Unit 5

Quantization Errors in DSP

FINITE WORDLENGTH EFFECTS

- Registers are basic storage device in digital system. The maximum size of the binary information that can be stored in a register is called Register word length.
- For storing the input data in registers they have to be quantized, to be coded in binary and it depends on register word length.
- This quantization and coding will introduce error in the input data as the analog data (infinite precision) is converted into digital data (Finite Precision).
- While performing computation, the results may exceed the size of the register used for storing the result. Then results should be truncated or rounded off to accommodate which makes the system non-linear and leads to limit cycle behaviour. The effects of Truncation & rounding can be represented in terms of additive error signal, which is called as round-off noise.

FINITE WORDLENGTH EFFECTS

Effects due to finite precision representation of number in a digital system are commonly referred to as Finite Word Length effects. Some of the finite word length effects in digital filters are

- Errors due to quantization of input data by A/D converter
- Errors due to quantization of filter coefficients.
- Errors due to rounding the product in multiplication.
- Errors due to overflow in addition.
- Limit Cycles.

* Digital Signal Processing algorithms are mainly used in Digital systems like digital computers.
* In Digital systems, the numbers and coefficients are stored in finite length registers.
* Therefore, the numbers are quantized by
    * truncation (or)
    * rounding off

Coefficient Quantization error

Truncation- Simply cutoff the remaining bits

0.5485

Rounding off -0.7878

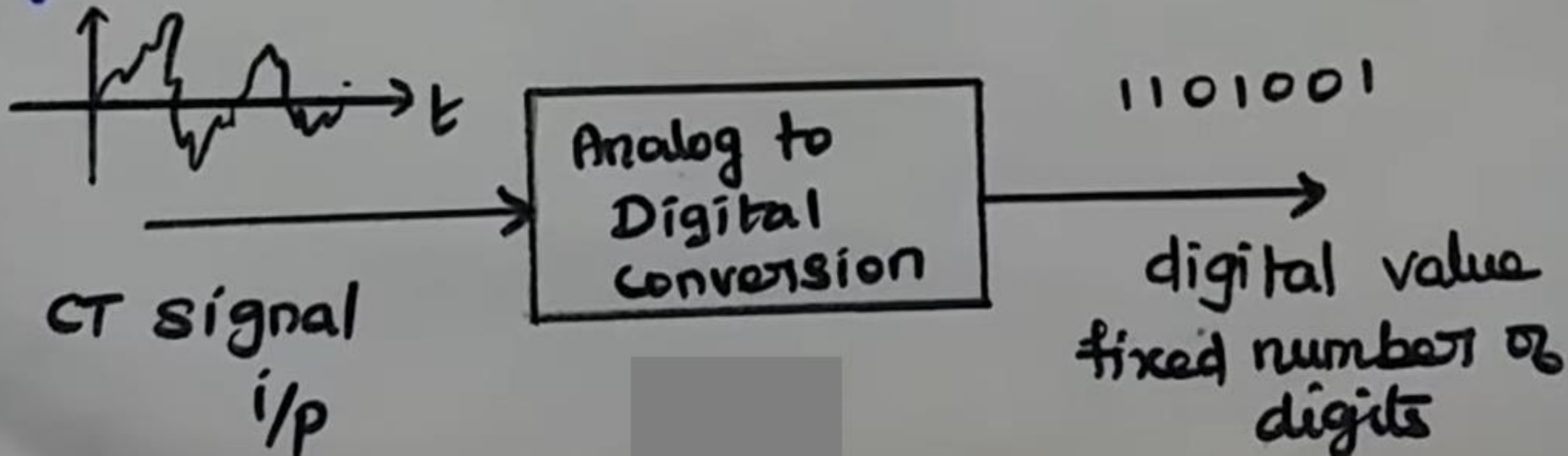
-0.788

Effects / Errors due to quantization :-

* Input quantization error
* Product quantization error
* Coefficient quantization error
* Round off noise &
* Limit cycle oscillations

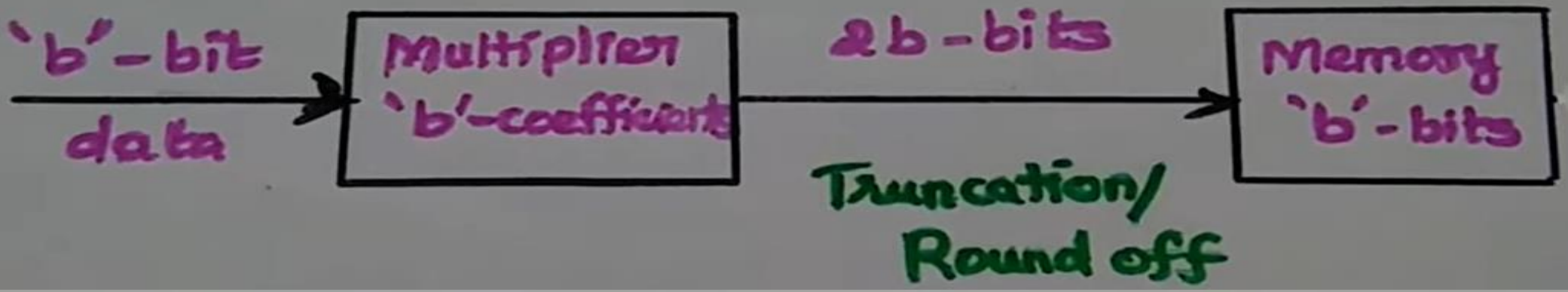
Input quantization error :-

→ Input quantization error occurs when the continuous time signal is converted into digital value in A/D conversion process.



* Product Quantization Error

→ Product Quantization error occurs at the output of the multiplier.



→ Since 'b'-bit register is used, the multiplier output must be rounded (or) truncated to 'b' bits.

→ It produces a product error.

* Coefficient Quantization Error :-

→ Coefficient quantization error arises due to the process of quantizing the filter coefficients.

→ If the filter coefficients are quantized, the frequency response of the resulting filter may differ from the desired response.

* Limit cycle oscillations :-

→ It is a low level oscillation which arises due to the non-linearity associated with rounding off the internal filter coefficients

* Zero input limit cycle oscillation
* Overflow limit cycle oscillation

Number Representation

→ In DSP, a number 'N' can be represented to any desired format using Number system.

→ To represent the numbers in any digital hardware.

Types of Number Representation:-

(i) Fixed Point Representation

* Sign magnitude form
* 1's complement form
* 2's Complement form

(ii) Floating Point Representation

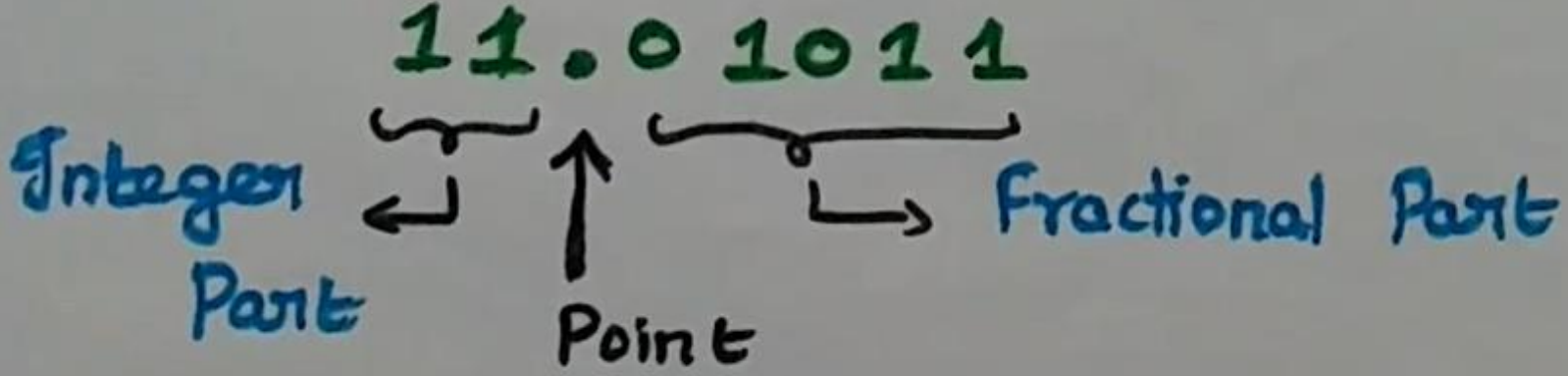
(iii) Block Floating Point Representation

Fixed Point Representation :-

→ The position of the binary point is fixed.

Ex:-

In Binary,



In Decimal,

3.34375

Representation of Negative Numbers in Fixed-point Arithmetic

* Sign - Magnitude form
* One's Complement form
* Two's Complement form

Sign-Magnitude form :-

→ The most significant bit (MSB) is set to '1' → to represent the negative sign.

'0' → to represent the positive sign

Ex :-

$-1.25 \Rightarrow 11.01$

$+1.25 \Rightarrow 01.01$

One's Complement form :-

→ In this method, the negative number is obtained by complementing all the bits.

Ex :-

$(0.875)_{10} \Rightarrow (0.1110000)_2$

$(-0.875)_{10} \Rightarrow (1.0001111)_2$

Two's Complement form :-

→ The negative number can be obtained by complementing all the bits and adding one to the least significant bit.

Ex :-

$(0.875)_{10} \Rightarrow (0.111000)_2$ complementing each bit

$\downarrow\downarrow\downarrow\downarrow\downarrow\downarrow\downarrow$

1.000111

$1+$ Adding one bit

$(-0.875)_{10} \Rightarrow 1.001000$

* Floating Point Representation :-

→ A positive number is represented as,

$$F = 2^C . M$$

Where,

M → Mantissa → $\frac{1}{2} \leq M < 1$

C → Exponent → either positive or negative

Ex :-

$$2.25 = 2^2 \times 0.25$$

$$F = 2^{010} \times 0.01$$

$$0.125 = 2^0 \times 0.001$$

$$F = 2^{000} \times 0.001$$

→ Negative floating point number is represented by considering the mantissa as a fixed point number.

→ The first bit of mantissa represents the sign of the floating point number.

* Block floating point representation :-

→ the combination of fixed point and floating point representations.

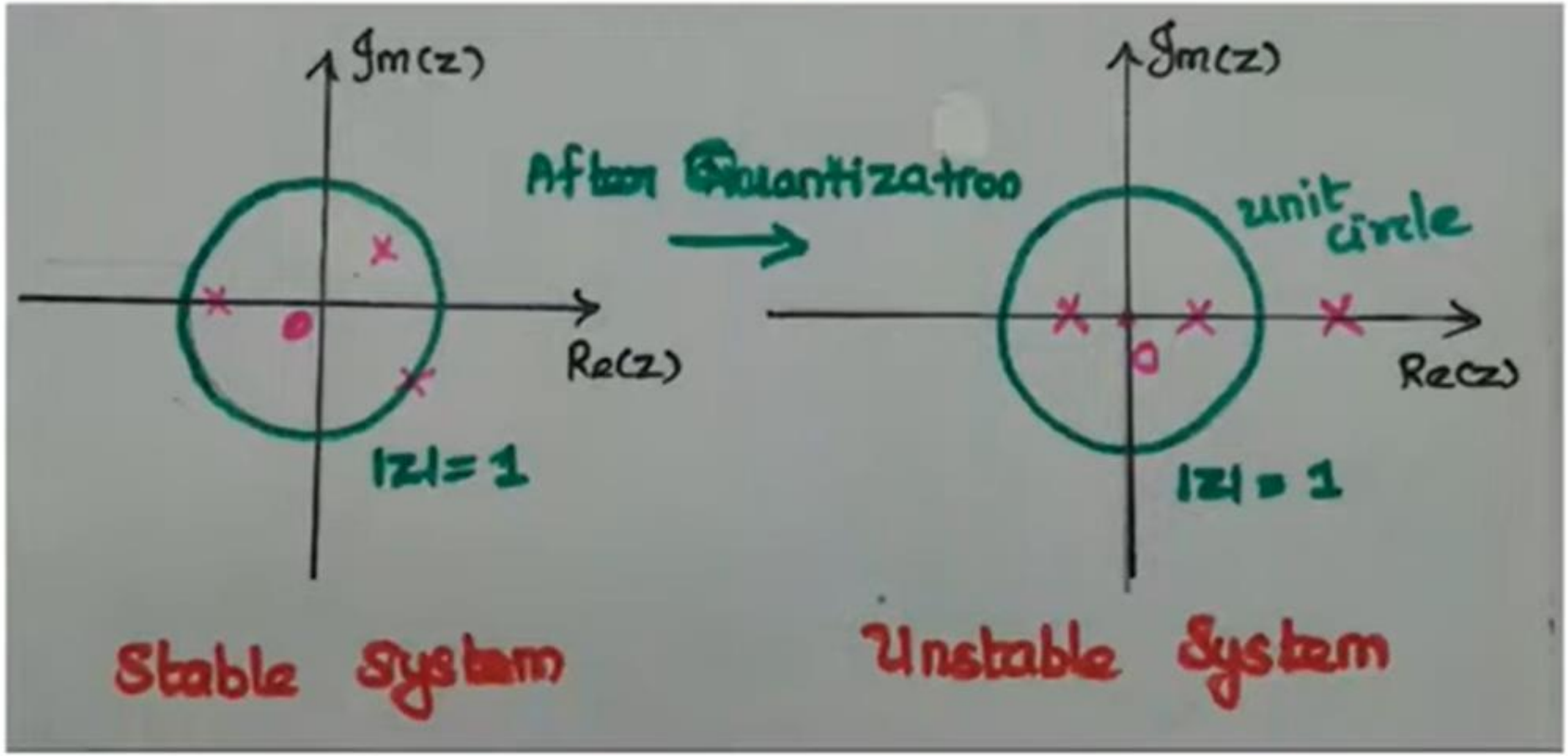
→ The set of signals is divided into blocks.

→ The arithmetic operations with in the block uses fixed point arithmetic and only one exponent per block is stored.

Fixed Point Representation	Floating Point Representation
* Fast & Inexpensive Implementation	Slow & Expensive Implementation.
* Limited Dynamic range	Large Dynamic range
* Round off errors occur only for addition	Round off errors can occur with both addition & multiplication

* Overflow occurs in addition process	Overflow does not occur.
* Low power consumption	High power consumption
* Less flexible	More flexible

Coefficient Quantization



It changes the pole locations

* Explanation :-

→ Theoritically, the filter coefficients are computed to infinite precision.

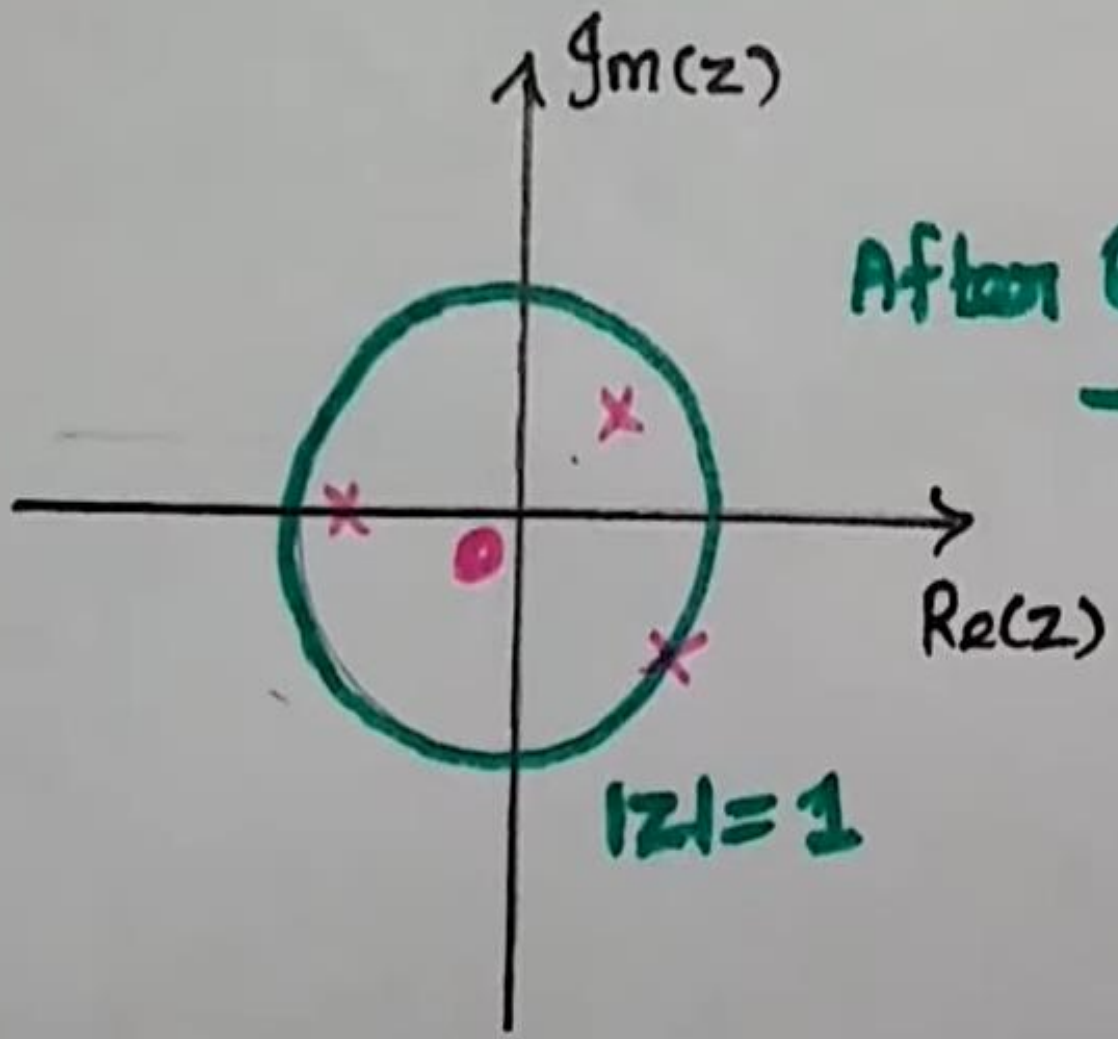
→ But in digital system, the filter coefficients are represented in binary and are stored in registers.

→ Hence the coefficients must be quantized using rounding / truncation method

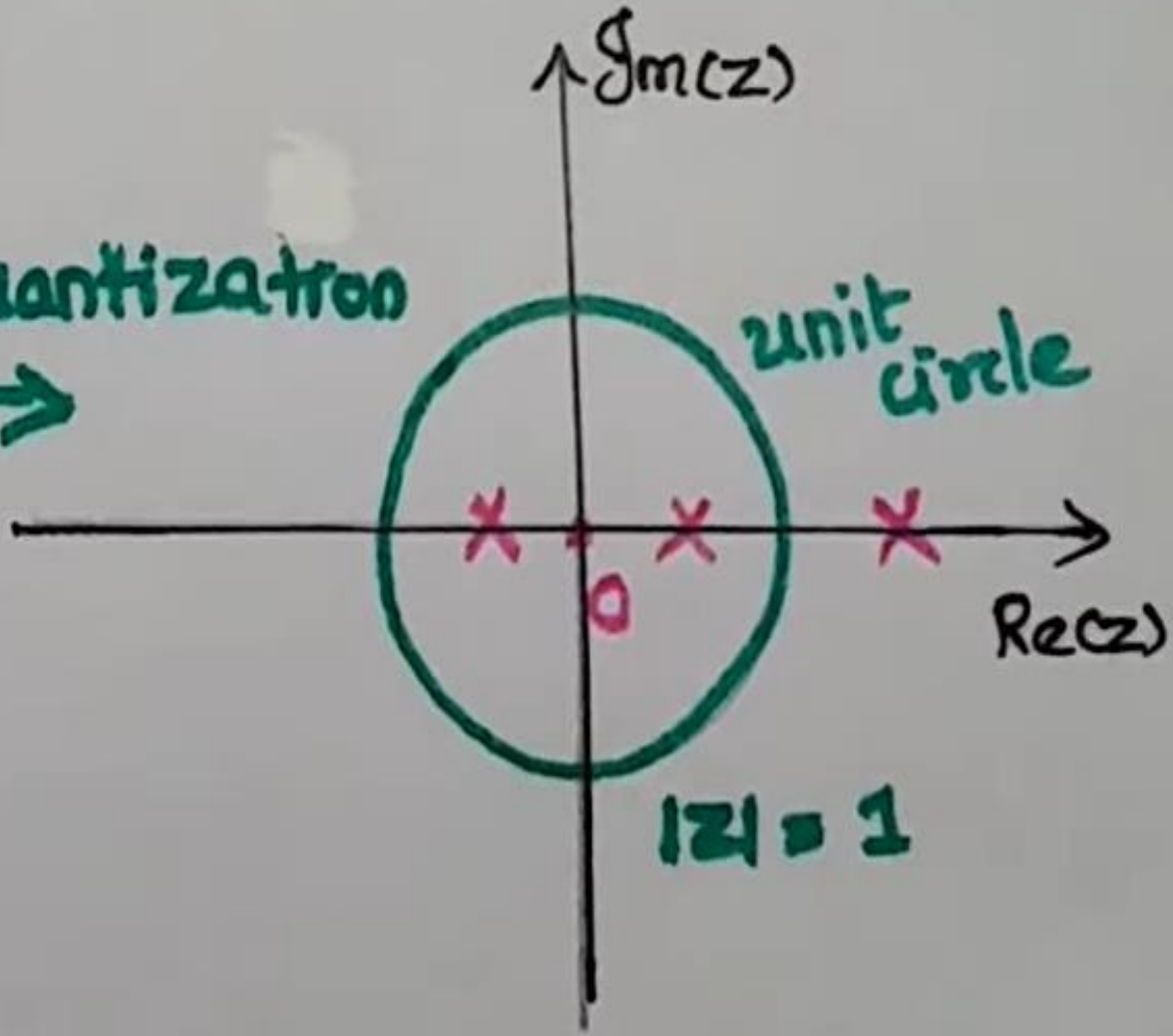
* Effects of coefficient Quantization :-

→ Due to the coefficient quantization, the frequency response of the filter deviates from the desired response of the system.

→ Filter may fail to meet the desired specifications.



After Quantization



Stable System

Unstable System

* For stable system, poles are inside the unit circle.

* Quantized coefficients may lie outside the unit circle of the z-plane leading to unstability.

* Method used to reduce coefficient Quantization

→ A high-order filter as a cascade of second-order sections.

Pbm:- Find the effect of coefficient quantization on pole locations for the given second order IIR filter using Direct form-I

$$H(z) = \frac{1}{(1-0.5z^{-1})(1-0.45z^{-1})}$$

Solution:-

Take 3 bits for truncation.

$$H(z) = \frac{1}{(1-0.5z^{-1})(1-0.45z^{-1})}$$

$$= \frac{1}{1-0.45z^{-1}-0.5z^{-1}+0.225z^{-2}}$$

$$H(z) = \frac{1}{1-0.95z^{-1}+0.225z^{-2}}$$

Coefficient Quantization:-

$$H(z) = \frac{1}{1 - Q[0.95z^{-1}] + Q[0.225z^{-2}]}$$

Q[0.95] :-

0.95

2

.90 — 1

2

.80 — 1

2

1.60 — 1

2

1.20 → 1

$Q[0.95]_{10} \Rightarrow Q[0.1111..]_2$

$\Rightarrow (0.111)_2$

$$Q[0.95]_{10} = (0.875)_{10}$$

Q [0.225]$_{10}$

0.225

2

0.450 → 0

2

0.900 → 0

2

1.800 → 1

2

1.600 → 1

2

1.200 → 1

Q [0.225]$_{10}$

⇒ [.00111...]$_2$

⇒ [0.001]$_2$

$\frac{1}{8} = 0.125$

Q [0.225]$_{10}$ ⇒ $(0.001)_2$

$= (0.125)_{10}$

$Q[0.225]_{10} \Rightarrow (0.125)_{10}$

$$H(z) = \frac{1}{1 - 0.875 z^{-1} + 0.125 z^{-2}}$$

Pole locations are changed to.

$0.95 \rightarrow 0.875$

$0.225 \rightarrow 0.125$

Quantization of filter coefficients in cascade method

$$H(z) = \frac{1}{(1-0.5z^{-1})(1-0.45z^{-1})}$$

Solution:-

No of bits = 3 ; Truncation method.

Q(0.5) :-

$$Q[0.5]_{10} = Q[0.100]_2$$

$$0.5 \times 2 = 1.0 \rightarrow 1$$

$$\boxed{Q[0.5] = 0.5}$$

Q [0.45] :-

0.45

$\times 2$

$0.90 \rightarrow 0$

$\times 2$

$1.80 \rightarrow 1$

$\times 2$

$1.60 \rightarrow 1$

$\times 2$

$1.20 \rightarrow 1$

$$Q[0.45]_{10} = Q[0.0111\ldots]_2$$

$$= (0.0\dot{1}1)_2$$

$$\Rightarrow 0.\left(0+\frac{1}{4}+\frac{1}{8}\right)$$

$$\Rightarrow 0.\left(\frac{3}{8}\right)$$

$$\boxed{Q[0.45] = 0.375}$$

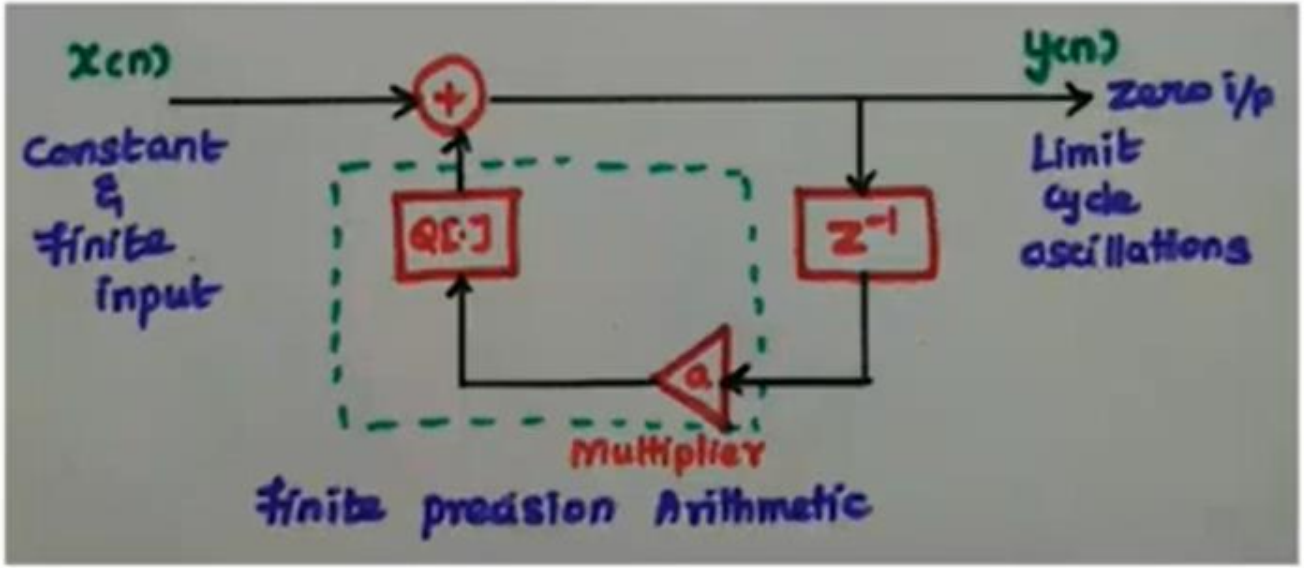
Pole locations are changed from

0.5- 0.5

0.45-0.375

Limit Cycle Oscillations

* Zero - Input Limit Cycle Oscillations Due to Product Quantization



$$\text{Dead Band} = \frac{\frac{1}{2}\, 2^{-b}}{1-|a|}$$

Limit Cycle Oscillations

* Defn:-

→ It is a low-level oscillation due to the product quantization effects in a stable IIR digital filters [Recursive filters]

* Zero - Input Limit cycle oscillation :-

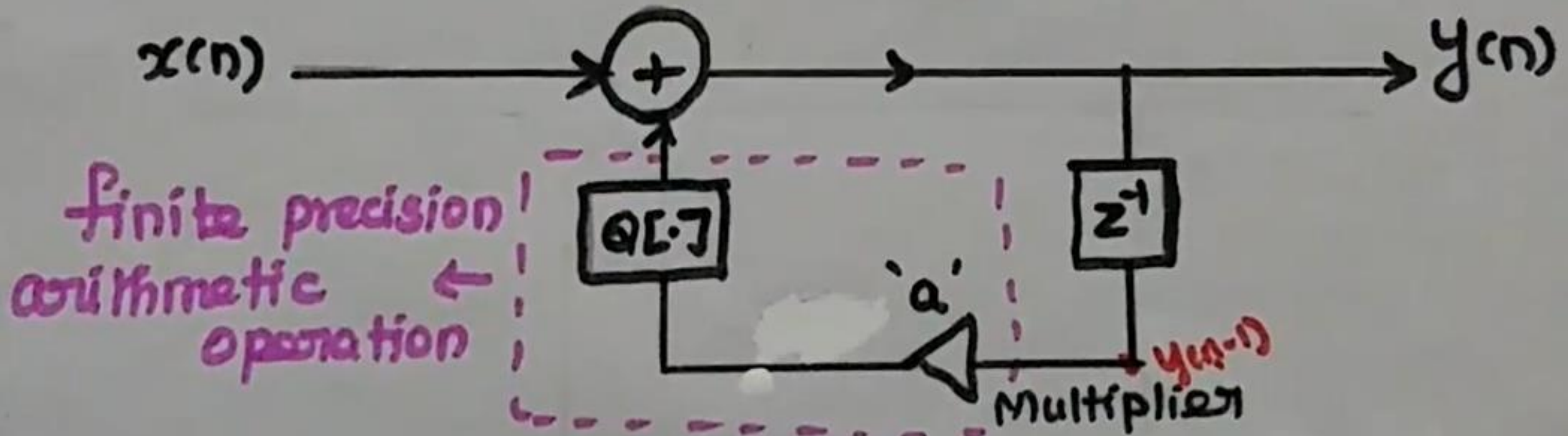
→ When a stable IIR filter is excited by a constant and finite input signal, the output must decay to zero.

→ But it does not occur due to the product quantization process which produces the non-linearities in the system operation.

* Ex:

Consider a first order IIR filter,

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



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

* The non-linearities due to this finite precision arithmetic operations cause Periodic oscillations in the output

→ Such oscillations in recursive systems are called zero-input limit cycle osillations.

Analysis of Limit cycle oscillations:-

[Important Problem in Limit cycle oscillation]

The first order recursive filter is given as,

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

$$\boxed{y(n) = x(n) + Q[a\,y(n-1)]}$$

where,

$a = \frac{1}{2^3}$ and the input signal is,

$$x(n) = \begin{cases} 0.875 & ; n = 0 \\ 0 & ; \text{otherwise} \end{cases}$$

Assume that the register size is $b = 3+1$

$b \Rightarrow$ 3 bits + 1 sign bit

$n \Rightarrow$ no. of samples in the sequence

n=0 :-

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

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

$$y(0) = 0.875 + 0$$

$$\boxed{y(0) = 0.875}$$

* n = 1 :-

$$y(1) = x(1) + Q\left[\frac{1}{2} \times y(1-1)\right]$$

$$= 0 + Q\left[\frac{1}{2} y(0)\right]$$

$$y(1) = Q\left[\frac{1}{2} \times 0.875\right] = Q[0.4375]$$

0.4375

× 2

0.8750 → 0

× 2

1.7500 → 1

× 2

1.5000 → 1

× 2

1.0000 → 1

$= Q[0.0111]$

[Round off to 3 bits]

$\Rightarrow$ 0.011 + 1 (10)

0.100

$$y(1) = Q[0.0111] = (0.100)_2$$

$$\boxed{y(1) = 0.5}$$

$0.100 \downarrow 1 \times \frac{1}{2} \Rightarrow 0.5$

* n = 2 :-

$$y(2) = x(2) + Q\left[\frac{1}{2} \times y(2-1)\right]$$

$$= 0 + Q\left[\frac{1}{2} \times y(1)\right] = Q\left[\frac{1}{2} \times 0.5\right]$$

$$y(2) = Q[0.25]$$

$$= Q[0.010]$$

0.25

2 | 0.50 → 0

2 | 1.00 → 1

$$y(2) = (0.010)_2 = 0.25$$

$$\boxed{y(2) = 0.25}$$

* n = 3 :-

$$y(3) = x(3) + Q\left[\tfrac{1}{2}\, y(3-1)\right]$$

$$= 0 + Q\left[\tfrac{1}{2} \times 0.25\right]$$

$$= Q[0.125] = Q[0.001]$$

0.125

2

0.250 → 0

2

0.500 → 0

2

1.000 → 1

$$y(3) = (0.001)_2$$

$$\boxed{y(3) = 0.125}$$

*n = 4 :-

$$y(4) = 0 + Q\left[\frac{1}{2}\, y(3)\right]$$

$$= Q\left[\frac{1}{2} \times 0.125\right]$$

$$= Q\left[0.0625\right]$$

0.0625	
2	
0.1250	→ 0
2	
0.2500	→ 0
2	
0.5000	→ 0
2	
1.0000	→ 1

$y(4) = Q[0.0625]$

$= Q[0.0001]$

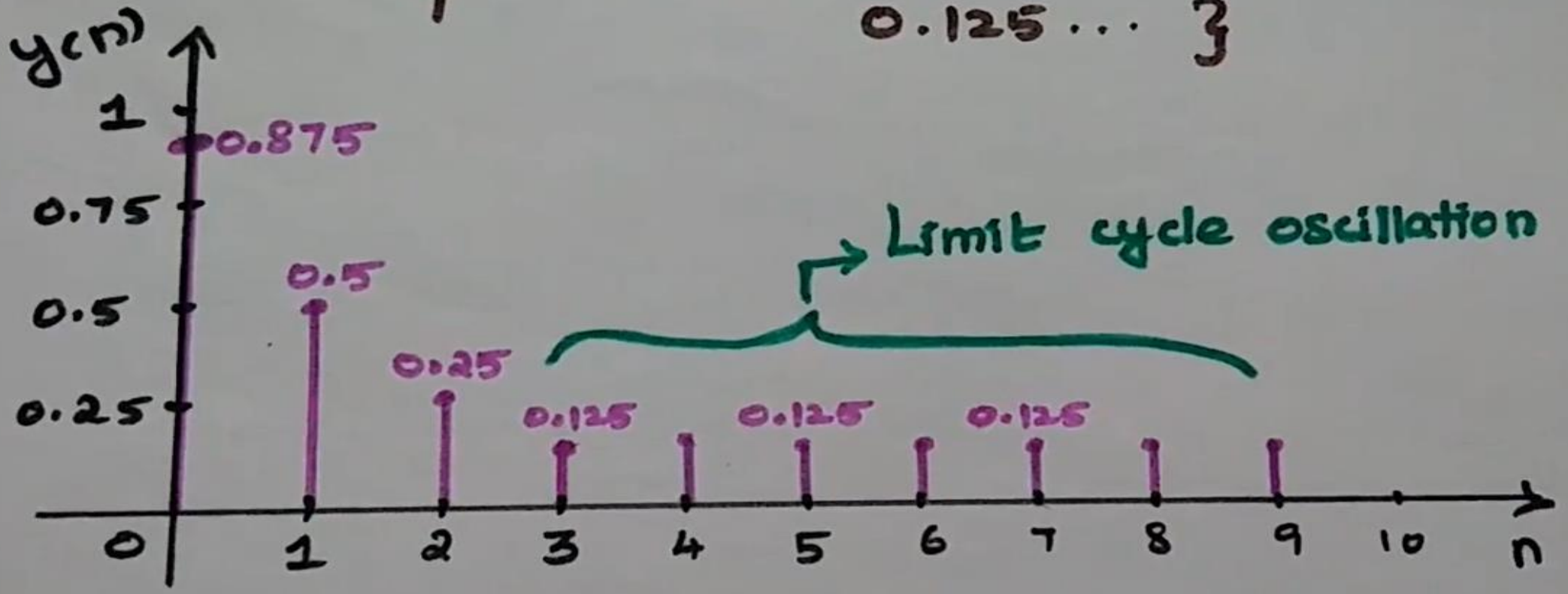
$0.0001 +$

$\Rightarrow \quad 0.0001$

$y(4) = (0.0001)_2$

$$\boxed{y(4) = 0.125}$$

$y(n) = \{ 0.875, 0.5, 0.25, 0.125, 0.125, 0.125 \ldots \}$



When $n \geq 4$, the value of $y(n)$ is equal to 0.125. This is known as limit cycle oscillations.

* Dead Band :-

→ The limit cycles occur as a result of the quantization effects in multiplications.

→ The amplitudes of the output during a limit cycle are confined to a range of values.

→ This range of values is known as dead band.

* For the first order filter, the dead band is given as,

$$\text{Dead Band} = \frac{\frac{1}{2} 2^{-b}}{1 - |a|}$$

Note :-

→ Limit cycle oscillation does not exist in FIR filters since there is no feedback network in the structure.

Zero - Input
Limit Cycle Oscillation

An Important Problem solved

$$y(n) = x(n) + 0.95\, y(n-1)$$

Prob Find the characteristics of a limit cycle oscillation of the given first order digital system.

$$y(n) = 0.95\, y(n-1) + x(n)$$

where,

$$x(n) = \begin{cases} 0.875 & ; n = 0 \\ 0 & ; \text{otherwise} \end{cases}$$

Also determine the dead band of the filter.

Solution:-

Given:

$$a = \alpha = 0.95 \ ;$$

$$x(0) = 0.875$$

Assume that the number of bits:

b ⇒ 4 bits + 1 sign bit.

The output with rounding value is given as,

$$y(n) = x(n) + Q\,[0.95\,y(n-1)]$$

n=0 :-

$$y(0) = x(0) + Q\,[0.95\,y(0-1)]$$

$$= 0.875 + Q\,[0]$$

$$\boxed{y(0) = 0.875}$$

* **n = 1 :-**

$$y(1) = x(1) + Q[0.95\,y(1-1)]$$

$$= 0 + Q[0.95\,y(0)]$$

$$= Q[0.95 \times 0.875]$$

$$y(1) = Q[0.83125]$$

$$\Rightarrow Q[0.11010\ldots]_2$$

$\Rightarrow Q[0.\underline{11010}..]_2$

0.83125

2 | 0.83125 → 1

2 | 1.66250 → 1

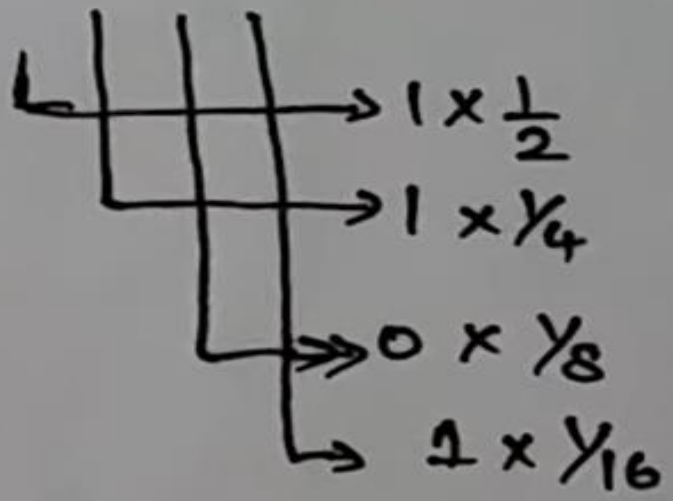
2 | 1.32500 → 0

2 | 0.65000 → 0

2 | 1.30000 → 1

2 | 0.60000 → 0

$y(1) \Rightarrow (0.1101)_2$



$\Rightarrow \frac{1}{2} + \frac{1}{4} + \frac{1}{16} \Rightarrow \frac{8+4+1}{16} = \frac{13}{16}$

$y(1) = Q[0.11010..] = (0.1101)_2$

$y(1) = 0.8125$

* n = 2 :-

$$y(2) = x(2) + Q[0.95\, y(2-1)]$$

$$= 0 + Q[0.95 \times 0.8125]$$

$$= Q[0.771875]_{10}$$

$$= Q[0.110001\ldots]_2$$

$$y(2) = Q[0.110001]_2$$

$$y(2) = (0.1100)_2 = (0.75)_{10}$$

$$\boxed{y(2) = 0.75}$$

* n=3 :-

$$y(3) = x(3) + [0.95\, y(3-1)]$$

$$= 0 + Q[0.95 \times 0.75]$$

$$= Q[0.7125]_{10}$$

$$= Q[0.101101..]_2 \Rightarrow (0.1011)_2$$

$$y(3) = (0.1011)_2 = (0.6875)_{10}$$

$$\boxed{y(3) = 0.6875}$$

* n=4 :-

$$y(4) = x(4) + Q[0.95\, y(4-1)]$$

$$= 0 + Q[0.95 \times 0.6875]$$

$$= Q[0.653125]_{10}$$

$$= Q[0.1010010..]_2 = [0.1010]_2$$

$$y(4) = (0.1010)_2 = (0.625)_{10}$$

$$\boxed{y(4) = 0.625}$$

* n = 5 :-

$$y(5) = x(5) + Q[0.95\, y(5-1)]$$

$$= 0 + Q[0.95 \times 0.625]$$

$$= Q[0.59375]_{10}$$

$$= Q[0.100011]_2 = [0.1010]_2$$

$$0.1001 + \frac{1}{1010}$$

$$y(5) = (0.1010)_2 = (0.625)_{10}$$

$$\boxed{y(5) = 0.625}$$

$y(n) = \{0.875, 0.8125, 0.75, 0.6875,$

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

$0.625, 0.625, 0.625 \ldots\}$

$y(4)$ $y(5)$ . . . . .

For $n \geq 5$, the output remains constant at 0.625 causing limit cycle oscillations.

Dead Band :-

$$\text{Dead band} = \frac{\frac{1}{2} 2^{-b}}{1 - |\alpha|}$$

$$\Rightarrow \frac{\frac{1}{2} \cdot 2^{-4}}{1 - 0.95} \Rightarrow \frac{\frac{1}{2} \cdot \frac{1}{24}}{0.05}$$

$$\Rightarrow \frac{1/32}{0.05} = \frac{0.03125}{0.05}$$

$$\boxed{\text{Dead band} = 0.625}$$

Convert the following decimal numbers in floating point with five bits for mantissa and three bits for exponent.

(i) 4.5

first convert the decimal number into binar.

The binary equivalent of 4 is 100

The binary equivalent of .5 is .1

$0.5 \times 2 = 1$

$4.5_{10} = 100.1_2$

we have to represent by using 5 bit (4 + 1 Sign bit)

If the binary point is shifted three bits left we will get exponent.

Sign bit for exponent
↓ → shifted three bits left so +3
011

$.1001 \times 2^{011}$

0.1001×2^{011}

↳ Sign bit for mantissa (0 for positive, 1 for negative)

Floating point representation of $4.5_{10} = 0.1001 \times 2^{011}$

(ii) 1.5_{10}

First convert 1.5 to binary

1.1_2

To bring this into floating point format the binary point is shifted 1 bit left.

$.11 \times 2^{001}$

Since we have to represent by using five bit mantissa the above one is modified as

0.1100×2^{001}.

$1.5_{10} = 0.1100 \times 2^{001}$

(iii) 6.5_{10}

$6.5_{10} = 110.1_2$.

Shifting the binary point three bit left.

0.1101×2^{011}

$6.5_{10} = 0.1101 \times 2^{011}$

(iv) 7_{10}

convert 7_{10} to binary

111_2

we have to represent the binary in floating point format

The floating point format is

$2^C \times M$

C -> exponent (first bit represents sign)

M -> Mantissa (first bit represents sign)
positive number – 0
negative numbers – 0

111_2 is 111.0

Shifting the binary point three bits left.

0.1110×2^{011}

[Since the binary point is shifted three bits left in the exponent we are having 3].

$7_{10} = 0.110 \times 2^{011}$

(v) -7_{10}

convert 7_{10} to binary

111_2.

111.0

1.1110×2^{011}

↳ sign bit (1 since we have −7)

$$-7_{10} = 1.1110 \times 2^{011}$$

(vi) 0.25_{10}

convert 0.25_{10} to binary.

0.25×2

0.5×2

1.0

$$0.25_{10} = .01_2$$

$.01 \times 2^{000}$

$$0.25_{10} = 0.0100 \times 2^{000}$$

(vii) -0.25_{10}

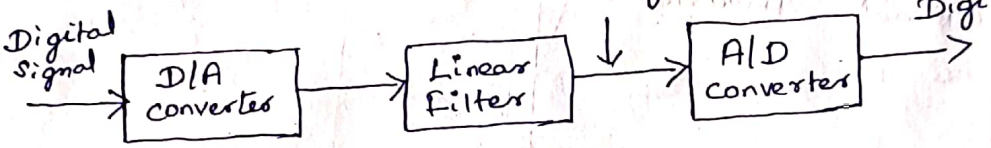
$$0.25_{10} = 1.0100 \times 2^{000}$$

Multirate Signal Processing :-

When the digital signal processing system handles processing at multiple sampling rates, then it is called multi-rate digital signal processing.

There are two methods available for sampling rate conversion.

1) First method - D/A conversion and resampling at required rate



* In this method, a digital signal is made to pass through a D/A converter then it is filtered if necessary, and then it is resampled at the desired sampling rate. The resampling of analog signal is carried out by using an A/D converter.

2) Second method - Sampling rate conversion in digital domain.

* In this method, sampling rate conversion is carried out entirely in the digital domain.
* This method doesnot need a D/A and A/D converters.
* This method uses interpolator, or decimator or both depending upon the sampling rate conversion factor.

First method

Advantages :-

(i) New sampling rate can have any value. It need not be related to old sampling rate.

(ii) The processing is straight forward and simplest.

Disadvantages :

(i) D/A and A/D conversion introduces additional distortion in the signal.

(ii) Two additional converters adds to hardware cost.

Second Method :-

Advantages :-

(i) Distortion due to sampling and quantization is reduced.

(ii) No hardware cost, since processing is totally in digital domai

Applications of Multirate DSP.

Multirate DSP find its application in

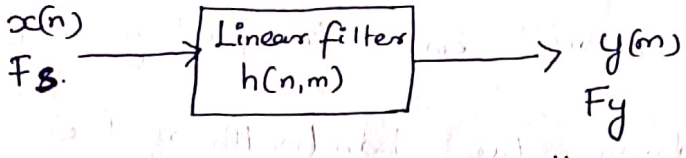
(i) Sub-band coding.

(ii) Voice privacy using analog phone lines.

(iii) Signal compression by Subsampling.

(iv) A/D, D/A Converters.

Sampling Rate Conversion as Linear filtering.



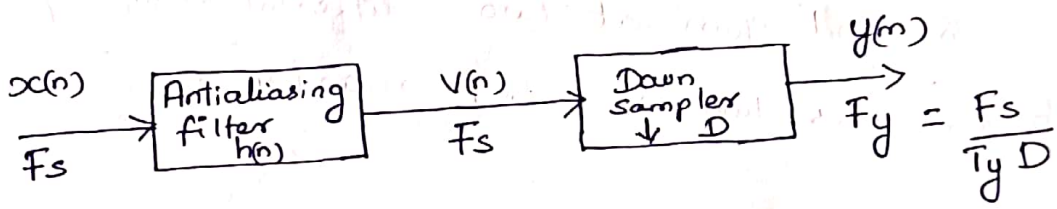
* The input signal has the sampling rate of F_s
* The output signal has the sampling rate of F_y.
* The linear filter has impulse response $h(n,m)$, which relates $x(n)$ and $y(m)$ at two different sampling rates. Hence $h(n,m)$ has two variables n and m.

Decimation by a Factor D

* Decimation by a factor D, means to reduce the sampling rate by a factor D. It is also called down sampling by D.
* Let us consider a discrete-time signal $x(n)$ with spectrum $X(\omega)$ is to be down sampled by an integer factor D.
* The spectrum $X(\omega)$ is assumed to be non-zero in the frequency interval $0 \leq |\omega| \leq \pi$
* Let the sampling frequency F_s and the maximum frequency F_{max} be related as $F_{max} \leq \frac{F_s}{2}$.

* If we reduce the sampling rate simply by selecting every D^{th} value of signal $x(n)$, the resulting signal would be an aliased version of $x(n)$ with folding frequency $\frac{f_s}{2D}$.

* To avoid aliasing, the bandwidth of the signal $x(n)$ is reduced to $f_{max} \leq \frac{f_s}{2D}$ or $\omega \leq \frac{\pi}{D}$



$F_s = \frac{1}{T_s}$.

* The input signal or sequence $x(n)$ is passed through a antialiasing filter ie. Low pass filter.

* The LPF is characterised by the impulse respone $h(n)$ and frequency response $H_D(\omega)$.

$$H_D(\omega) = \begin{cases} 1, & |\omega| \leq \frac{\pi}{D} \\ 0, & \text{elsewhere} \end{cases}$$

* The filter eliminates the spectrum of $X(\omega)$ in the ra

$\frac{\pi}{D} < \omega < \pi$.

Derivation of Decimation Equation:-

The output of the filter is a sequence V(n) which is given by

$$V(n) = \sum_{k=0}^{\infty} h(k)\, x(n-k) \qquad \text{— (1)}$$

When V(n) is down sampled by the factor D, we get

$$y(m) = V(mD)$$

$$y(m) = \sum_{k=0}^{\infty} h(k)\, x(mD-k) \qquad \text{— (2)}$$

* The down sampling operation is a time varying operation

Relationship between the spectrum of x(n) and y(m)

* The frequency-domain characteristics of the output sequence y(m) may be obtained by relating the spectrum of y(m) to the spectrum of the input sequence x(n).

The sequence $\tilde{V}(n)$ can be defind as follows.

$$\tilde{V}(n) = \begin{cases} V(n), & n = 0, \pm D, \pm 2D, \pm 3D, \cdots \\ 0, & \text{elsewhere} \end{cases} \qquad \text{— (3)}$$

* The sequence $\tilde{V}(n)$ may be viewed as a sequence obtained by multiplying V(n) with a periodic train of impulses p(n), with period D.

$$\tilde{V}(n) = V(n) \cdot p(n) \quad \text{— (4)}$$

The discrete Fourier series representation of $p(n)$ is given by.

$$p(n) = \frac{1}{D} \sum_{k=0}^{D-1} e^{\frac{j 2\pi k n}{D}} \quad \text{— (5)}$$

$$y(n) = V(n) \cdot p(n)$$

$$y(n) = \tilde{V}(nD). \quad \text{— (6)}$$

$$y(n) = V(nD)\, p(nD)$$

Taking z- transform of the output sequence $y(n)$

$$y(n) = \tilde{V}(nD)$$

$$Y(z) = \sum_{m=-\infty}^{\infty} \tilde{V}(nD)\, z^{-n} \quad \text{— (7)}$$

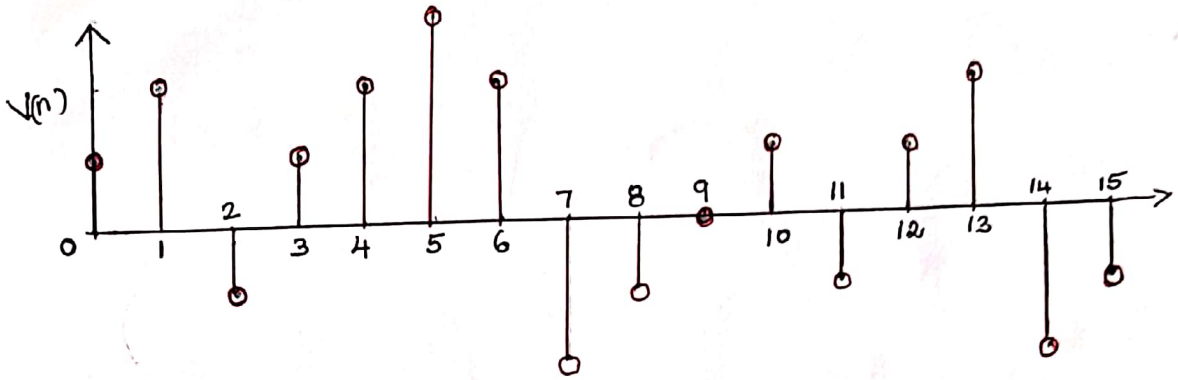
we know that $\tilde{V}(n) = 0$, except multiples of D.

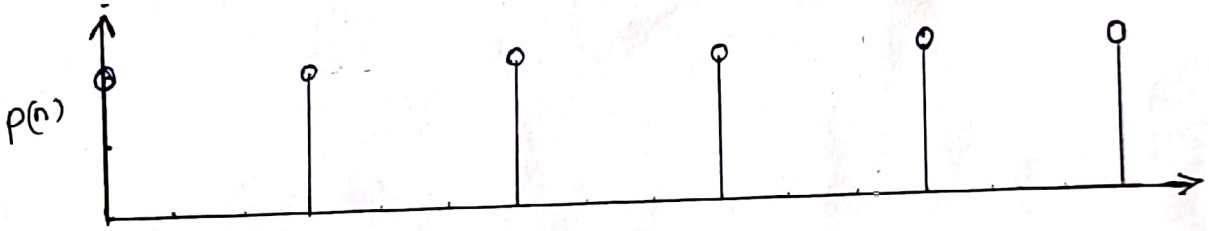
Hence the variable 'm' in above equation can be manipulated as follows.

$$Y(z) = \sum_{m=-\infty}^{\infty} \tilde{V}(m)\, z^{\frac{-m}{D}}$$

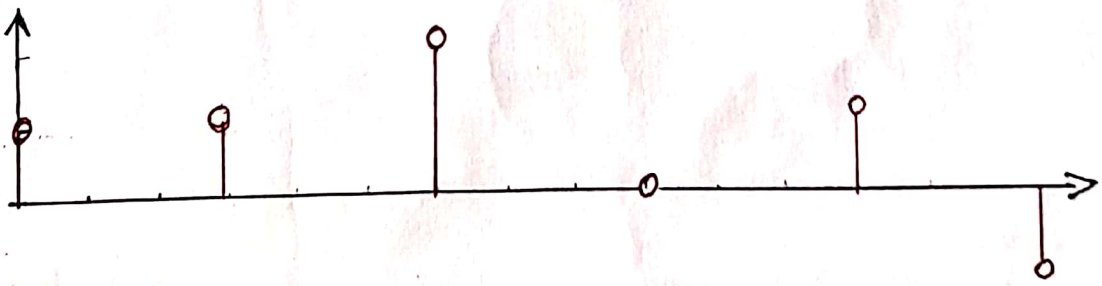
$$Y(z) = \sum_{m=-\infty}^{\infty} V(m)\, p(m)\, z^{\frac{-m}{D}} \quad \text{— (8)}$$

Down Sampling operation. Observed as multiplication of two sequences.





y(m) = v(n).p(n)



$$Y(z) = \sum_{m=-\infty}^{\infty} V(m) \cdot \left[\frac{1}{D} \sum_{K=0}^{D-1} e^{j\frac{2\pi mK}{D}} \right] z^{-\frac{m}{D}}$$

$$Y(z) = \sum_{m=-\infty}^{\infty} V(m) \left[\frac{1}{D} \sum_{K=0}^{D-1} e^{-j\frac{2\pi K}{D}} \cdot z^{+\frac{1}{D}} \right]^{-m}$$

$$Y(z) = \frac{1}{D} \sum_{K=0}^{D-1} \sum_{m=-\infty}^{\infty} V(m) \cdot \left[e^{-j\frac{2\pi K}{D}} \cdot z^{+\frac{1}{D}} \right]^{-m}$$

$$Y(z) = \frac{1}{D} \sum_{K=0}^{D-1} V\left[e^{j\frac{2\pi K}{D}} \cdot z^{\frac{1}{D}} \right] \quad \text{—— (9)}$$

where $$V\left[e^{-j\frac{2\pi K}{D}} z^{\frac{1}{D}} \right] = \sum_{m=-\infty}^{\infty} V(m) \left[e^{-j\frac{2\pi K}{D}} \cdot z^{\frac{1}{D}} \right]^{-m}$$

But $V(n) = h(n) * x(n)$.

Hence thier z transform will multiply.

i.e $V(z) = H(z) * X(z)$

$$Y(z) = \frac{1}{D} \sum_{K=0}^{D-1} H\left(e^{-j\frac{2\pi K}{D}} \cdot z^{\frac{1}{D}} \right) X\left(e^{-j\frac{2\pi K}{D}} \cdot z^{\frac{1}{D}} \right) \quad \text{—— (10)}$$

* If $Y(z)$ is evaluated on unit circle, then we get spectrum of $y(n)$. For this we have to put $z = e^{j\omega}$.

Here w_y is frequency of $y(n)$.

$$Y(e^{jw_y}) = \frac{1}{D} \sum_{K=0}^{D-1} H\left(e^{-j\frac{2\pi K}{D}} \cdot e^{j\frac{w_y}{D}}\right) X\left(e^{-j\frac{2\pi K}{D}} \cdot e^{j\frac{w_y}{D}}\right)$$

$$Y(e^{jw_y}) = \frac{1}{D} \sum_{K=0}^{D-1} H\left(e^{j\frac{w_y - 2\pi K}{D}}\right) X\left(e^{\frac{w_y - 2\pi K}{D}}\right)$$

We can write $Y(e^{jw_y})$ as $Y(w_y)$

$$Y(w_y) = \frac{1}{D} \sum_{K=0}^{D-1} H\left(\frac{w_y - 2\pi K}{D}\right) X\left(\frac{w_y - 2\pi K}{D}\right)$$

From the above equation we observe that $X\left(\frac{w_y - 2\pi K}{D}\right)$ indicates replicas of $X\left(\frac{w_y}{D}\right)$ at $0, 2\pi, 4\pi, 6\pi, \ldots 2\pi(D-1)$.

These replicas are generated due to down sampling operation.

If we consider only the first spectrum substitute $K=0$ in the above equation. we will get.

$$Y(w_y) = \frac{1}{D} H\left(\frac{w_y}{D}\right) X\left(\frac{w_y}{D}\right) \quad \text{---} \quad (11)$$

The antialiasing filter bandlimits the signal to $\frac{\pi}{D}$.

It has the magnitude response of

$$H(\omega_s) = \begin{cases} 1 & \text{for } \omega_s \leq \frac{\pi}{D} \\ 0 & \text{elsewhere.} \end{cases} \quad \text{---(12)}$$

$$\omega_s = 2\pi \frac{f}{f_s}$$

$$\omega_y = 2\pi \frac{f}{f_y}.$$

The two sampling rates are related as $F_y = \frac{F_s}{D}$

$$\omega_y = 2\pi \cdot \frac{F}{\frac{f_s}{D}}$$

$$\omega_y = 2\pi \frac{f}{f_s} \cdot D.$$

$$\boxed{\omega_y = \omega_s D}$$

or

$$\boxed{\omega_s = \frac{\omega_y}{D}}$$

$$H\left(\frac{\omega_y}{D}\right) = \begin{cases} 1 & \text{for } \frac{\omega_y}{D} \leq \frac{\pi}{D} \\ 0 & \text{elsewhere.} \end{cases}$$

putting $H\left(\frac{w_y}{D}\right) = 1$ in equation.

$$Y(w_y) = \frac{1}{D} X\left(\frac{w_y}{D}\right) \quad \text{---} \quad (13)$$

The above equation relates the spectrums of x(n) and y(n)

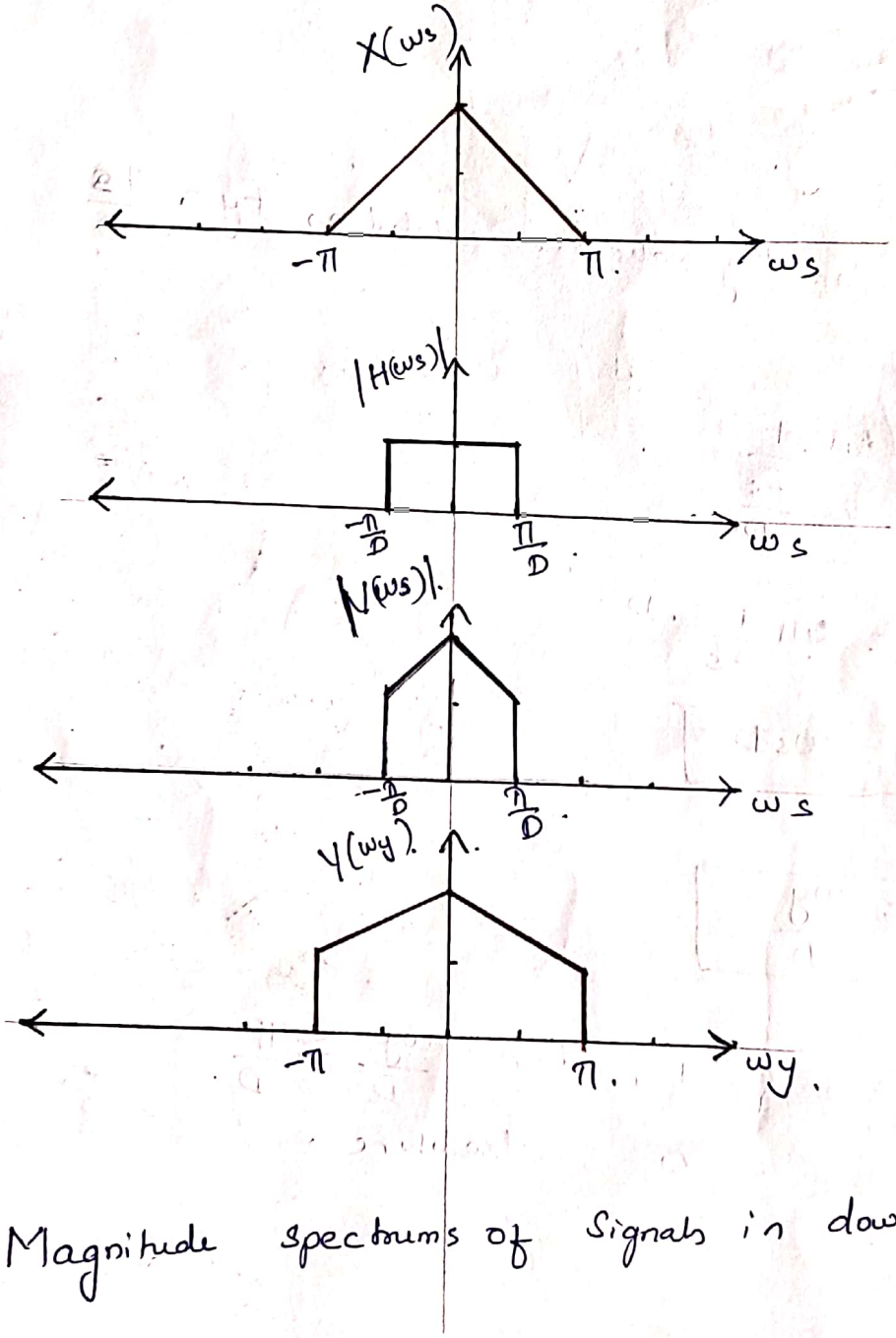


Fig. Magnitude spectrums of signals in down sampling operation

Interpolation by factor I.

* Interpolation by a factor I, means to increase the sampling rate by a factor I.
* It is also called upsampling by I.
* If the sampling frequency of input signal is f_s, then it is increased by I. Thus the sampling frequency of the output signal is $F_y = I f_s$.
* The interpolator simply puts $(I-1)$ zeros between sucessive samples of $x(n)$.
* The anti-imaging filter removes the image spectrums and interpolates the samples of $V(n)$

* The output $y(n)$ of the anti imaging filter is given as the convolution of $h(n)$ and $V(n)$ i'e.

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

But the Value of $V(k)$ is zero except at multiples of I.

Hence the product

$$h(m-k)\,V(k) = \begin{cases} 0 & \text{If } k \text{ is not integer multiple of } I \\ \text{Non zero} & \text{If } k \text{ is integer multiple of } I. \end{cases}$$

Hence we can replace 'k' by kI in the above equation of convolution ie

$$y(m) = \sum_{k=-\infty}^{\infty} h(m-kI)\,V(kI)$$

But $V(kI) = x(k)$, hence above equation becomes,

$$y(m) = \sum_{k=-\infty}^{\infty} h(m-kI)\,x(k) \qquad \text{---- (1)}$$

This is an equation for output signal.

Relationship between the spectrums of $x(n)$ and $y(n)$

$V(n)$ can be expressed in terms of $x(n)$ as.

$$V(m) = \begin{cases} x\left(\frac{m}{I}\right), & m = 0, \pm I, \pm 2I, \ldots \\ 0 & \text{Otherwise.} \end{cases}$$

This means $V(n)$ is non zero only at integer multiples of I.

Taking z transform of $V(n)$ we get.

$$V(z) = \sum_{m=-\infty}^{\infty} V(m)\,z^{-m}.$$

Note that $V(n) = 0$, except when 'n' is integer multiple of I

Hence above equation can be written as

$$V(z) = \sum_{m=-\infty}^{\infty} V(mI)\, z^{-mI}$$

But $V(mI) = x(m)$ Hence the above equation becomes.

$$V(z) = \sum_{m=-\infty}^{\infty} x(m)\, z^{-mI}$$

$$= \sum_{m=-\infty}^{\infty} x(m) \left(z^{I}\right)^{-m}$$

$$\boxed{V(z) = X(z^{I})} \qquad \text{---- (2)}$$

If $V(z)$ is evaluated on unit circle, then its spectrum can be obtained.

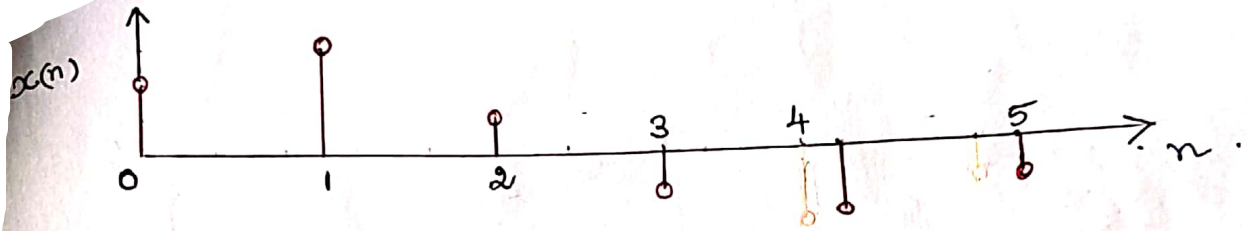
putting $z = e^{jwy}$ in above equation.

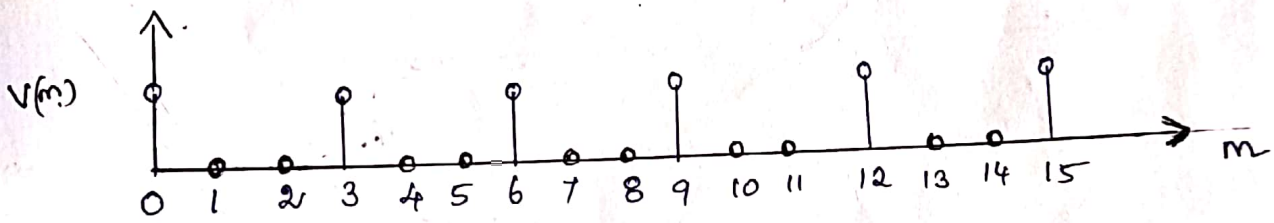
$$V(e^{jwy}) = X(e^{jwyI})$$

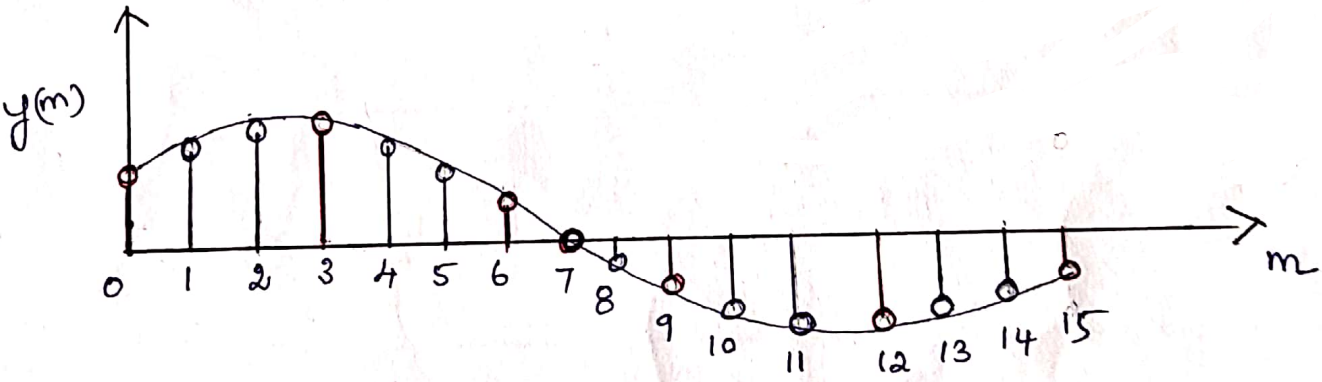
$V(e^{jwy})$ can be written as $V(wy)$ and $X(e^{jwy})$ as $X(wy)$

$$V(wy) = X(wyI) \qquad \text{---- (3)}$$

Waveform of Up sampling.







But $f_y = I f_s$.

$$\omega_y = \frac{2\pi f}{f_y} = \frac{2\pi f}{I f_s} = \frac{\omega_x}{I}.$$

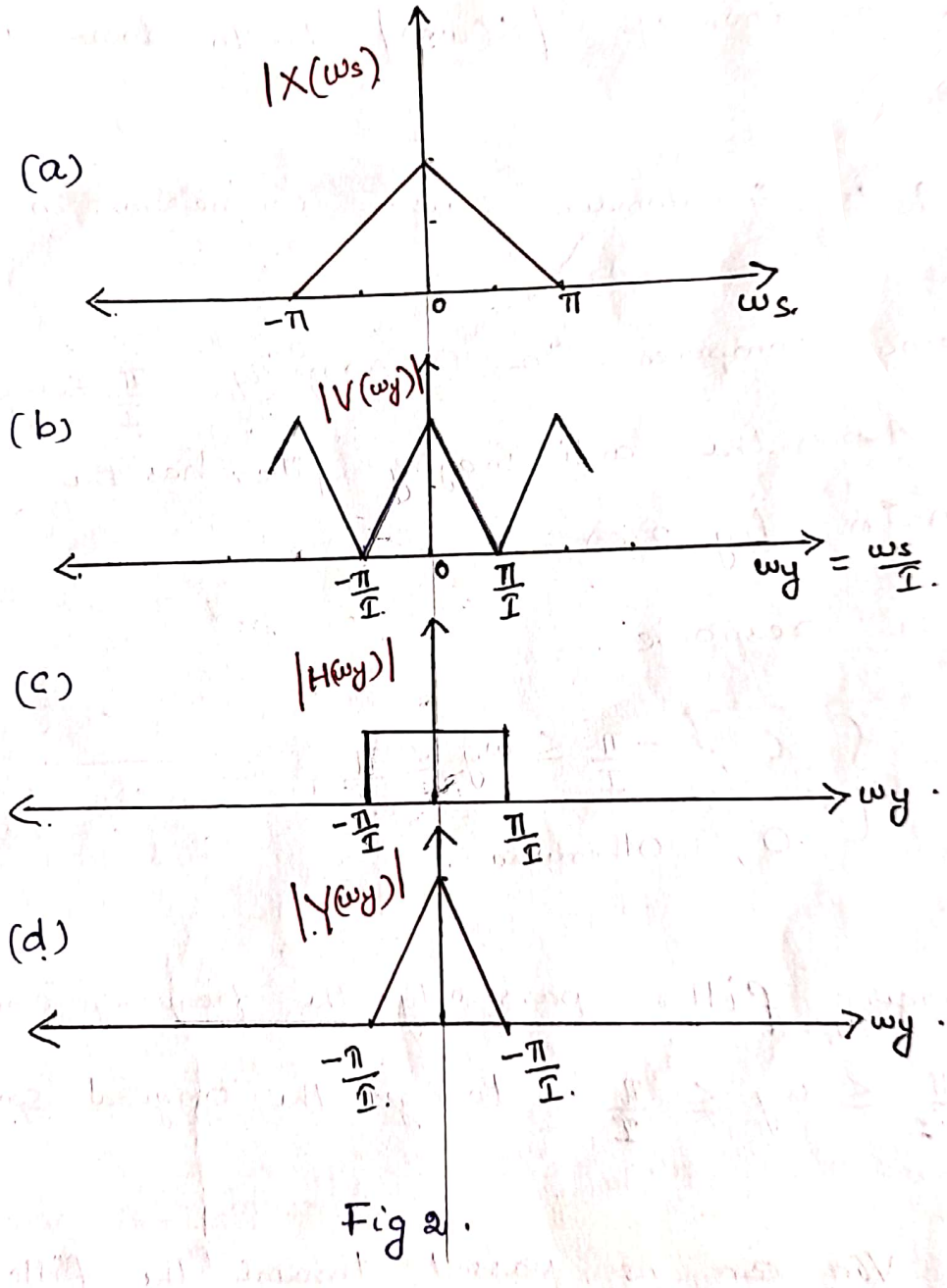


Fig 2.

Spectrum of signals in upsampling operation.

The Fig 2(a) shows an arbitrary spectrum of $x(n)$.

The spectrum of $x(n)$ has the frequencies from $-\pi$ to π.

ie $-\pi \leq \omega_s \leq \pi$.

The Fig 2(b) shows the spectrum of $V(n)$. It is the I- fold periodic repetition of $|X(\omega_s)|$, due to upsampling operation.

Upsampling in time domain causes compression in frequency domain.

The frequency components in the range of $-\frac{\pi}{I} \leq \omega_y \leq \frac{\pi}{I}$ are unique. Hence the anti-Imaging filter has the response as shown in fig 2(c).

It has the response.

$$H(\omega_y) = \begin{cases} C, & -\frac{\pi}{I} \leq \omega_y \leq \frac{\pi}{I} \\ 0, & \text{otherwise.} \end{cases} \quad \text{——} \quad (4)$$

The antiImaging filter pass only the frequency components. from $-\frac{\pi}{I} \leq \omega_y \leq \frac{\pi}{I}$ to get the original spectrum back.

The Signal $V(n)$ is passed through the filter. Hence from equation (3) and equation (4) we can write the spectrum of output signal as.

$$Y(\omega_y) = \begin{cases} C\,X(\omega_y I), & -\frac{\pi}{I} \leq \omega_y \leq \frac{\pi}{I} \\ 0, & \text{otherwise.} \end{cases} \quad -(5)$$

This equation gives the relationship between the spectrums of output and input signals.

Value of Scale factor C.

The scale factor is selected such that

$$y(m) = x\left(\frac{m}{I}\right) \quad \text{for } m = 0, \pm I, \pm 2I, \ldots$$

By inverse fourier transform $y(m)$ can be obtained as.

$$y(m) = \frac{1}{2\pi} \int_{-\pi}^{\pi} Y(w_y)\, e^{jwm}\, dw_y.$$

For $m = 0$

$$y(0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} Y(w_y)\, e^{0} \cdot dw_y.$$

But $Y(w_y) = C\, x(w_y I)$

$$y(0) = \frac{1}{2\pi} \int_{-\frac{\pi}{I}}^{\frac{\pi}{I}} C\, x(w_y I)\, dw_y.$$

But $w_y = \frac{w_s}{I}$ or $w_s = w_y I$

Hence the integration limits will be $-\pi$ to π ie.

$$y(0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} C\, X\left(\frac{w_s}{I}\right) \frac{dw_s}{I}$$

$$y(0) = \frac{C}{I}\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} x(\omega_s)\, d\omega_s\right]$$

$$y(0) = \frac{C}{I}\, x(0)$$

Thus $y(0) = x(0)$, if $C = I$.

This is the value of scaling factor.

Sampling Rate Conversion by a Rational factor I/D.

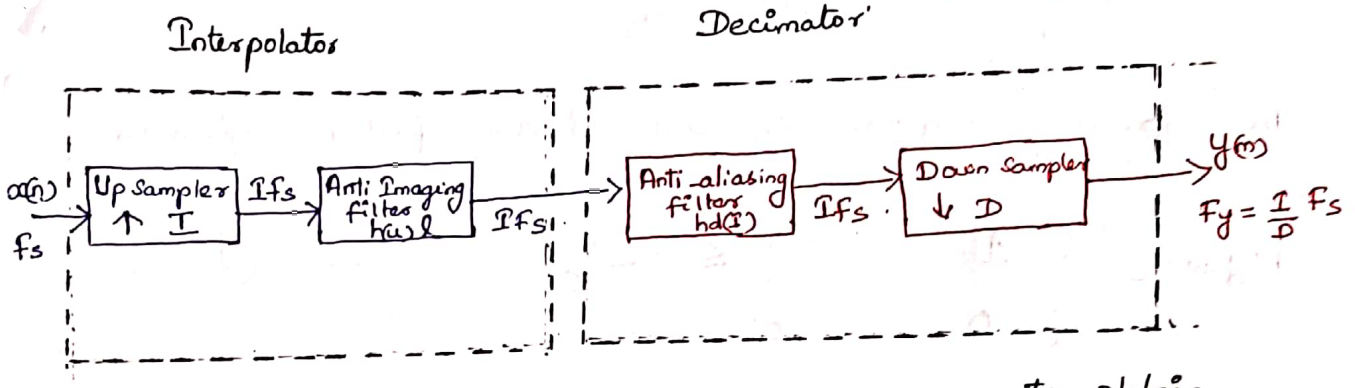


Fig:- Cascade of Interpolator and Decimator to obtain Sampling rate conversion by I/D.

* The interpolation by a factor I is obtained first to increase the sampling rate to IF_s.
* The output of interpolator is then decimated by a factor D, so that the final output rate is $f_y = \frac{I}{D} f_s$.
* Such operation can be implemented by cascade connection of interpolator and decimator.
* In the above figure there are two lowpass filters.
* The overall cutoff frequency will be minimum of the two cutoff frequencies.

The frequency response of anti Imaging filter is given as.

$$H_u(\omega) = \begin{cases} C & -\frac{\pi}{I} \le \omega \le \frac{\pi}{I} \\ 0 & \text{otherwise.} \end{cases} \quad \text{---} \quad (1)$$

$$H_u(\omega) = \begin{cases} I & -\frac{\pi}{I} \le \omega \le \frac{\pi}{I} \\ 0 & \text{otherwise} \end{cases}$$, since $C = I$ (Scaling)

The frequency response of anti-aliasing filter is given as,

$$H_d(\omega) = \begin{cases} 1 & -\frac{\pi}{D} \le \omega \le \frac{\pi}{D} \\ 0 & \text{otherwise} \end{cases} \quad \text{—— (2)}$$

* The overall cascading effect of lowpass filters will have a cutoff frequency which is minimum of $\frac{\pi}{I}$ and $\frac{\pi}{D}$.

Hence we can write frequency response of combined filter as.

$$H(\omega) = \begin{cases} I & |\omega| \le \min\left(\frac{\pi}{D}, \frac{\pi}{I}\right) \\ 0 & \text{Otherwise} \end{cases} \quad \text{—— (3)}$$

* Thus single filter can be used, as defined by above equation. The block diagram can be modified as follows.

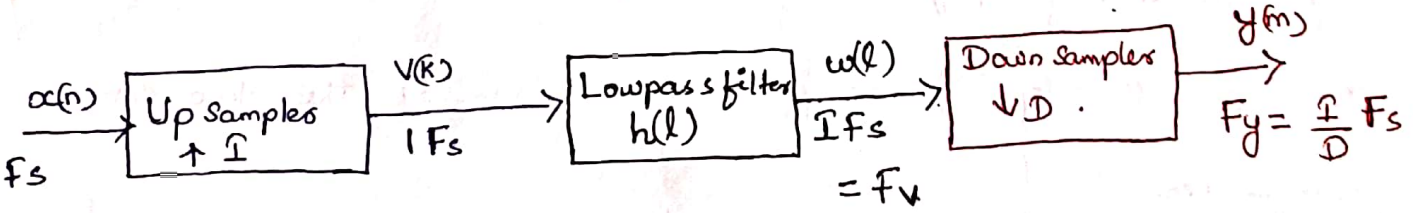


Fig:- Sampling rate conversion by a factor $\frac{I}{D}$.

Derivation for output y(m)

The output of lowpass filter is given as.

$$w(l) = \sum_{k=-\infty}^{\infty} h(l-k)\, v(k).$$

$$= \sum_{k=-\infty}^{\infty} h(l-kI)\, x(k) \quad \text{— (4)}$$

* The output of down sampler is given as.

$$y(m) = w(mD)$$

$$y(m) = \sum_{k=-\infty}^{\infty} h(mD-kI)\, x(k) \quad \text{— (5)}$$

This is the equation for output sequence.

Relationship between the spectrums of x(n) and y(m)

* The output signal of upsampler is V(k) with frequency ω_v.

For this signal, using $V(\omega_y) = X(\omega_y I)$

we can write.

$$V(\omega_v) = X(\omega_v I) \quad \text{— (6)}$$

* The signal V(k) is passed through a low pass filter and the output signal is w(l)

Hence we can write.

$$W(\omega_v) = H(\omega_v) \cdot V(\omega_v)$$

putting the values from equation 3 and equation 6 in above equation we get.

$$W(\omega_v) = \begin{cases} I \times (\omega_v I), & |\omega| \le \min\left(\frac{\pi}{D}, \frac{\pi}{I}\right) \\ 0, & \text{Otherwise.} \end{cases} \quad \text{— (7)}$$

We know that the spectrum of downsampled signal is given as.

$$Y(\omega_y) = \frac{1}{D} \times \left(\frac{\omega_y}{D}\right)$$

But the input to the down sampler is $w(l)$.

Hence above equation can be written as.

$$Y(\omega_y) = \frac{1}{D} W\left(\frac{\omega_y}{D}\right) \quad \text{— (8)}$$

But $\omega_v = \frac{2\pi f}{f_v} = \frac{2\pi f}{f_y} = 2\pi \frac{f}{F_y} \cdot \frac{1}{D} = \frac{\omega_y}{D}$.

So $\omega_v = \frac{\omega_y}{D}$

Hence the right hand side of above equation (7) can be written as.

$$Y(\omega_y) = \frac{1}{D} W(\omega_v)$$

putting ω_v from equation (6)

$$Y(\omega_y) = \begin{cases} \frac{I}{D} \times (\omega_v I) & |\omega_v| \le \min\left(\frac{\pi}{D}, \frac{\pi}{I}\right) \\ 0 & \text{otherwise.} \end{cases}$$

Since $\omega_v = \frac{\omega_y}{D}$ above equation can be modified as

$$Y(\omega_y) = \begin{cases} \frac{I}{D} \times \left(\frac{I}{D}\omega_y\right) & , \left|\frac{\omega_y}{D}\right| \leq \min\left(\frac{\pi}{D}, \frac{\pi}{I}\right) \\ 0 & \text{otherwise.} \end{cases}$$

In the above equation $\left|\frac{\omega_y}{D}\right| \leq \min\left(\frac{\pi}{D}, \frac{\pi}{I}\right)$ can be written as

$$|\omega_y| \leq \min\left(\pi, \frac{D}{I}\pi\right)$$

Then we have.

$$Y(\omega_y) = \begin{cases} \frac{I}{D} \times \left(\frac{I}{D}\omega_y\right) & |\omega_y| \leq \min\left(\pi, \frac{D}{I}\pi\right) \\ 0 \end{cases} \quad -(4$$

This is the required equation that relates the spectrums of input and output.

8.11 Polyphase Structure of Decimator

In section 6.9.4 we studied about the polyphase realization of FIR filters, where the transfer function $H(z)$ is decomposed into M branches given by

$$H(z) = \sum_{m=0}^{M-1} z^{-m} P_m(z^M) \tag{8.22}$$

where

$$P_m(z) = \sum_{n=0}^{[(N+1)/M]} h(Mn+m) z^{-n} \tag{8.23}$$

The z-transform of an infinite sequence is given by

$$H(z) = \sum_{n=-\infty}^{\infty} h(n) z^{-n} \tag{8.24}$$

In this case $H(z)$ can be decomposed into M-branches as

$$H(z) = \sum_{m=0}^{M-1} z^{-m} P_m(z^M)$$

where $P_m(z) = \sum_{r=-\infty}^{\infty} h(rM+m) z^{-r}$

$$H(z) = \sum_{m=0}^{M-1} \sum_{r=-\infty}^{\infty} z^{-m} h(rM+m) z^{-rM}$$

$$= \sum_{m=0}^{M-1} \sum_{r=-\infty}^{\infty} h(rM+m) z^{-(rM+m)} \tag{8.25}$$

$$x_m(r) = x(rM-m)$$
$$x_m(-r) = x(-rM-m)$$
$$x_m(n-r) = x[n-(rM+m)]$$

Let $h(rM+m) = p_m(r)$

$$\Longrightarrow H(z) = \sum_{m=0}^{M-1} \sum_{r=-\infty}^{\infty} p_m(r) z^{-(rM+m)}$$

$$Y(z) = \sum_{m=0}^{M-1} \sum_{r=-\infty}^{\infty} p_m(r) X(z) z^{-(rM+m)}$$

$$y(n) = \sum_{m=0}^{M-1} \sum_{r=-\infty}^{\infty} p_m(r) x[n-(rM+m)] \tag{8.26}$$

Let $x_m(r) = x(rM-m)$ then

$$y(n) = \sum_{m=0}^{M-1} \sum_{r=-\infty}^{\infty} p_m(r) x_m(n-r) \tag{8.27}$$

$$= \sum_{m=0}^{M-1} p_m(n) \star x_m(n)$$

$$= \sum_{m=0}^{M-1} y_m(n) \tag{8.28}$$

where $y_m(n) = p_m(n) \star x_m(n)$

The operation $p_m(n) \star x_m(n)$ is known as polyphase convolution, and the overall process is polyphase filtering.

If $M = 3$ then

$$y(n) = \sum_{m=0}^{2} y_m(n)$$

$$= y_0(n) + y_1(n) + y_2(n)$$

$$= p_0(n) \star x_0(n) + p_1(n) \star x_1(n) + p_2(n) \star x_2(n) \tag{8.29}$$

We know that $x_m(n)$ can be obtained first by delaying $x(n)$ by m units, and then downsampling by a factor M. Next $y_m(n)$ can be obtained by convolving $x_m(n)$ with $p_m(n)$. The structure of a polyphase decimator with 3 branches and a sampling rate reduction by a factor three is shown in Fig. 8.51.

For a general case of M branches and a sampling rate reduction by a factor M, the structure of polyphase decimator is shown in Fig. 8.52. The splitting of $x(n)$ into the low rate sub-sequence $x_0(n), x_1(n) \ldots x_{M-1}(n)$ is often represented by a commutator.

In the configuration shown in Fig. 8.52 the input values $x(n)$ enter the delay chain at high rate. Then the M downsampler sends the group of M input values

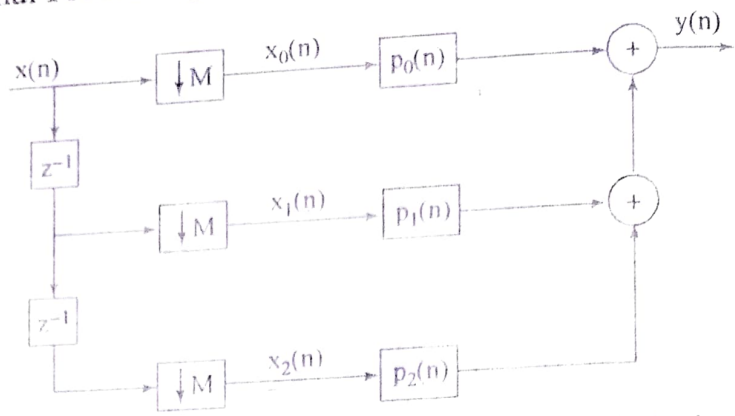


Fig. 8.51 Polyphase structure of a 3 branch decimator

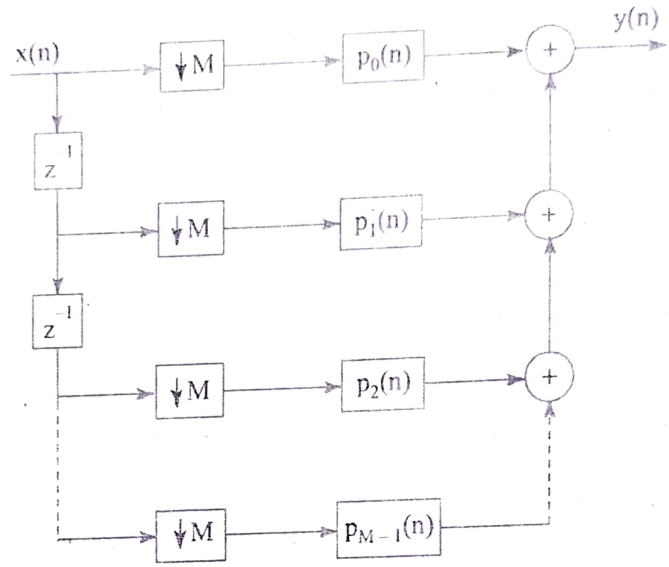


Fig. 8.52 Polyphase structure of a M-branch decimator

to M filters at time $n = mM$. For example at time $n = 0$, the value of $x_0(0) = x(0), x_1(0) = x(-1), x_2(0) = x(-2), \ldots x_{M-1}(0) = x(-M+1)$ are sent.

In Fig. 8.53 to produce the output $y(0)$, the commutator must rotate in counter-clockwise direction starting from $m = M-1, \ldots m = 2, m = 1, m = 0$ and give the input values $x(-M+1), \ldots x(-2), x(-1), x(0)$ to the filters $p_{M-1}(n), \ldots p_2(n), p_1(n), p_0(n)$.

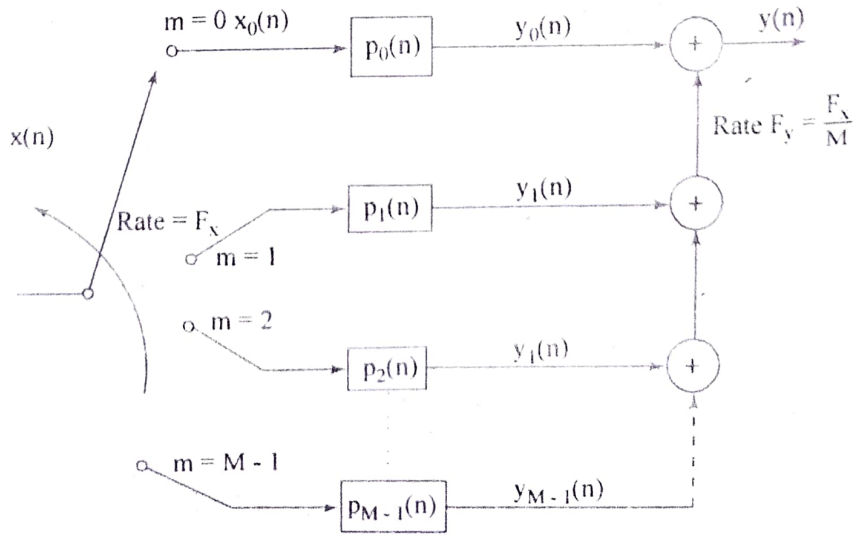


Fig. 8.53 Polyphase decimator with a commutator

8.12 Polyphase Decimation Using the z-transform

The z-transform representation of a decimator is shown in Fig. 8.54.

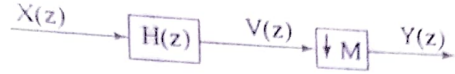


Fig. 8.54

The M-branch polyphase decomposition of $H(z)$ is given by

$$H(z) = \sum_{m=0}^{M-1} z^{-m} P_m(z^M) \tag{8.30}$$

The subfilters $P_0(z), P_1(z) \ldots, P_{M-1}(z)$ are FIR filters and when combined in the right phase sequence produce the original filter $H(z)$.

Now replacing the transfer function $H(z)$ in Fig. 8.54 by the polyphase structure results as in Fig. 8.55.

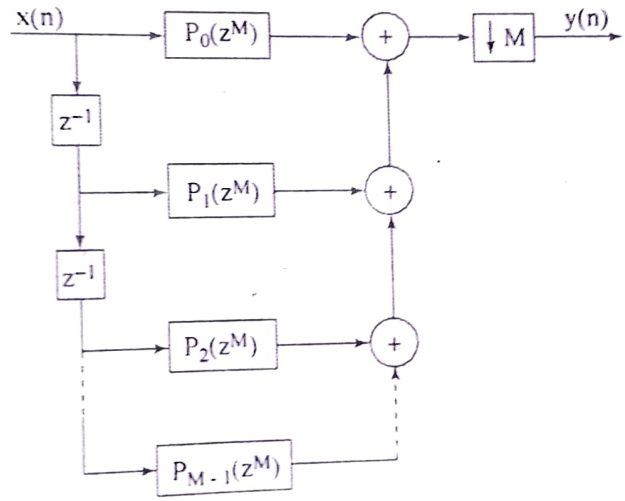


Fig. 8.55

Using first identity Fig. 8.55 can be changed as in Fig. 8.56.

The third identity in Fig. 8.25 can be used to derive the version in Fig. 8.57 in which both the number of filter operations and the amount of memory required are reduced by a factor of M.

8.13 Polyphase Structure of Interpolator

By transposing the decimator structure shown in Fig. 8.57, we can obtain polyphase structure for interpolator, which consists of a set of L sub-filters connected in parallel as shown in Fig. 8.58.

Here the polyphase components of impulse response are given by

$$p_m(n) = h(nL + m) \quad m = 0, 1, 2, \ldots, L-1 \tag{8.31}$$

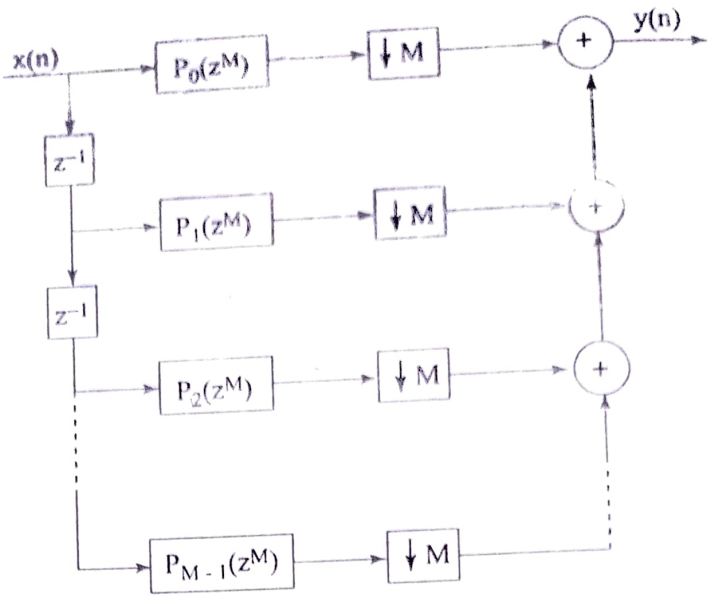


Fig. 8.56

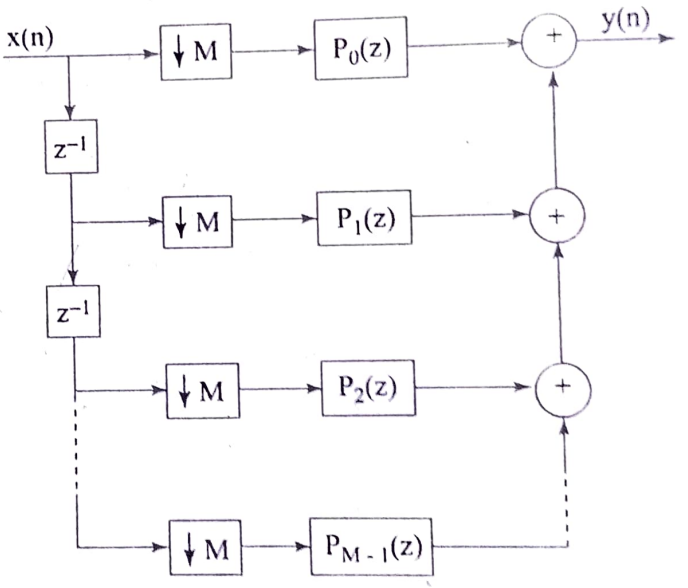


Fig. 8.57

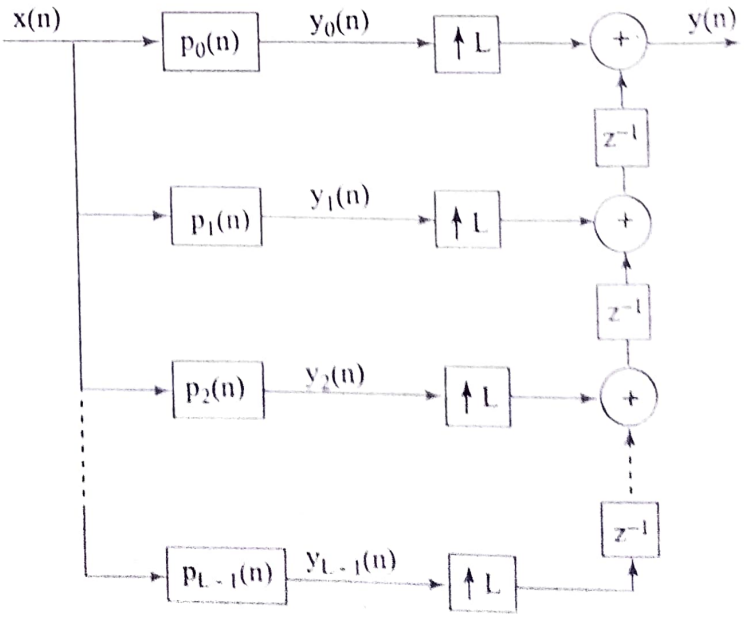


Fig. 8.58

where $h(n)$ is the impulse response of anti-imaging filter. The output of L sub-filters can be represented as

$$y_m(n) = x(n)p_m(n) \quad m = 0, 1, 2, \ldots, L-1 \tag{8.32}$$

By upsampling with a factor L and adding a delay z^{-m} the polyphase components are produced from $y_m(n)$. These polyphase components are all added together to produce the output signal $y(n)$.

The output $y(n)$ also can be obtained by combining the signals $x_m(n)$ using a commutator as shown in Fig. 8.59.

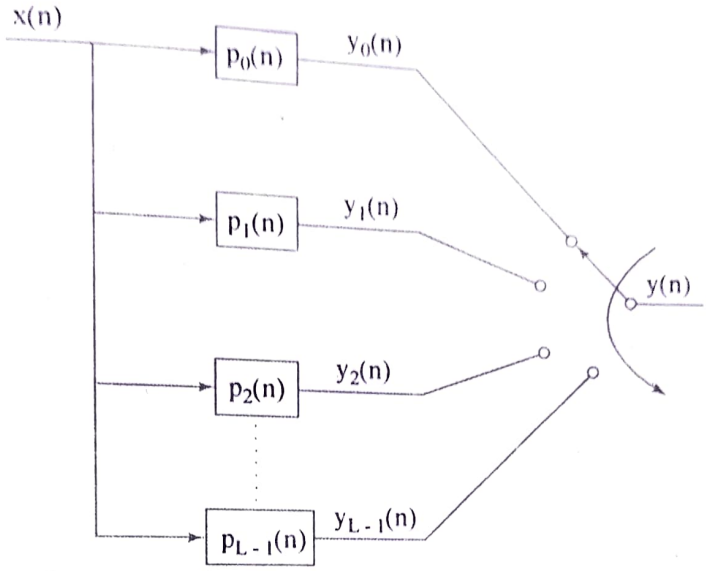


Fig. 8.59

8.14 Polyphase Interpolation Using the z-transform

The block diagram of an interpolator is shown in Fig. 8.60.

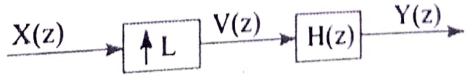


Fig. 8.60

The transfer function $H(z)$ of the interpolator is given by

$$H(z) = \sum_{m=0}^{L-1} z^{-m} p_m(z^L) \tag{8.33}$$

$H(z)$ in Fig. 8.60 can be realized using polyphase structure as shown in Fig. 8.61.

Using Fourth and Sixth identity an efficient structure for interpolator can be obtained as shown in Fig. 8.62.

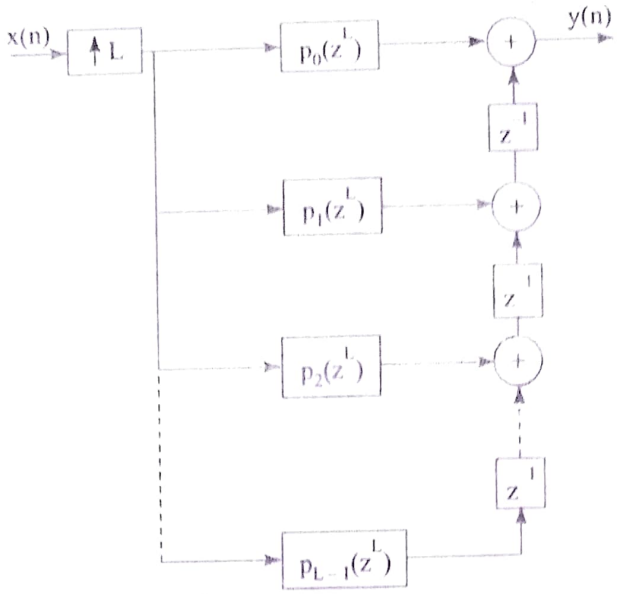


Fig. 8.61

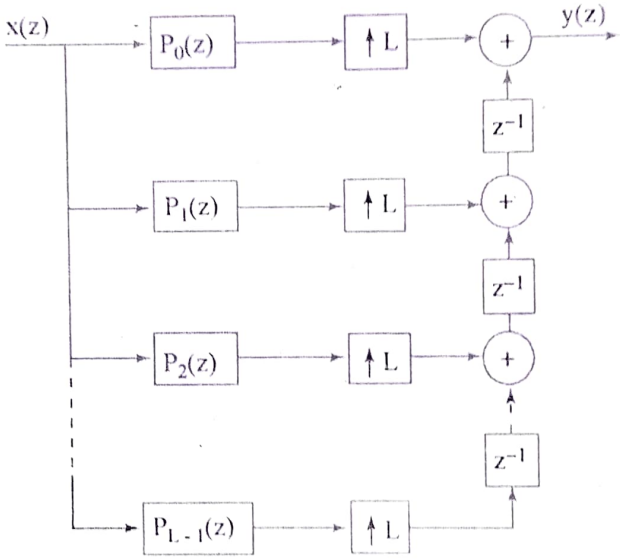


Fig. 8.62

8.15 Multistage Implementation of Sampling Rate Conversion

If the decimation factor M and/or interpolation factor L are much larger than unity, the implementation of sampling rate conversion in a single stage is computational inefficient. Therefore for performing sampling rate conversion for either $M >> 1$ and /or $L >> 1$ we go in for multistage implementation.

If the interpolation factor $L >> 1$, then we express L into a product of positive integers as

$$L = \prod_{i=1}^{N} L_i \tag{8.34}$$

Then each interpolator L_i is implemented and cascaded to get N stages of interpolation and filtering as shown in Fig. 8.63.