can be derived from the first half ~~of~~ Ex:

$$X(N-1) = X^*(1)$$

$$X(N-2) = X^*(2)$$

and so on.

This is conjugate symmetry.

Symmetry Property of complex Sequence $x(n)$ with length N

$$x(n) \xleftrightarrow[\text{DFT}]{\text{N-point}} X(K)$$

$$x^*(K) \xleftrightarrow{\text{N point}} X^*((N-K)_N)$$

$$x^*(n) \longleftrightarrow X^*((-K)_N)$$

Shifted by N terms

$$X^*((N-K-N)_N)$$

$$X^*((-K)_N)$$

$$x^*((-n)_N) \xleftrightarrow[\text{DFT}]{\text{N point}} X^*(K)$$

(Prove these yourself)

Two Real Sequence of same length N

$$\underbrace{x_1(n),\ x_2(n)}$$

combine and make a new sequence

$$\underset{\sim}{x_3(n)} = x_1(n) + j x_2(n)$$

Complexity of finding DFT of $x_1(n)$ and $x_2(n)$ separately $= 2n^2$

$$x_3(n) \xleftrightarrow[\text{DFT}]{\text{N point}} X_3(K)$$

Complexity of finding DFT of $x_3(n)$

$= n^2$ reduced by 50%

$$x_1(n) = \frac{x_3(n) + x_3^*(n)}{2}$$

$$x_2(n) = \frac{x_3(n) - x_3^*(n)}{2j}$$

DFT

$$X_2(K) = \frac{X_3(K) - X_3^*((-K)_N)}{2j} \qquad X_1(K) = \frac{X_3(K) + X_3^*((-K)_N)}{2}$$

12/10/18

DFT

Spectrogram [Short Time Fourier Transform]
[Time-Frequency Representation)

Till now we discussed,

$x(n)$ of length N

$$x(n) \xleftrightarrow[\text{DFT}]{\text{N-point}} X(K)$$

Time Domain Representation ⟷ Frequency Domain Representation

In this case, We can get only one of the two representation either time or frequency.

Ex → In an FM modulated signal, we have both time and frequency dependent information. If we now transform this signal we will lose the time domain content.

We want to look at changes in frequency domain wrt time in one representation. → STFT
(3-dimensional plane)
2 independent variables (t, f)

input $x(0), x(1), x(2) \ldots \ldots x(L)$
[length = $L+1$

In Block Convolution, when we divided the input into blocks, each block is independent of other.
i.e they are Non-overlapping blocks.

Let us take a input with 100 length and divide into block size of 10

$x(0)$	$x(9)$
$x(10)$	$x(19)$

} Non overlapping

Max. allowed overlap is called 100% overlap.

$x(0)$ ... $x(9)$ → 0 overlap

$x(1)$ ... $x(10)$ → Max overlap

$x(2)$ ... $x(11)$

Range of overlap (1 – 10) samples

$0 < m < N-1$

N block size M overlap $1 < M < N$

$x(0)$ $x(1)$ $x(2)$

$x(N-1)$ $x(N)$ $x(N-1)$

$x(0)$ $x(M)$ $x(2M)$ . . . . $x(L)$

0
0
0
0

} Add zeros to divide the input in equal block lengths

$x(N-1)$ $x(M+N-1)$ $x(2M+N-1)$

Scanned by CamScanner

$$= D^H_{N\times N} \begin{bmatrix} x(0) & x(M) & x(2M) & \vdots \\ \vdots & & & x_{L-1} \\ \vdots & & & 0 \\ \vdots & & & 0 \\ x(N-1) & x(M+N-1) & x(2M+N-1) & 0 \end{bmatrix}$$

Represents the zeroth ~~angle~~ component m

$$S(K, m) = \begin{bmatrix} X(0) & X(M) & X(2M) & \cdots \\ \vdots & \vdots & \vdots & \\ X(N-1) & X(M+N-1) & X(2M+N-1) & \end{bmatrix}$$

K ↓ block length

m ↑ no. of blocks ↓ gives the time information

DFT Transformed

K → frequency
m → column index

If All the zeroth components are equal it means they are same in time domain also.

Rate is adjusted by changing the amount of overlap.

Maximum overlap (M = 1) maximum rate

Minimum overlap (M = N) minimum rate (only Nth sample)

22/10/18

Short Time Fourier Transform

$x(n) = \{x(0) \ldots\ldots x(L)\}$

$1 \leq M \leq N$

$\uparrow$ max overlap $\downarrow$ No overlap

$$= D^H \begin{bmatrix} x(0) & x(M) & x(2M) & \vdots \\ \vdots & & & x(L) \\ \vdots & & & 0 \\ & & & 0 \\ & & & 0 \\ x(N-1) & x(M+N-1) & x(2M+N-1) & 0 \end{bmatrix}$$

M Space apart in time

$$= \begin{bmatrix} \underbrace{X_1(0) \quad X_2(0)} & X_3(0) \\ \vdots & \vdots & \vdots \\ X_1(N-1) & X_2(N-1) & X_3(N-1) \end{bmatrix}$$

$\downarrow$

$X(0) = \langle s_0(n), x(n) \rangle$

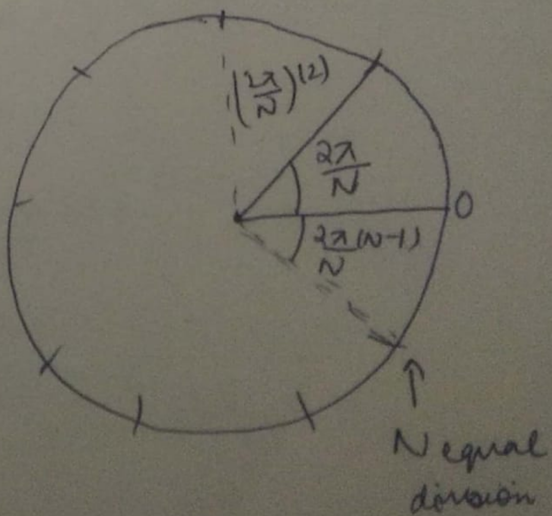
$s_1(n) = e^{j\frac{2\pi}{N}n}$

$s_{N-1}(n) = e^{j\frac{2\pi}{N}(N-1)n}$

$= e^{j2\pi n - j\frac{2\pi}{N}n}$

$= 1\, e^{-j\frac{2\pi}{N}n}$

$= s_1^*(n)$



$|X(0)|^2$

↳ Gives the energy of the corresponding component

$$|X(0)|^2 = X(0) \cdot X^*(0)$$

$$|X(0)|^2 + \ldots\ldots |X(N-1)|^2$$

$= $ Energy of signal

We see the individual contribution of each component and identify which one is dominant.

$$\begin{bmatrix} X_1(1) & X_2(1) & X_3(1) & \cdots \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \\ \vdots & \vdots & \vdots & \\ X_1(N-1) & X_2(N-1) & X_3(N-1) & \end{bmatrix} = X(K, m)$$

$K \downarrow$ Frequency component

$m \downarrow$ column index

Only these components are present and rest are zero

means the input signal is not changing with frequency.

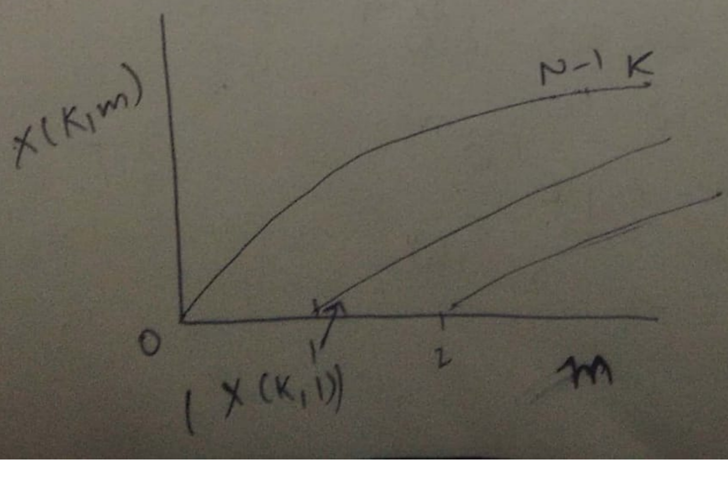
No change in frequency content.

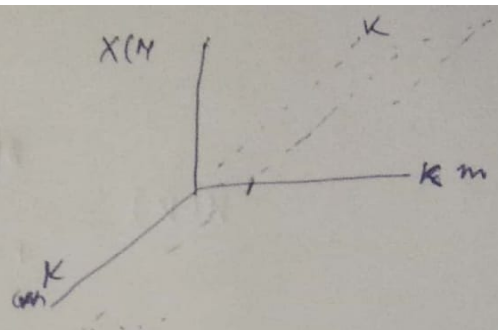
If we choose $M=1$ no. of columns will increase and hence complexity will increase which is unnecessary.

$x, y \rightarrow$ time and Frequency

z - strength

$|X(K, m)|^2 \rightarrow$ energy





If we take $|X(K,m)|^2$

and $\angle X(K,m)$ for plot

then it is called Energy Spectrum

And this analysis is called Spectral Analysis

The larger the N, the more frequency component ~~can~~ are ~~be~~ observed.

Decimation in Time fast Fourier Transform

DIT FFT

↳ N is required

↳ Now N should be some power of 2

$N = 2^0, 2^1, 2^2, 2^3 \ldots$

$N = 1, 2, 4, 8, \ldots$

input signal $x(n)$

$$X(K) = \sum_{i=0}^{N-1} x(n) W_N^{Kn}$$

Divide $x(n)$ into 2 equal parts → odd index ↘ even index

$g(n) = x(2n)$ length $= \frac{N}{2}$

$h(n) = x(2n+1)$ length $= \frac{N}{2}$

16 87

Now, we ~~have~~ find out DFT for each of these parts

i.e for $g(n)$ and $h(n)$

of size $\frac{N}{2}$

Hence the complexity is reduced

$$G(K) = \sum_{n=0}^{N/2-1} g(n) W_N^{2Kn}$$

$$W_{N/2}^{nK} = W_N^{2Kn} = e^{-j\frac{2\pi}{N}2nK} = e^{-j\frac{2\pi}{N/2}nK}$$

$$H(K) = \sum_{n=0}^{N/2-1} x(2n+1) W_N^{(2n+1)K}$$

$2 \times \frac{N}{2} - 1$

$(N-2)$

$$W_N^{(2n+1)K} = e^{-j\frac{2\pi}{N}(2n+1)K}$$

$$= e^{-j\frac{2\pi}{N}2nK} \cdot e^{-j\frac{2\pi}{N}K}$$

$$= W_N^{2nK} \; W_N^{K}$$

$$H(K) = \sum_{n=0}^{N/2-1} x(2n+1) W_N^{2nK}$$

$$X(K) = \sum_{n=0}^{N-1} x(n) W_N^{Kn}$$

$$= \sum_{n=0}^{N/2-1} x(2n) W_N^{2Kn} + \sum_{n=0}^{N/2-1} x(2n+1) W_n^{(2n+1)K}$$

$$X(K) = G(K) +) W_N^{K} \sum_{n=0}^{N/2-1} x(2n+1) W_N^{(2n)K}$$

$$\boxed{X(K) = \underbrace{G(K)}_{\substack{N/2 \\ (\text{length})}} + W_N^K \underbrace{H(K)}_{\substack{N/2 \\ \text{length}}}}$$

which gives $N/2$ length of $X(K)$

N point DFT is periodic with N

$$G(K) = G\left(K + \frac{N}{2}\right)$$

$$H(K) = H\left(K + \frac{N}{2}\right)$$

The left terms of X(K) will be

$$X\left(K+\frac{N}{2}\right) = G\left(N+\frac{K}{2}\right) + W_N^{K+N/2} H(K+N/2)$$

$$X\left(K+\frac{N}{2}\right) = G(K) + W_N^{K+N/2} H(K)$$

$$W_N^{K+N/2} = W_N^K \cdot W_N^{N/2}$$

$$= W_N^K e^{-j\cancel{2}\frac{\pi}{N}\times\frac{N}{\cancel{2}}}$$

$$= -W_N^K$$

$$X\left(K+\frac{N}{2}\right) = G(K) - W_N^K H(K)$$

$$X(K) = G(K) + W_N^K H(K) \qquad 0 \le K \le \frac{N}{2}-1$$

$$X\left(K+\frac{N}{2}\right) = G(K) - W_N^K H(K)$$

Now we can divide g(n) into parts

we can go on dividing until we get 1 sample

If we have $x(n) = 10$ 1 point DFT

$X(K) = 10$ (No multiplication required)

For 8, 3 divisions required

For divisions complexity $= \log_2 N$

For 2 point DFT we require 1 multiplication

~~Total~~ complexity for merging $\frac{N}{2}$

Total complexity $= \boxed{\frac{N}{2}\log_2 N}$

DIT FFT , Determine DFT of Sequence

$$x(n) = [4, -3, 2, 0, -1, -2, 3, 1]$$

$N = 8$

$N^2 = 64$

$x(0) = 4$	$x(0)$	$x(0)$	$x(0) = 4$
$x(1) = -3$	$x(2)$	$x(4)$	$x(4) = -1$
$x(2) = 2$	$x(4)$	$x(2)$	$x(2) = 2$
$x(3) = 0$	$x(6)$	$x(6)$	$x(6) = 3$
$x(4) = -1$	$x(1)$	$x(1)$	$x(1) = -3$
$x(5) = -2$	$x(3)$	$x(5)$	$x(5) = -2$
$x(6) = 3$	$x(5)$	$x(3)$	$x(3) = 0$
$x(7) = 1$	$x(7)$	$x(7)$	$x(7) = 1$

Merging equation:

$$X(K) = G(K) + W_N^K H(K)$$

$$0 \le K \le N/2 - 1$$

$$X(K + N/2) = G(K) - W_N^K H(K)$$

For getting 2 point DFT , $N = 2$

$K = 0$ using this

$x(0) = 4$		3			8			
$x(4) = -1$	W_2^0	5			$5 + j$			
$x(2) = 2$		5	W_4^0	1	-2			
$x(6) = 3$	W_2^0	-1	W_4^1	$-j$	$5 - j$			
$x(1) = -3$		-5			-4	W_8^0	1	
$x(5) = -2$	W_2^0	-1			$-1 + j$	W_8^1	$\frac{1-j}{\sqrt{2}}$	$\sqrt{2}j$
$x(3) = 0$		1	W_4^0	1	-6	W_8^2	$-j$	
$x(7) = 1$	W_2^0	-1	W_4^1	$-j$	$-1 - j$	W_8^3	$\frac{-1-j}{\sqrt{2}}$	$+\sqrt{2}j$

4 multiplications 4 multiplications 4 multiplications

$X(K)$

4
$5+(1+\sqrt{2})j$
$-2+6j$
$5-j+\sqrt{2}j$
12
$5+(1+\sqrt{2})j$
$-2-6j$
$5-j+\sqrt{2}j$

$$\frac{\sqrt{2}(5+j)+1-j}{\sqrt{2}}$$

$$(5\sqrt{2}+1)+\frac{(\sqrt{2}-1)j}{\sqrt{2}}$$

Since $x(n)$ is real
$X(K)$ should satisfy conjugate
symmetry

Total no. of multiplication reduced to $\underline{\underline{12}}$

$$\frac{N}{2}\log_2 N \qquad (4)\log_2(8)$$

$$(4\times 3)$$

$$\frac{(1-j)}{\sqrt{2}}(-1+j)$$

$$\frac{1}{\sqrt{2}}(\cancel{-1}+j+j+\cancel{1})$$

$$\frac{2j}{\sqrt{2}}$$

$$\sqrt{2}j$$

$$-\frac{(1+j)(1+j)}{\sqrt{2}}$$

$$-\frac{(\cancel{1}-\cancel{1}+2j)}{\sqrt{2}}$$

$$-\sqrt{2}j$$

31/10/18

2 FFT Algorithms → DIT FFT, DIF FFT } are useful in reducing the complexity from $\{N^2\}$ to $\{\frac{N}{2}\log_2 N\}$ For computing a N point DFT

exact terms

* where N is a power of 2 i.e $N = 2^x$ and x is any integer (this assumption is also the limitation)

in O() representation

$$O(N^2) \longrightarrow O(N \log_2 N)$$

$1 \quad \underbrace{2 \quad 4}_{1} \quad 8 \quad 16 \quad \underbrace{32 \quad 64}_{32} \quad 128 \quad 256 \quad 512$

As the numbers are increasing the spacing b/w them is also increasing, which means now there are more numbers for which we cannot find DFT using these methods.

For ex: B/w 256 and 512, there are 256 numbers for which the assumption is not satisfied.

To use this algorithm at any cost and no restriction on input size.

Solution 1
→ Modulo
$x(0)$, $x(1)$ and $x(2)$. Apply Modulo operator, we will get a length = 2. $x(2)$ will get added to $x(0)$

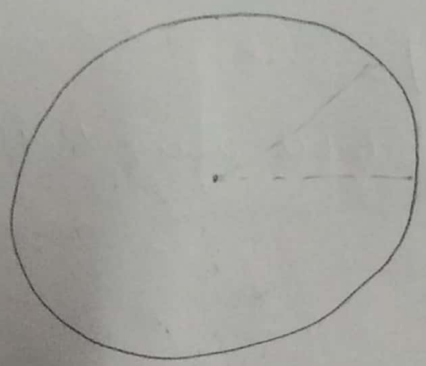
Solution 2:
Add zeros to make a 3 point DFT into a 4 point DFT.
This is a better option.

Suppose we want to have a 17 point DFT then we will compute 32 point DFT. We will have to see which is a bigger module 17^2 or $16 \log_2 32$.

Error in the worst case scenario, using this algorithm is a useful option.

We need to go the upper bound ceil 129—256. So this algorithm can be used as a default

Spectral Representation



DTFT
↓
Complete information

DFT
↳ Gives information about Specific N points

$$-\pi \leq w \leq \pi$$

When w is a continuous variable from $-\pi$ to π then it is a DTFT.

If we compute N point DFT then we divide the circle into N equal parts.

Scanned by CamScanner

Spectral component of the input signal on that particular angles.

N → ∞ DFT is equal to DTFT

In a continuous signal we are looking for the number of cycles per second.

Sampling Frequency = f_s. This is for the signals of continuous type

f_s = Number of samples / sec

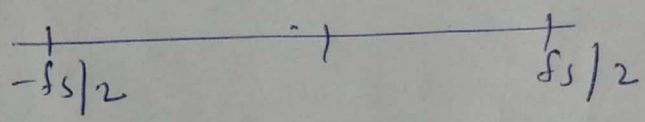
The connection point is Nyquist theorem.

This is a connection point b/w a continuous time signal to a discrete time signal.

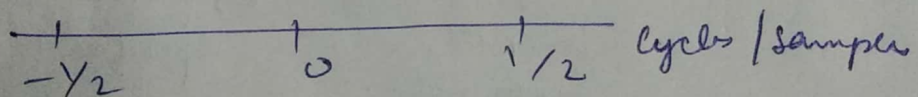
Input cannot have more than $f_s/2$.

If the input is more than $f_s/2$. Then we get aliasing in the spectral domain.

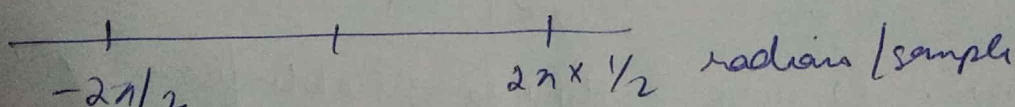
Nyquist Interval :

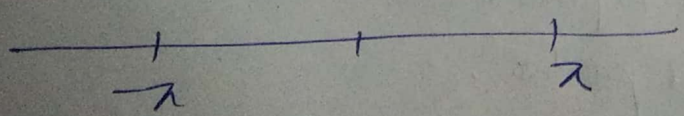


The highest Frequency representable using Nyquist interval is $f_s/2$



cycle = 2π radians





This is the scale we are operating on

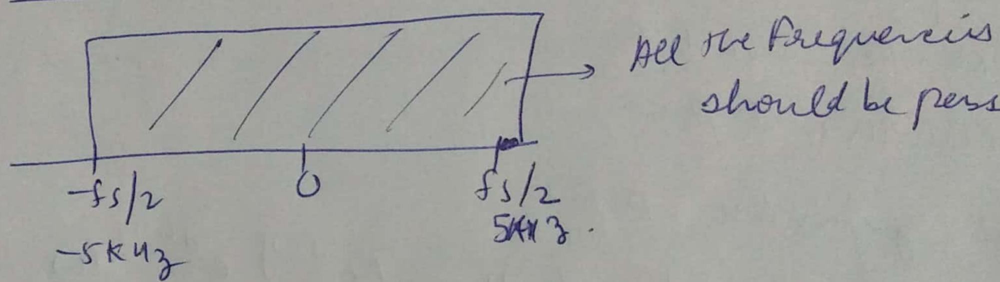
Using Nyquist Sampling theorem can be connected to digital.

In f_s is increased in analog, representable frequency will increase.

Digital representation is a fixed representation with respect to fixed f_s. If f_s is changed, there will be a different representation, They will map to different frequencies

f_s - sampling frequency

Nyquist Interval



All the frequencies after 5KHz should be blocked

$f_s = 10KHz$

~~If $f_s = 10KHz$~~

~~Reverse symmetric spectrum~~ because

w scale ranging from $\frac{-\pi f}{f_s}$ to $\frac{\pi f}{f_s}$

Low Pass Filter - It should pass all the frequencies.

Ideal low pass filter $\omega_c = \pi/5$ radian/samp

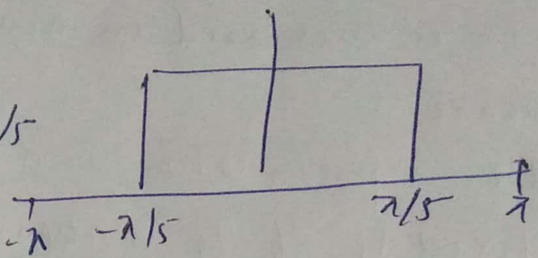
In terms of analog frequencies $f_c = 1KHz$ with $f_s = 10KHz$

$\omega = \frac{2\pi f}{f_s}$ ⟶ This is the conversion to get back to the digital scale.

$H(e^{j\omega}) = 1 \qquad -\frac{\pi}{5} \le \omega \le \frac{\pi}{5}$

$|H(e^{j\omega})| = 1 \qquad -\pi/5 \le \omega \le \pi/5$

$\qquad = 0 \qquad$ otherwise



$\angle H(e^{j\omega}) = 0 \qquad -\pi < \omega < \pi$

Zero phase response

An impulse response can be implemented using

→ Non Recursive FIR

→ Recursive IIR

Applying Inverse DTFT

$$h_{LPF}(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_{LPF}(e^{j\omega})\, e^{j\omega n}\, d\omega$$

$$\frac{1}{2\pi}\int_{-\pi/5}^{\pi/5} 1 \cdot e^{j\omega n}\, d\omega = \left[\frac{e^{j\omega n}}{jn}\right]_{-\pi/5}^{\pi/5}$$

Generalizing : For any general ω_c $\left[\frac{e^{j\omega n}}{jn}\right]_{-\omega_c}^{\omega_c}$

$$\frac{1}{2\pi}\left[\frac{e^{j\frac{\pi}{5}n}}{jn} - \frac{e^{-j\pi/5 n}}{jn}\right]$$

$$= \frac{1}{n}\sin\left(\frac{\pi}{5}n\right) = \frac{\sin\left(\pi/5\, n\right)}{n}$$

$$\left[\frac{e^{j\omega n}}{jn}\right]_{-\omega_c}^{\omega_c} = \frac{1}{2\pi}\left[\frac{e^{j\omega_c n}}{jn} - \frac{e^{-j\omega_c n}}{jn}\right]$$

$$= \frac{\sin \omega_c n}{\pi n}$$

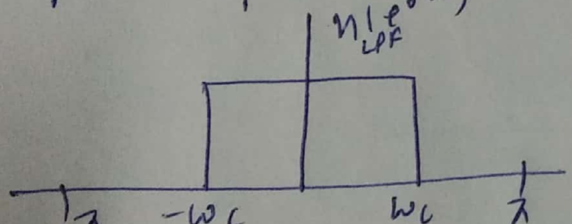
Where ω_c is any frequency ranging from 0 to π

$h_{LPF}(n) = \frac{\sin \omega_c n}{\pi n}$ This function is not valid for $n=0$

Infinite Impulse Response $h_{LPF}(0) = \frac{\omega_c}{\pi}$ $n=0$

Perfect non causal system

phase response should be zero



We are calling this as ideal filter because there is a sharp gain,

Gain can be any value b/w 0 and 1. There is only one such point of uncertainity ω_c.

This filter has zero phase response.

If zero phase response then there is no need to wait for the output.

5/11/12

To build:

$\boxed{\text{N-length}}$ FIR System (LowPass Filter with cut off w_c)

It should form a causal system

Ideal System gives us IIR and noncausal system
Now, we have to leave some information to get FIR and causal system.

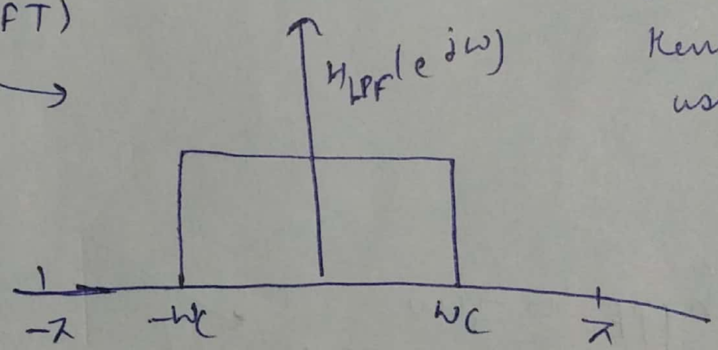
There can be infinite methods to convert IIR into FIR i.e ~~it~~ its the conversion is not unique.
We need to provide some extra information to ~~do~~ ~~so~~ make it unique.
~~Eg~~ Evaluate each method and define a cost Function,

~~of~~ ~~cost~~ cost Function → minimum (unique)
→ maximum (unique)

How to determine the cost function?

since it is a continuous function Hence we are using DTFT not DFT

Ideal LPF Filter (DTFT)



After converting it into FIR, find its DTFT coefficients and compute

$$\int_{-\pi}^{\pi} |H_{LPF}(e^{j\omega}) - \hat{H}_{LPF}(e^{j\omega})|^2 \, d\omega$$

→ error (Least Square error)

The one that gives the least solution is the best ~~va~~ solution,

→ defined in transform domain

This cost function is the difference energy function

By using Parseval's Relationship, converting the previous equation into time domain

$$\sum_{n=-\infty}^{\infty} |h_{LPF}(n) - \hat{h}_{LPF}(n)|^2$$

The least value using both will be equal

If we take $-2 \le n \le 2$ ↳ one function is defined in this range ($\hat{h}_{LPF}$)

$$SE = \underbrace{\sum_{n=-\infty}^{-3} |h_{LPF}(n) - \hat{h}_{LPF}(n)|^2}_{} + \underbrace{\sum_{n=-2}^{2} |[h_{LPF}(n) - \hat{h}_{LPF}(n)]|^2}_{} + \underbrace{\sum_{n=3}^{\infty} |(h_{LPF}(n) - \hat{h}_{LPF}(n))|^2}_{}$$

square error

other is going from $-\infty$ to ∞

$\hat{h}_{LPF}$ is zero within these ranges

Hence the cost is Fixed

This sum plays the major role

We want this sum to be zero.

↳ Best case

This has variable cost

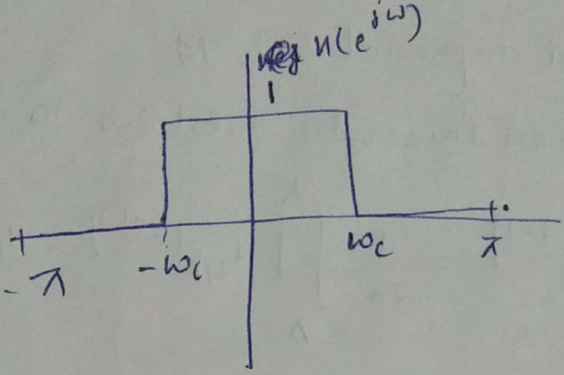
In this $-2 \le n \le 2$, the value of $\hat{h}_{LPF}$ should be equivalent to ideal Filter

Least Square Solution

$$\hat{h}_{LPF}(n) = h_{LPF}(n)$$

12/11/18

Ideal Frequency Response of LPF



$H(e^{j\omega}) = 1 \qquad -\omega_c \le \omega \le \omega_c$

$\qquad\quad = 0 \qquad$ otherwise

IDTFT $\angle H(e^{j\omega}) = 0$ $\longrightarrow$ zero phase response is most appreciated systems and hence is ideal

$h_{LPF}(n) = \dfrac{\sin \omega_c n}{\pi n} \qquad n \neq 0 \qquad = h_{LPF}(-n)$

↳ symmetric about y axis

(Infinite length)

$\qquad = \dfrac{\omega_c}{\pi} \qquad n = 0$

but we require a N length FIR system

We can calculate the error in ① Time Domain

② Frequency Domain

These two representations have one thing equal which is the energy of the signal.

Now error signal $= \hat{h}_{LPF}(n) - h_{LPF}(n)$

does not represent energy.

We need to use Parseval's relation to equate the two

$$E_{LPF}(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left| \hat{H}_{LPF}(e^{j\omega}) - H_{LPF}(e^{j\omega}) \right|^2 d\omega$$

$$e^2_{LPF}(n) = \sum_{-\infty}^{\infty} \left[\hat{h}_{LPF}(n) - h_{LPF}(n) \right]^2 \quad -①$$

Now we need to find the value of N for which e^2_{LPF} is minimum.

we can divide eq① into 3 parts

$$e^2_{LPF}(n) = \underbrace{\sum_{-\infty}^{(N/2-1)} (\ )}_{I} + \underbrace{\sum_{-N/2}^{N/2} (\ )}_{II} + \underbrace{\sum_{N/2+1}^{\infty} (\ )}_{III}$$

but $\hat{h}_{LPF}(n)$ is defined only in range → $-N/2 \le n \le N/2$

which makes I^{st} and III^{rd} term to be of fixed cost and independent of $\hat{h}_{LPF}(n)$.

Now $e^2_{LPF}(n)$ will be minimum when II^{nd} term is zero which is possible when $\boxed{\hat{h}_{LPF}(n) = h_{LPF}(n)}$

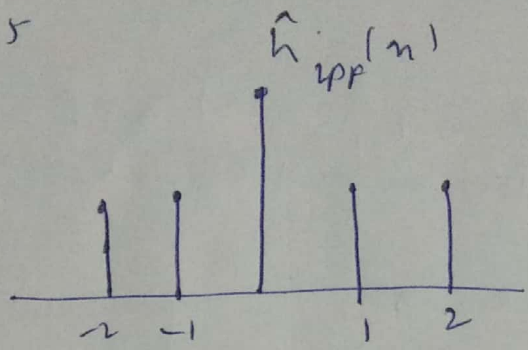
This solution is still not unique as we can take any value of N

$\underbrace{-N/2 \le n \le N/2}$

by choosing this range we as it gives us symmetry about y axis.

If we don't mention symmetricity about y axis then we will not achieve zero phase response.

Example For $N = 5$



Now, the resultant system is non-causal. (Finite non causality)
but we required a causal system.

Causal System $\rightarrow \hat{h}_{LPF}(n-2)$

phase will be changed $\underset{\sim}{e^{-j2\omega}}$

$$\hat{h}_{LPF}(n) \xleftrightarrow{DTFT} \hat{H}_{LPF}(e^{j\omega})$$

$$\hat{h}_{LPF}(n-2) \xleftrightarrow{DTFT} e^{-j2\omega}\,\hat{H}_{LPF}(e^{j\omega})$$ I st conclusion

phase will be $\underbrace{0}_{\text{Initial phase}} + \underbrace{2\omega}_{\text{new}}$ $\Big] \rightarrow$ linear phase

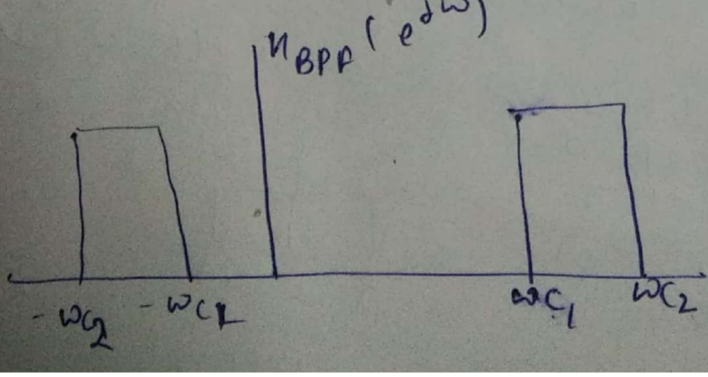
(This is implementable)

this will preserve the envelope of the signal.

II nd $\rightarrow$ Our device was best in terms of least square method

Result $\rightarrow$ Linear Phase N-length FIR lowpass filter

Derive for HPF, BPF, Band Stop filter



Scanned by CamScanner

14/11/18

Format of Project Report

i) First page should include Title & Group member details

ii) Paper

iii) Individual Understanding (comprehension) of paper

iv) Implementation — Program (with sufficient comment) — Results

v) Conclusion of Results

More weightage

---

① We for any impulse response, if it ~~was~~ is symmetric then it has a linear phase in frequency domain

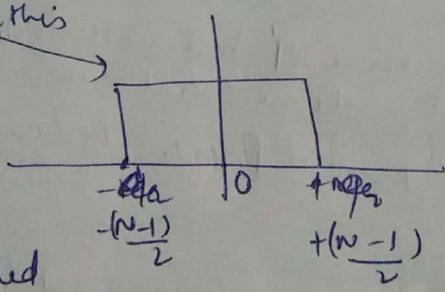
② The final frequency response of $\hat{h}_{LPF}(n)$

Two possible Symmetry → Even Symmetry → Odd Symmetry

N must be Odd For this

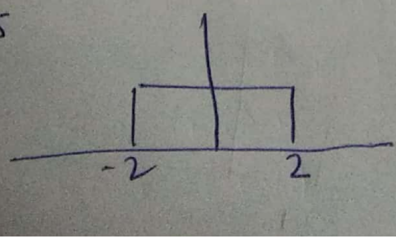
because when we plot from $-N/2$ to $N/2$

0 is also included which makes N odd.



Where N is the length of the System

For ex For $N=5$



comes through the coefficients

Length of FIR system — even / Odd

Small number

comes through roots

Order of FIR System — even / Odd

FIR system can be represented in terms of polynomial of z.

No. of coefficients > No. of roots

For n^{th} order system

No. of coefficients $\rightarrow n+1$

roots $\rightarrow n$

Two possible combination

~~System~~ Symmetry & Order

Symmetry & Length

Good way of representing

It has -1 dimension

of less

Eq

N^{th} Order FIR SYSTEM (Low pass)

ex.
(LPF, BPF, HPF)

N → even
  → odd

Symmetry → even
         → odd

4 possible combinations

Case 1: Even Order even Symmetry
Case 2: Odd Order even Symmetry
Case 3: even order Odd Symmetry
Case 4: Odd order Odd symmetry

results in → ① Non causal zero phase
           → ② Causal linear phase

Case 1: Even Order even Symmetry // (Odd length even symmetry)
N represents Order here not length

For even Symmetry

$$h(n) = h(-n)$$

N = 4
(length = 5)

$$h(n) = \{h(0), h(1), h(2), h(3), h(4)\}$$

Index & Causal System Starts from n = 0

Symmetry point

$$h(n) = h(0)\delta(n) + h(1)\delta(n-1) + h(2)\delta(n-2) + h(3)\delta(n-3) + h(4)\delta(n-4)$$

$$H(z) = h(0) + h(1)z^{-1} + h(2)z^{-2} + h(3)z^{-3} + h(4)z^{-4}$$

Frequency Response ← $H(z)\big|_{z=e^{j\omega}}$ $= h(0) + h(1)e^{-j\omega} + h(2)e^{-j2\omega} + h(3)e^{-j3\omega} + h(4)e^{-j4\omega}$

For even symmetry about h(2)

$$h(1) = h(3)$$

$$h(0) = h(4)$$

$$H(e^{j\omega}) ~~H(z)~~ = e^{-j2\omega}\left[h(0)e^{j2\omega} + h(1)e^{j\omega} + h(2) + h(3)e^{-j\omega} + h(4)e^{-j2\omega}\right]$$

$$= e^{-j2\omega}\left[h(0)e^{j2\omega} + h(1)e^{j\omega} + h(2) + h(1)e^{-j\omega} + h(0)e^{-j2\omega}\right]$$

$$= e^{-j2\omega}\left[2h(0)\left(\frac{e^{j2\omega}+e^{-j2\omega}}{2}\right) + h(1)\left(e^{j\omega}+e^{-j\omega}\right) + h(2)\right]$$

$$H(e^{j\omega}) = \underbrace{e^{-j2\omega}}_{\text{Phase part}}\underbrace{\left[2h(0)\cos 2\omega + 2h(1)\cos\omega + h(2)\right]}_{\text{magnitude part}}$$

Phase part
which has linear phase
-2ω

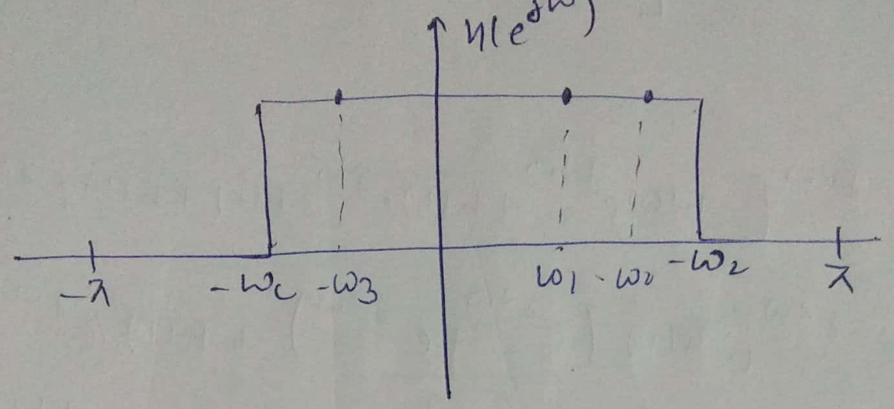
$$\boxed{H(e^{j\omega}) = e^{-j\frac{N}{2}\omega}\left[h\left(\frac{N}{2}\right) + 2\sum_{n=1}^{N/2} h\left(\frac{N}{2}-n\right)\cos n\omega\right]}$$

If we know the frequency response, by using the above equation we can find out the impulse response scale factors in time domain ($h(\frac{N}{2})$ etc)

$$H(e^{j\omega}) = e^{-j2\omega}\left[2h(0)\cos 2\omega + 2h(1)\cos\omega + h(2)\right]$$

we need to find $h(0)$, $h(1)$, $h(2)$

For a LPF, the frequency response we have is



— (1)

we have 3 unknowns ($h(0), h(1), h(2)$) , then minimum
three equations are required ,
But, we have more than 3 equations here.
which means more equations , less number of unknowns.
Hence, there can be many solutions.

$$H(z) = h(0) + h(1)z^{-1} + h(2)z^{-2} + h(1)z^{-3} + h(0)z^{-4}$$

Order of polynomial = 4

Degree of freedom = No. of coefficients = 2 ($h(0), h(1)$)

We are designing a N order polynomial with
$\frac{N}{2}$ degree of freedom.
This is the limitation of this technique.
This is the cost we are paying for getting the FIR
LPF filter linear phase.
If we remove the linear phase condition, we will get
N order polynomial with N degree of freedom
but that will increase the no. of unknowns too.

Let us assume three frequencies w_1, w_2, w_3 on Graph ①

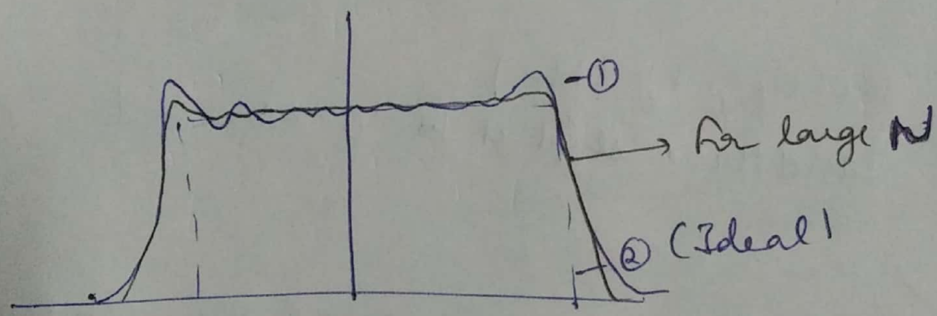
using these we will get the ~~[illegible]~~ coefficient which might not have minimum error.

In order to get minimum error, we should use least square method using all the three points available.

For least square

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} |H(e^{j\omega}) - \hat{H}(e^{j\omega})|^2 \, d\omega$$

For large N, the estimated ~~[illegible]~~ impulse response will have more coefficients matching with the ideal response, then it will be more close to ideal behaviour.



~~[illegible]~~

But even for $N \to \infty$, there will be some error, because of the discontinuity at two places.

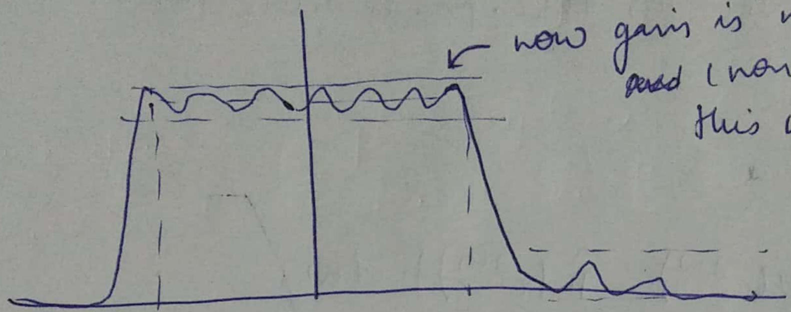
Error can never be zero.

Scanned by CamScanner

Impulse Response with odd Symmetry

① Differentiator (FM Modulation / Demodulation)

② Hilbert Transform (SSB)



now gain is not 1 (non uniform)

this will cause envelope distortion

To get a Finite Response

Use a Window

$$\begin{cases} w(n) = 1 & 0 \le n \le N-1 \\ w(n) = 1 & -N/2 \le n \le N/2 \end{cases}$$

multiply by (h(n))

and then shift it to $0 \le n \le N-1$

In Frequence domain, the one with least number of oscillation gives the best solution

19/11/18

For $N=2$

no. of coefficients $= 3$

$\{h(0), h(1), h(2)\}$

for even symmetry $h(0) = h(2)$

$$H(z) = h(0)z^{0} + h(1)z^{-1} + h(2)z^{-2}$$

$$H(e^{j\omega}) = h(0) + h(1)e^{-j\omega} + h(2)e^{-j2\omega}$$

$$= h(0) + h(1)e^{-j\omega} + h(0)e^{-j2\omega}$$

$$= h(0)(1 + e^{-j2\omega}) + h(1)e^{-j\omega}$$

$$H(z^{-1}) = h(0) + h(1)z + h(2)z^{2}$$

multiply by z^{-2}

$$z^{-2}(H(z^{-1})) = h(0)z^{-2} + h(1)z^{-1} + h(2)$$

compare this with

$h(2) = h(0)$

$$H(z) = h(0) + h(1)z^{-1} + h(0)z^{-2}$$

These two equations are equal

$$\boxed{H(z) = z^{-N}H(z^{-1})}$$

This is called ~~receiving~~ mirror Image polynomial.

$H(z) = z^{-N}H(z^{-1})$ implies if we have a zero at some z then we will also have ~~root~~ zero at some z^{-1}

Ex: $H(z_1) = 0$

z_1

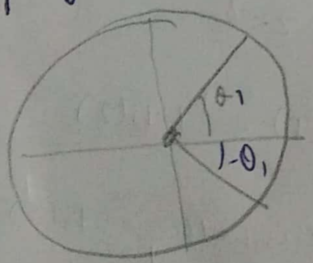
$z = z_1$ is the root of the polynomial

$$\boxed{H\left(\frac{1}{z_1}\right) = 0}$$

In order to make a linear phase systems, if we take a root z_1 as the other root. Then we should also take $\frac{1}{z_1}$

For a real polynomial → coefficients are real,
→ roots are real

For a complex polynomial complex roots should be in conjugate pair so that we get real coefficients.



This will make the ~~coeffien~~ polynomial real.

For a complex zero (root), we have total 4 zero

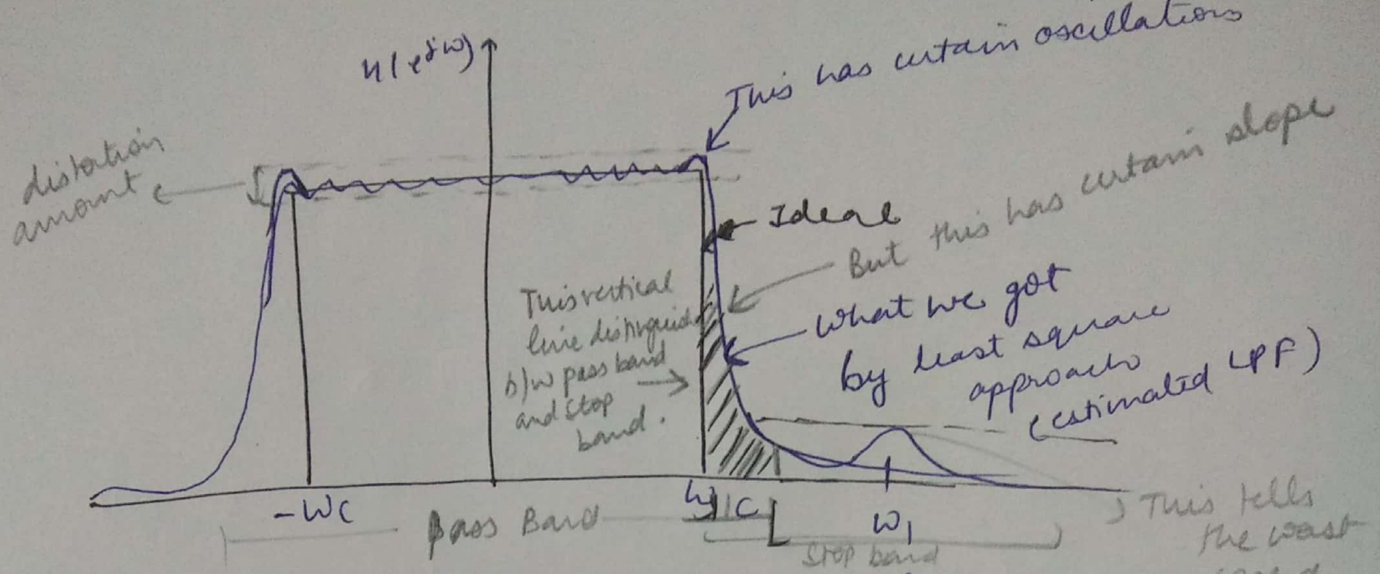
complex zero $(\alpha + \beta i)$ conjugate $(\alpha - \beta i)$

For symmetry and linear phase $\frac{1}{\alpha + \beta i}$

For symmetry and linear phase $\frac{1}{\alpha - \beta i}$

There, 4 ~~roots~~ zeros

Ideal Lowpass filter



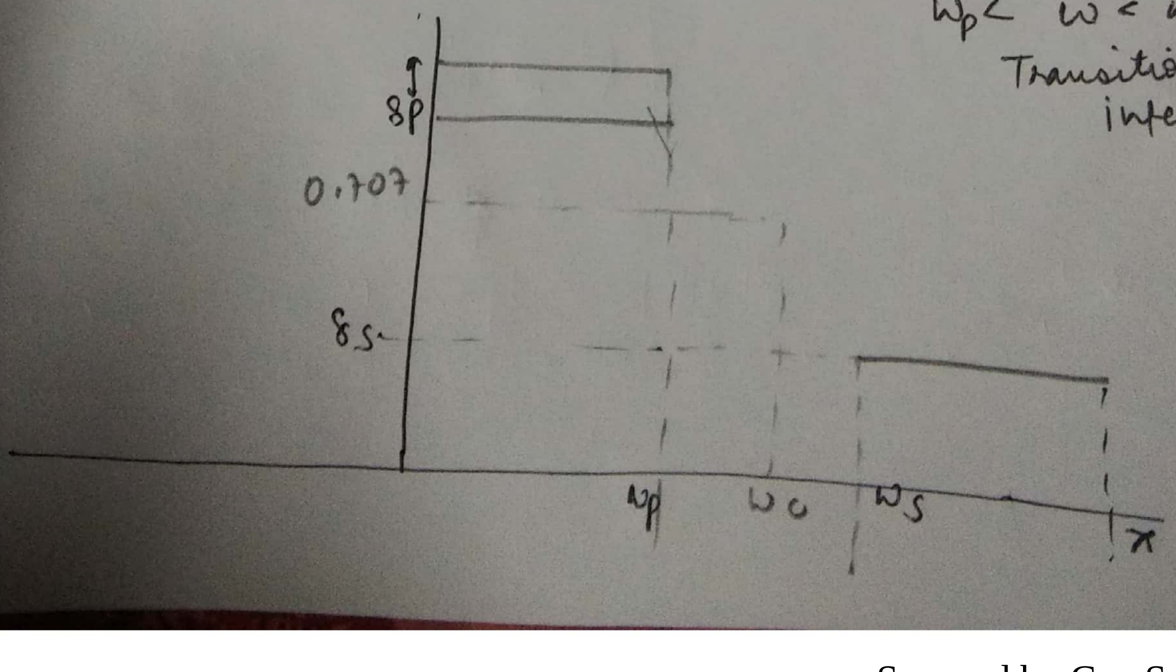
We will not be able to eliminate ω_1 frequency using our estimated LPF.

Highest gain ~~keep~~ is at the point of discontinuity

Highest gain in the ~~stop~~ band should be reduced to zero.

Highest gain in the pass band should be reduced to 1.

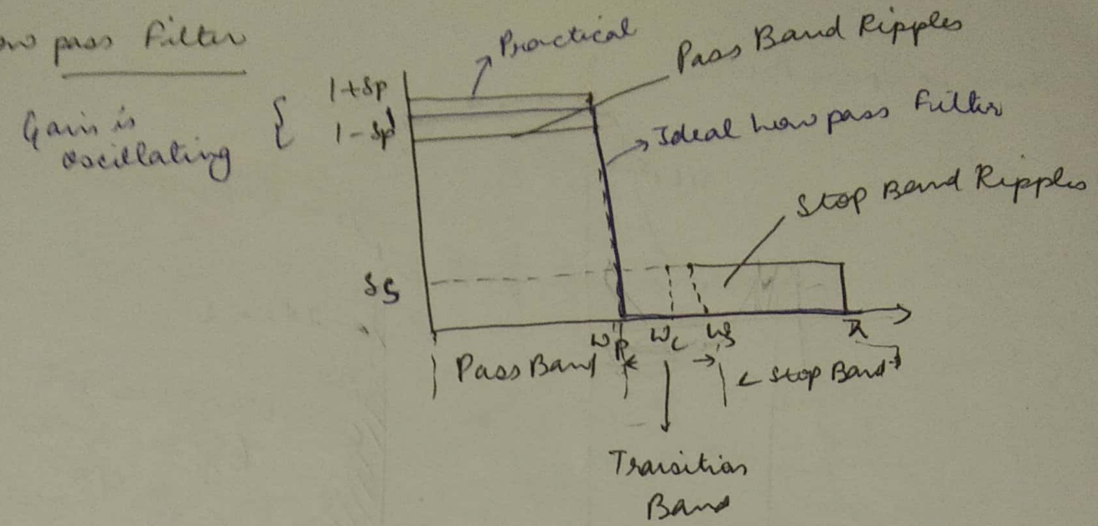
region marked like this → This region neither belongs to pass band nor stop band and is called transition region.



$\omega_p < \omega < \omega_s$ Transition interval

21/11/18

FIR Low pass Filter



If input signal ~~is~~ ~~dellow~~ has Frequencies in the range

$$w_p \le w \le w_s$$

then 1. For ideal filter

the ~~ol~~ signal will pass from w_p to w_c (w_c included)
and will be blocked from w_c to w_s (~~not~~ w_c not included)

uncertainty at w_c due to discontinuity

2. For practical filter

We can not define the value of input signal
for $\{w_p \le w \le w_s\}$ range

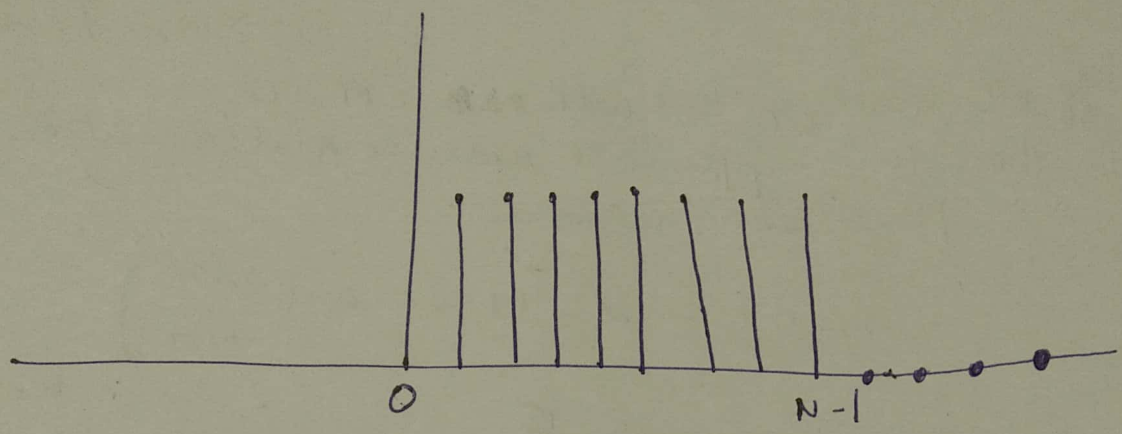
Now we want $\delta p \to 0$, $\delta s \to 0$
and $[w_p, w_s]$ to be just a single point,
or w_p, w_s as close as possible

If we increase the length/ of the systems, order
then the amplitude of oscillations due to δp ~~accumulate~~ ~~area~~ will decrease
~~the~~ except δp at discontinuity

For large, $\delta p \ne 0$
However if we make $\delta p = 0$, $\delta s = 0$ there will be
oscillation at discontinuity

But as $N \to \infty$ the transition region will be reduced to W_c

Fourier transform ~~would~~ is not ~~be~~ able to define the system at discontinuity. This is the limitation of Fourier Transform



Two conditions → Decrease Ripple at cost of Transition period
(Keeping N constant)

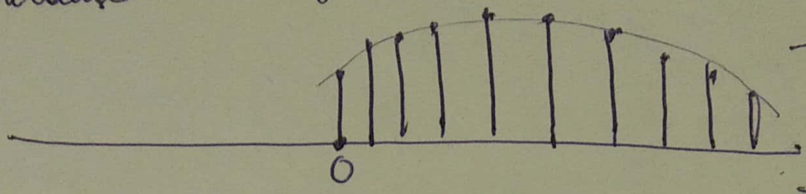
Decreasing Transition period with increasing ripples

Two Possibilities → Fixed window
→ Variable (adjustable) window

These two conditions belong to this

Hamming

Generalised Hamming window



→ Smooth Function
→ no discontinuity
→ sinc function

If we multiply this with our LPFilter then the ripples will reduce and transition period will increase

$$w(n) = \alpha - (1-\alpha)\cos\left(\frac{2\pi n}{N+1}\right)$$

$$= 0 \qquad n > N$$

When we say design a Filter for $\omega_c = 4KHz$ we should take $\omega_p = 4KHz$ not $\omega_c = 4KHz$

Decimation in Time FFT

Assumption $N = 2^x$

Is there any other possible way of taking N? power of any other number

Let us take one N-length sequence

where N is a power of 2

$$X(K) = \sum_{n=0}^{N-1} x(n) W_N^{nK} \quad \text{DFT eq.}$$

$$= \sum_{n=0}^{N/2-1} x(n) W_N^{nK} + \sum_{n=N/2}^{N-1} x(n) W_N^{nK}$$

$$= \sum_{n=0}^{N/2-1} x(n) W_N^{nK} + \sum_{n=N/2}^{N-1} x(n+N/2) W_N^{(n+N/2)K}$$

Rough

$m = n - N/2$

$n = m + \frac{N}{2}$

m

$$W_N^{(n+N/2)K} = W_N^{nK} \cdot W_N^{N/2 K}$$

$$W_N = e^{-j\left(\frac{\cancel{2}\pi}{\cancel{N}}\right) \times \frac{\cancel{N}}{\cancel{2}} K}$$

$$= \left(e^{-j\pi}\right)^K$$

$$= (-1)^K$$

$$= \sum_{n=0}^{N/2-1} x(n) W_N^{nK} + (-1)^K \sum_{n=0}^{N/2-1} x(n+N/2) W_N^{nK}$$

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

Scanned by CamScanner

For even 'k'

$$X(2r) = \sum_{n=0}^{N/2-1} [x(n) + x(n+N/2)] W_N^{2rn} = \boxed{\sum_{n=0}^{N-1} [x(n) + x(n+N/2)] W_{N/2}^{rn}}$$

$$X(2r+1) = \sum_{n=0}^{N/2-1} [x(n) - x(n+N/2)] W_N^{(2r+1)n}$$

$$= \sum_{n=0}^{N/2-1} [x(n) - x(n+N/2)] W_N^{2rn} \cdot W_N^{n}$$

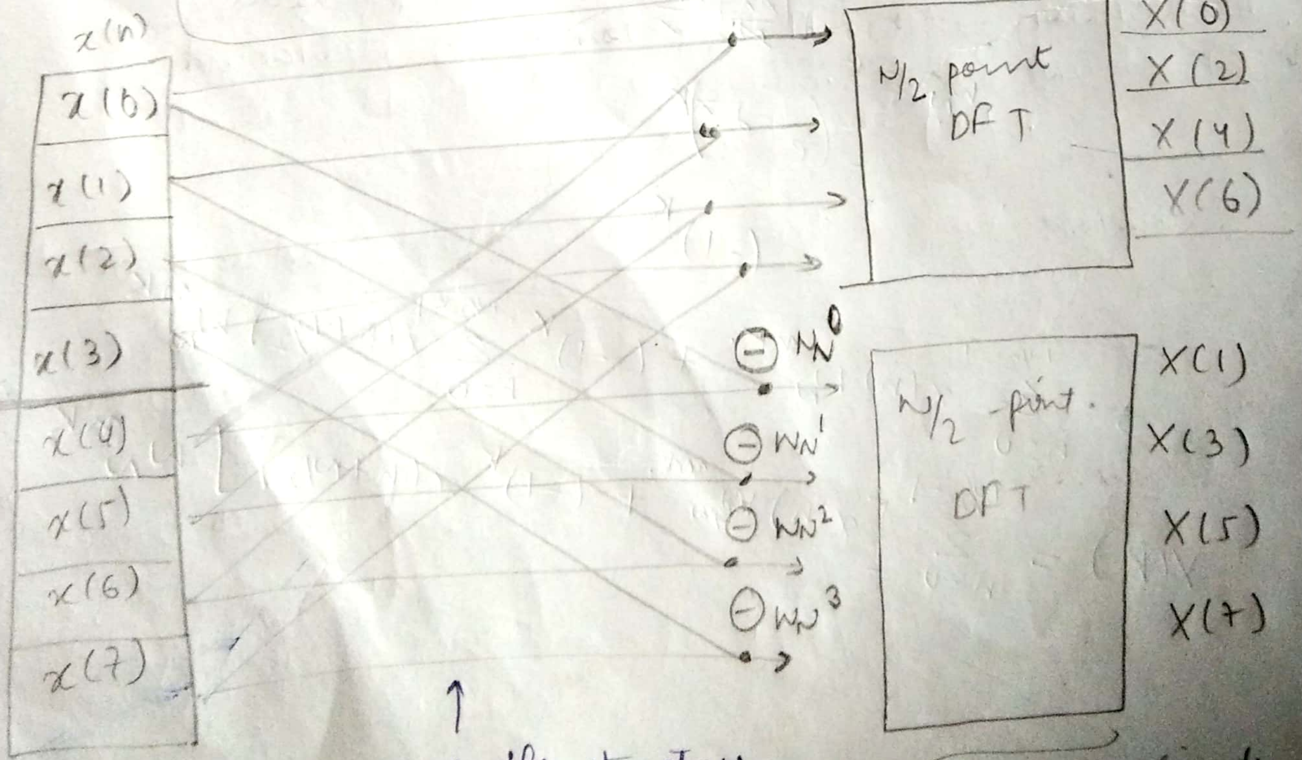
$$= \sum_{n=0}^{N/2-1} \left[[x(n) - x(n+N/2)] W_{N/2}^{rn} \cdot W_N^{n} \right.$$

DIF merging equations

$e^{-j\frac{2\pi}{N}n}$

$\rightarrow$ N/2 point DFT of $[x(n) + x(n+N/2)]$

$$\boxed{X(2r+1) = \sum_{n=0}^{N/2-1} \left[(x(n) - x(n+N/2))\, W_N^{n} \right] W_{N/2}^{rn}}$$



Butterfly structure

(This structure is reversed version of previous example)

We can again divide this into 4 $\left(\frac{N}{4}\right)$ point DFTs

The complexity is same as previous one, only the structure has changed.

Here we are decimating in 'K' (Frequency)

$\underline{N=8}$:

We will get

$X(0)$

$X(4)$

$X(2)$

$X(6)$

$X(1)$

$X(5)$

$X(3)$

$X(7)$

Bit order reversing:

Write in 3 bit Format $(2^3=8)$

$x(0)$	$x(000)$
$x(1)$	$x(001)$
$x(2)$	$x(010)$
$x(3)$	$x(011)$
$x(4)$	$x(100)$
$x(5)$	$x(101)$
$x(6)$	$x(110)$
$x(7)$	$x(111)$

reversing the order we get

$x(000)$	$x(0)$
$x(100)$	$x(4)$
$x(010)$	$x(2)$
$x(110)$	$x(6)$
$x(001)$	$x(1)$
$x(101)$	$x(5)$
$x(011)$	$x(3)$
$x(111)$	$x(7)$

(Desired)