UNIT – IV

Multirate DSP: Introduction, Sampling Rate Conversion, Decimation of Sampling rate by an Integer factor, Interpolation of sampling rate by an Integer Factor, Sampling rate alteration or conversion by a rational factor. Filter design and implementation for sampling rate alteration or conversion: Direct form FIR digital filter structures, Polyphase filter structure, Time varying digital filter structures. Sampling rate conversion by an arbitrary factor: First order approximation & Second order approximation method. Applications of Multirate Digital Signal Processing (MDSP).

There are various areas in which MDSP is used. Some of few are given as under:

1. Radar Systems
2. Antenna Systems
3. Speech and Audio Processing Systems
4. Communication Systems

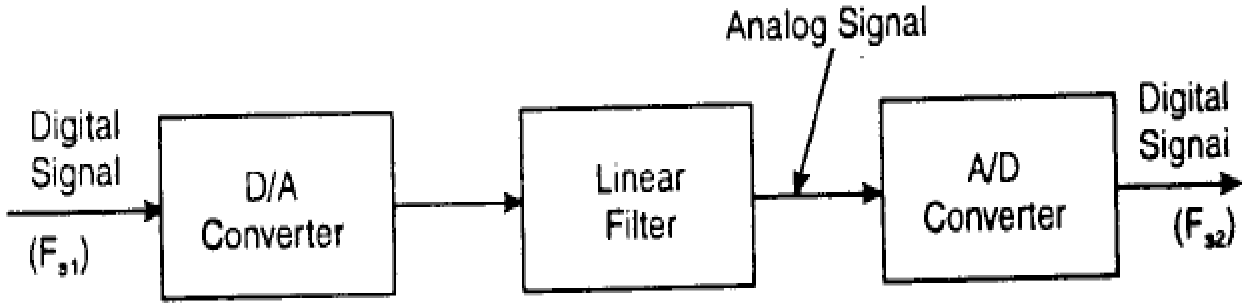
Advantages of using MDSP

1. Computational requirements are less
2. Storage for filter coefficients are less
3. Finite arithmetic effects are less
4. Filter order required in Multirate application are low
5. Sensitivity to filter coefficient lengths are less.

Sampling Rate Conversion Methods

There are two sampling rate conversion methods that are used in MDSP :

First Method : D/A Converter, Linear Filter and A/D Converter.



Sampling Rate Conversion using D/A Converter and A/D Converter.

Second Method :

In this method sampling rate conversion is perfoemed entirely in the digital-domain. This method does not require any ADC or DAC. This method uses interpolator or decimator or both depending upon the sampling rate conversion factor.

Advantages

of first method is that the new sampling rate can be arbitrarily selected and this new sampling rate has no special relationship with the old sampling rate.

Disadvantages

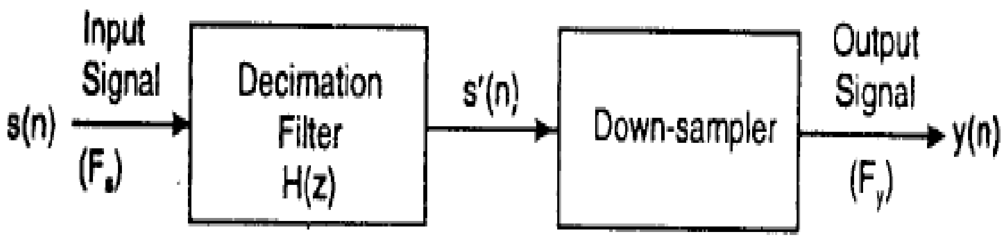
of first method is that there is a signal distortion introduced by the D/A convertor in the signal reconstruction and by the quantization noise in the ADC.

SAMPLING RATE CONVERSION

There are two cases of sampling rate conversion 01. DECIMATION AND 02. INTERPOLATION

Decimation. The process of reducing the sampling rate by a factor (D) is called Decimation of Sampling rate.

IT IS ALSO CALLED DOWN SAMPLING BY FACTOR INTEGER BY D



Block Diagram of a Decimator.

Decimation filter is used to band limit the signal before decimation operation. DOWN SAMPLER DECREASES THE SAMPLING RATE BY INTEGER FACTOR(D).

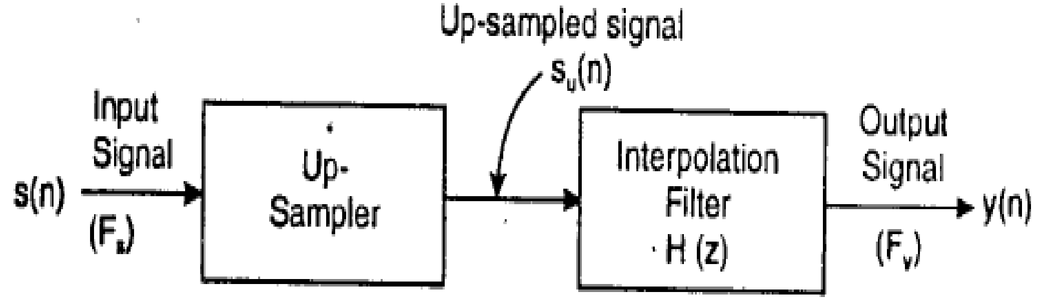
Decimation filter is used to avoid aliasing caused by down-sampling signal $s(n)$.

Prior to down-sampling the signal $s'(n)$ should be band limited

$|\omega| < \frac{\pi}{M}$ by means of a low pass filter (LPF), $H(z)$, CALLED decimator filter.

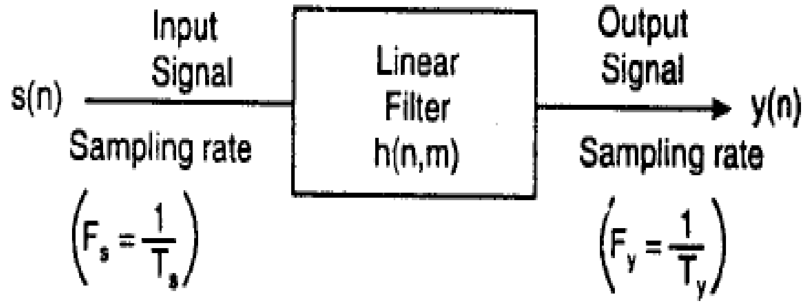
agk-rcet-bhilai

Interpolation. The process of increasing the sampling rate by integer factor (I) is called interpolation of sampling rate. or called up-sampling by factor (I).



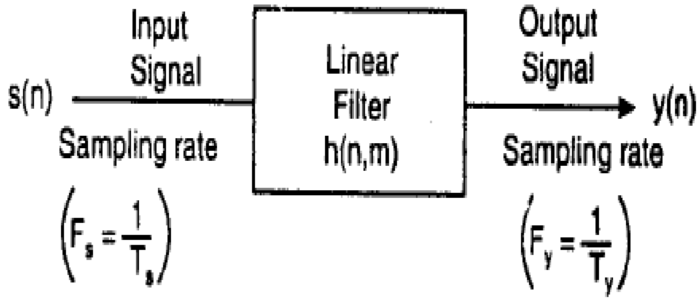
Block diagram of an Interpolator.

The process of sampling rate conversion in the digital-domain can be viewed as a linear filtering operation. It is shown in Fig.



Linear Filter.

agk-rcet-bhilai



input signal $s(n)$ is characterised by sampling rate $F_s = \frac{1}{T_s}$

output signal $y(m)$ is characterised by sampling rate $F_y = \frac{1}{T_y}$.

T_s and T_y are corresponding sampling intervals.

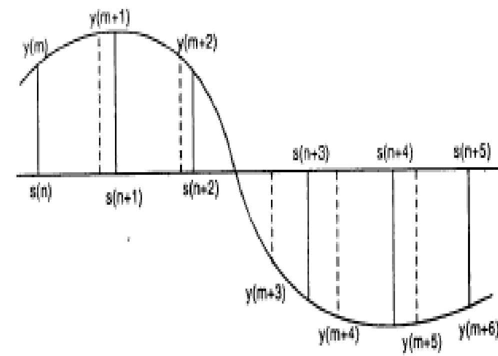
$$\frac{F_y}{F_s} = \frac{\text{Sampling frequency of output signal}}{\text{Sampling frequency of input signal}}$$

$$= \frac{I}{D} = \frac{\text{Prime Integer } (I)}{\text{Prime Integer } (D)}$$

But for the case of ratio $\frac{I}{D}$, both I and D should be prime integer.

Linear filter is characterised by a time-varying impulse response, $h(n, m)$.

Hence the input $s(n)$ and output $y(n)$ are related by convolution sum for time-varying system.



Sampling Rate conversion viewed as linear filtering process.

Where I is the integer factor by which interpolation of sampling rate

D is the integer factor by which decimation of the sampling rate is performed.

Decimation of Sampling Rate By a Integer Factor (D)

The process of reducing the sampling rate of a signal is called Decimation.

Let us assume that the discrete-time signal $s(n)$ with spectrum $S(\omega)$ is to be down sampled by an integer factor D.

The block diagram of decimation process is given in Fig.

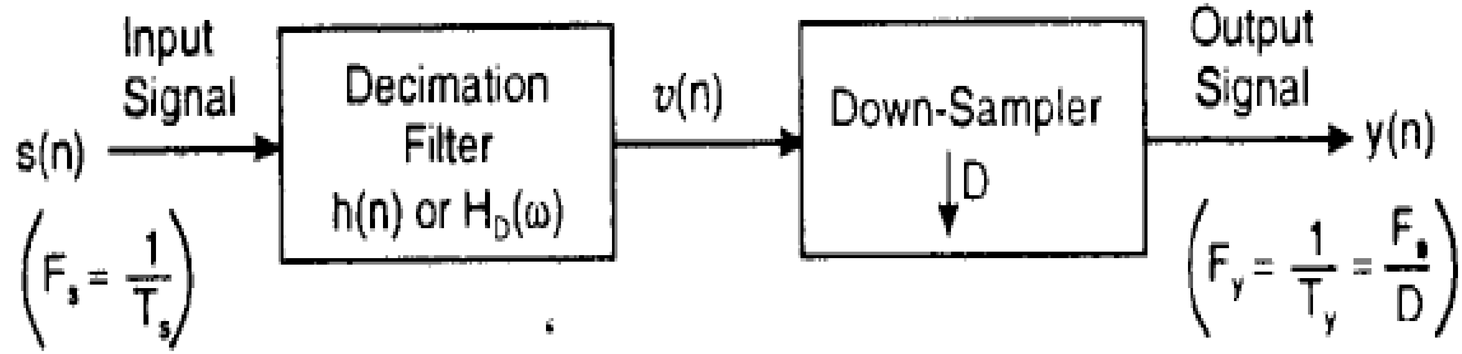
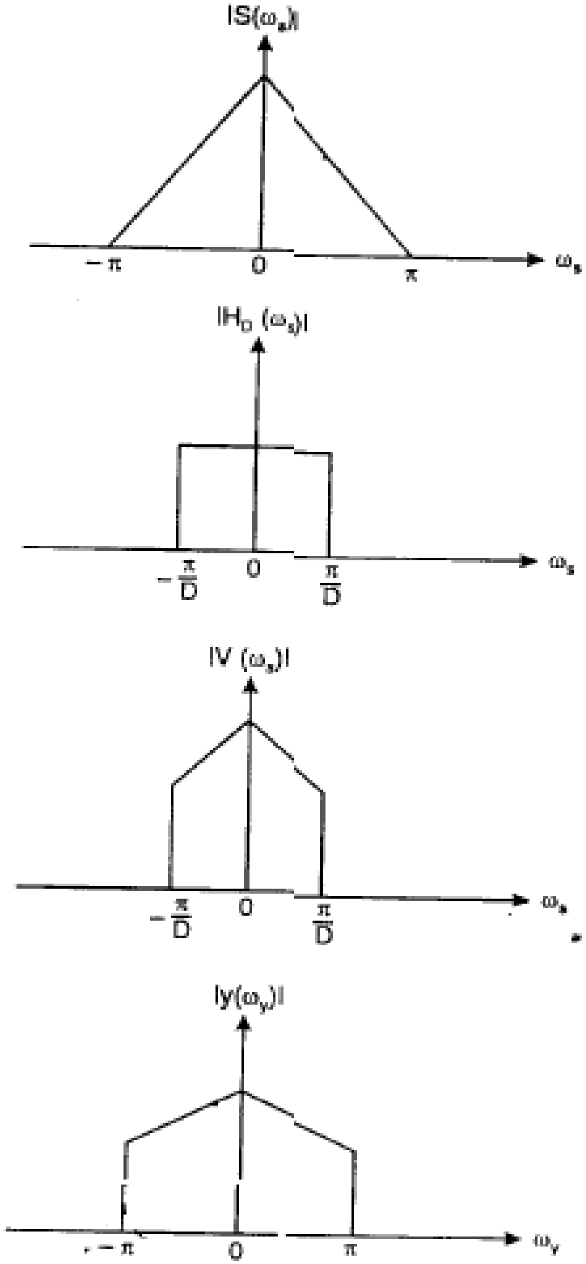


Fig. *Block Diagram of Decimation process.*

agk-rcet-bhilai



Hence, $$Y(\omega_y) = \frac{1}{D} H_D\left(\frac{\omega_y}{D}\right) S\left(\frac{\omega_y}{D}\right)$$
$$= \frac{1}{D} S\left(\frac{\omega_y}{D}\right)$$

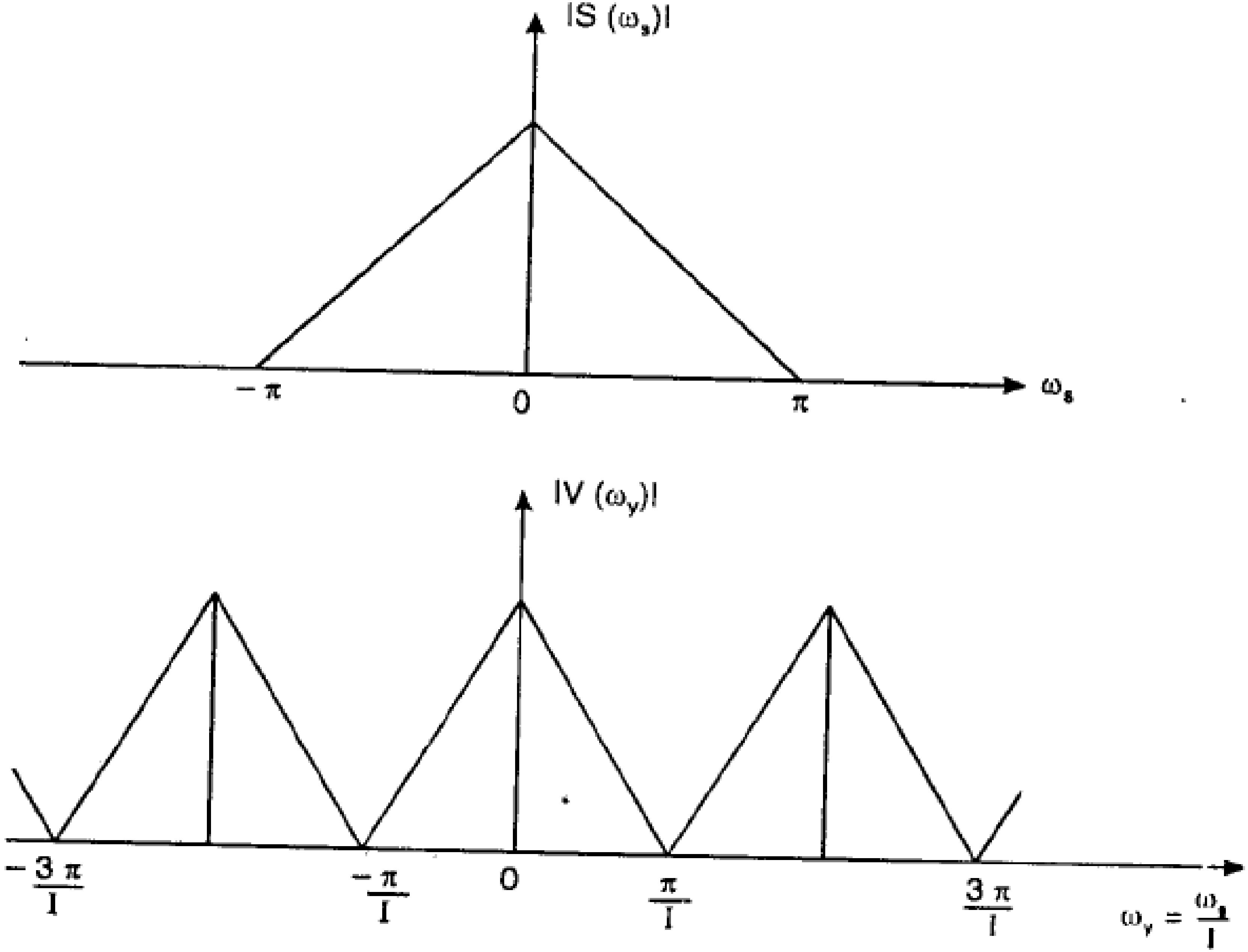
for $0 \le |\omega_y| \le \pi$. The spectra for the sequence $s(n)$, $v(n)$ and $y(m)$ are shown

Spectra of Signals in the decimation of s(n) by a factor D.

INTERPOLATION OF SAMPLING RATE BY A INTEGER FACTOR (I)

Increasing of sampling rate of a signal is called Interpolation. An increase in the sampling rate by an integer factor I can be accomplished by interpolating $(I-1)$ new samples between successive values of the signals. The interpolation process can be accomplished by various types of methods.

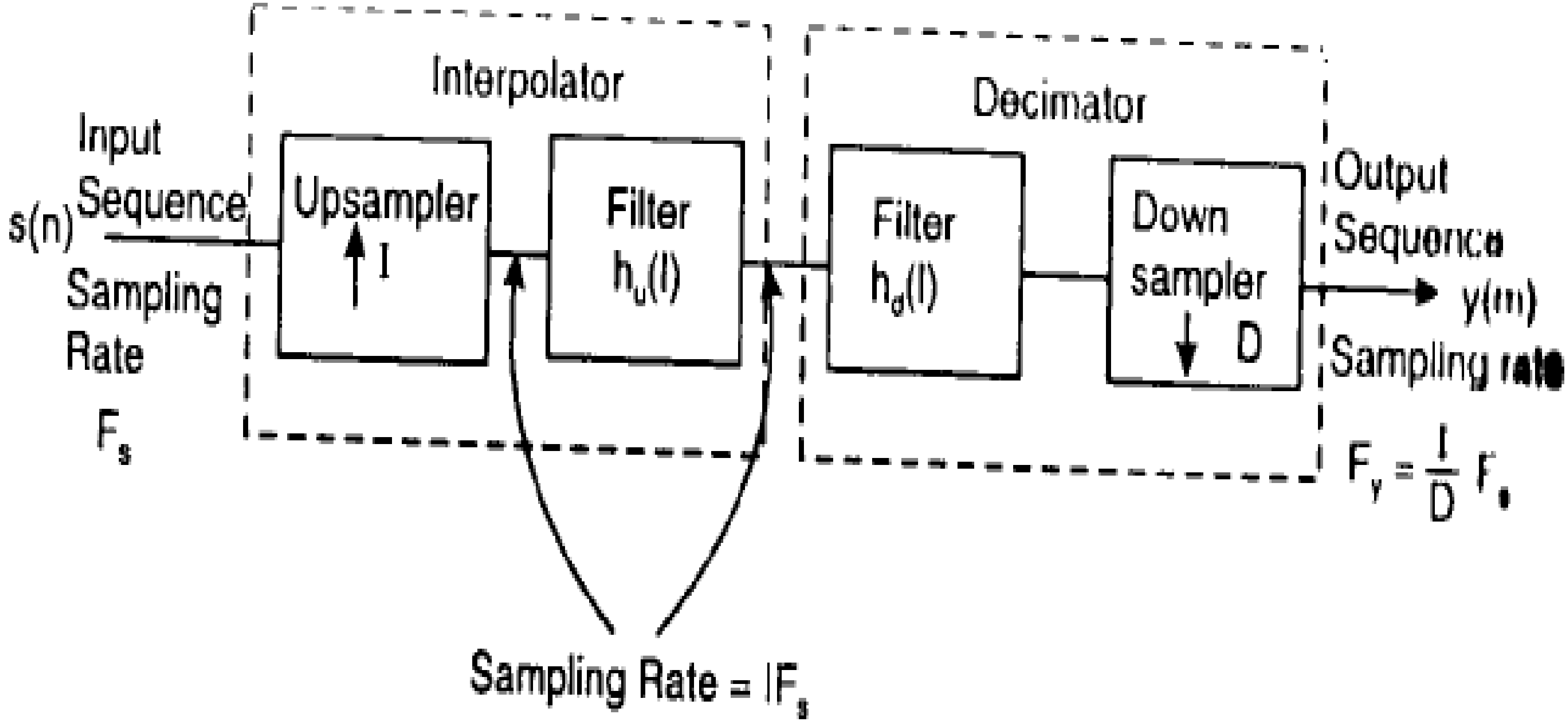
$$y(m) = \sum_{k=-\infty}^{\infty} h(m - kI)\, s(k)$$



|S (ωs)|
− π
0
π
ωs
|V (ωy)|
− 3π/I
− π/I
0
π/I
3π/I
ωy = ωs/I

SAMPLING RATE ALTERNATION OR CONVERSION BY A RATIONAL FACTOR $\left(\frac{I}{D}\right)$

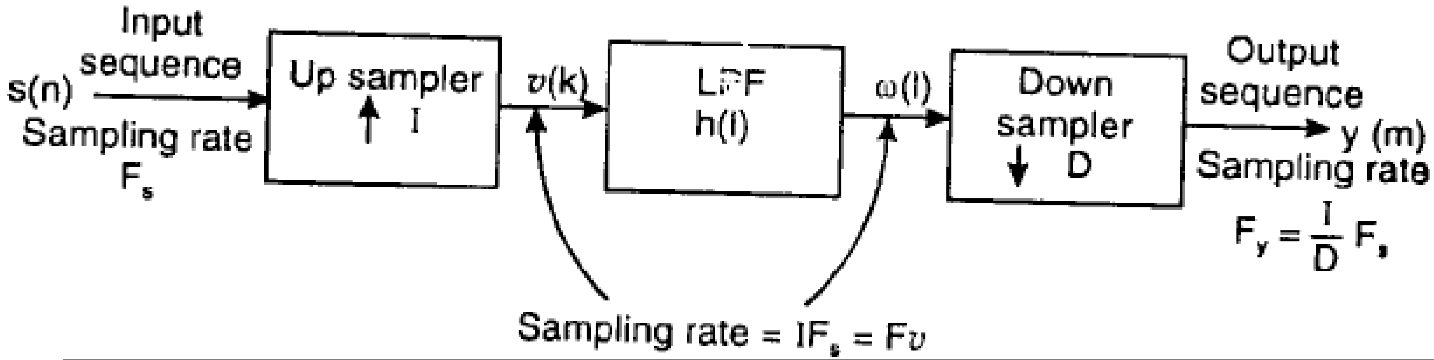
We now consider the general case of sampling rate conversion by first performing interpolation by the factor I and decimating the output of the interpolator by the factor D. In other words, a sampling rate conversion by the rational factor $\frac{I}{D}$ is accomplished by cascading an interpolator with a decimetor. It is illustrated in Fig.



Block diagram of a method for sampling Rate Conversion by a factor $\left(\frac{I}{D}\right)$.

$$H(\omega_v) = \begin{cases} I, & 0 \le |w_v| \text{ min. of } \left(\frac{\pi}{D}, \frac{\pi}{I}\right) \\ o, & \text{otherwise} \end{cases}$$

where $$\omega_v = \frac{2\pi F}{F_v} = \frac{2\pi F}{IF} = \frac{\omega_s}{I}$$



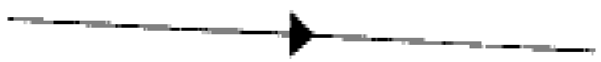
Block diagram of a method for sampling rate conversion by a factor $\left(\frac{I}{D}\right)$. Here two filters $h_u(l)$ and $h_d(l)$ are combined in a single LPF $h(l)$.

In the time-domain, the output of the Up-sampler is given as

$$v(l) = \begin{cases} s\left(\frac{l}{I}\right), & l = 0, \pm I, \pm 2I, \ldots \\ o, & \text{otherwise} \end{cases}$$

SIGNAL FLOW GRAPHS

(a) Unit Delay

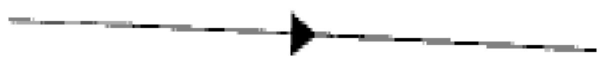


$x(n)$ Z^{-1} $y(n) = x(n-1)$

$X(e^{j\omega})$ $Y(e^{j\omega}) = e^{-j\omega} X(e^{j\omega})$

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

(b) M-sample delay

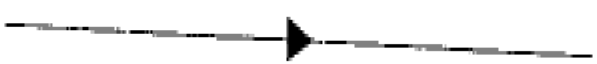


$x(n)$ Z^{-M} $y(n) = x(n-M)$

$X(e^{j\omega})$ $Y(e^{j\omega}) = e^{-jM\omega} X(e^{j\omega})$

$X(z)$ $Y(z) = z^{-M} X(z)$

(c) GAIN (multiplication by a constant)

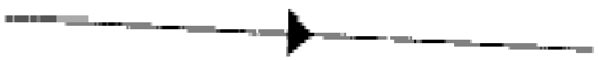


$x(n)$ k $y(n) = kx(n)$

$X(e^{j\omega})$ $Y(e^{j\omega}) = kX(e^{j\omega})$

$X(z)$ $Y(z) = kX(z)$

(d) GAIN AND DELAY by a factor M



$x(n)$ kz^{-M} $y(n) = kx(n-M)$

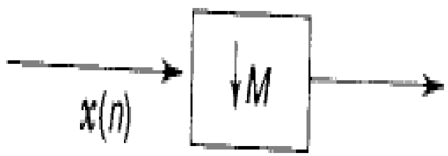
$X(e^{j\omega})$ $Y(e^{j\omega}) = ke^{-jM\omega} X(e^{j\omega})$

$X(z)$ $Y(z) = kz^{-M}X(z)$

SIGNAL FLOW GRAPH ....

Branch Operations in Signal Flow Graphs

(e) Sampling rate compressor / down sampler



$y(m) = x(Mm)$

$X(e^{j\omega})$

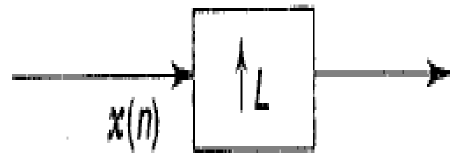
$X(z)$

$$Y(e^{j\omega}) = \frac{1}{M} \sum_{l=0}^{M-1} X(e^{j(\omega - 2\pi l)/M})$$

$$Y(z) = \frac{1}{M} \sum_{l=0}^{M-1} X(z^{1/M} w)$$

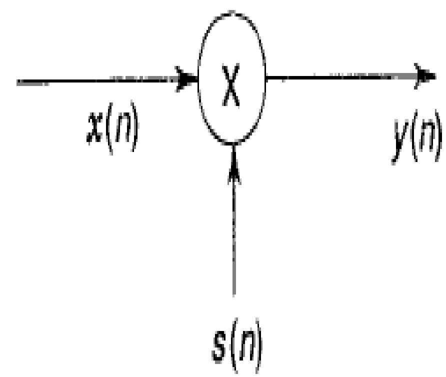
$$w = e^{-j2\pi l/M}$$

(f) Sampling rate expander / up sampler



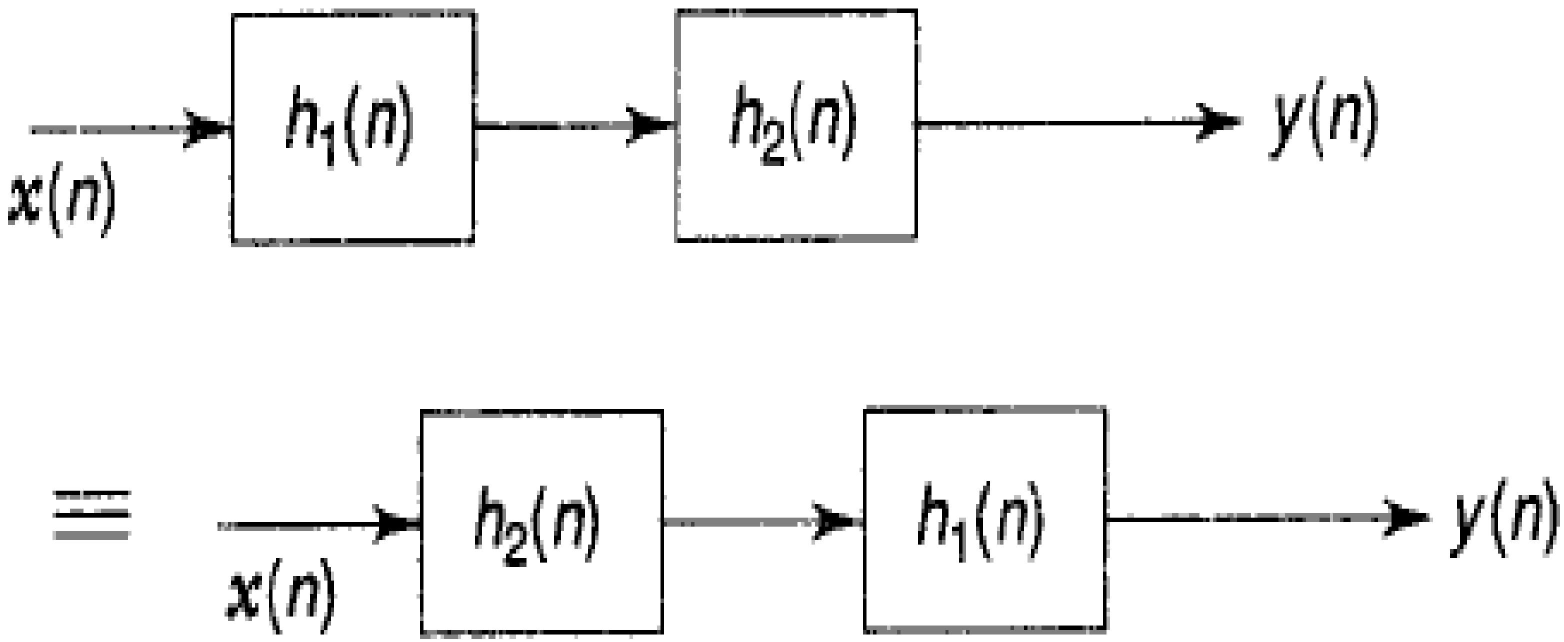
$$y(m) = \begin{cases} x(mL), & m = 0, \pm L \pm 2L, \ldots \\ 0 & \text{otherwise} \end{cases}$$

(g) Modulator



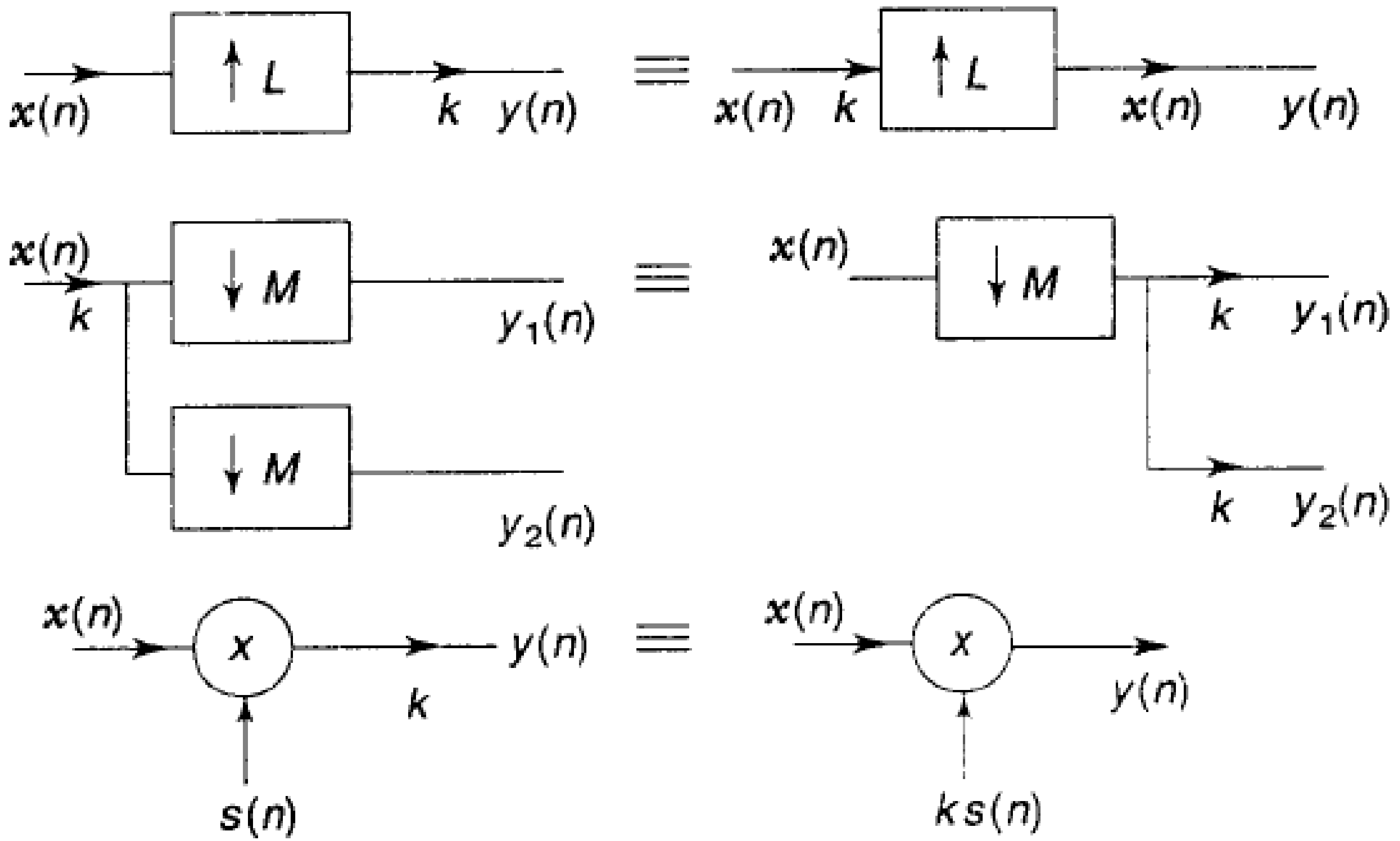
$y(n) = x(n)\, s(n)$

Manipulation of Signal Flow Graphs



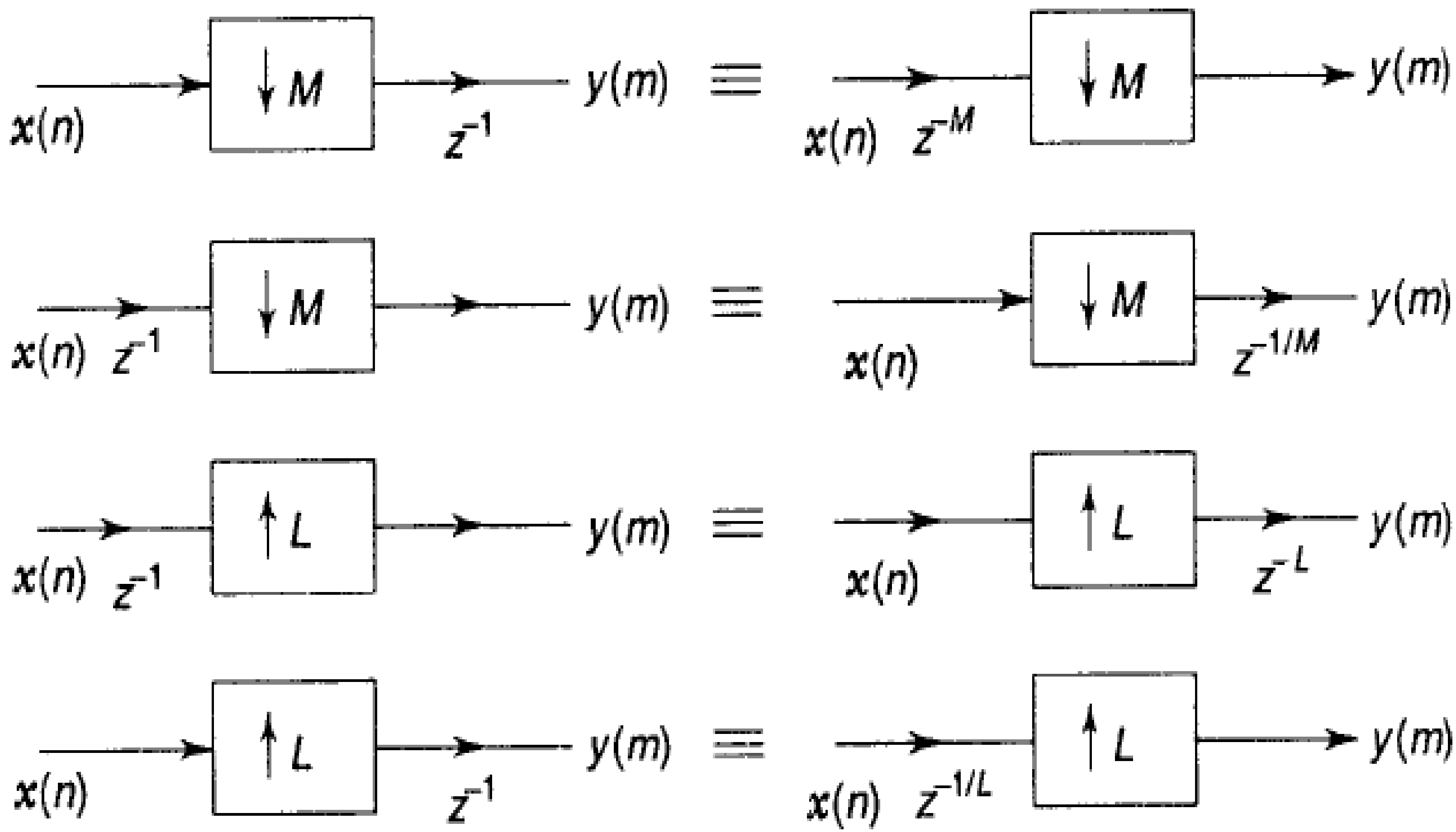
(a) Commutation of LTI systems

Manipulation of Signal Flow Graphs



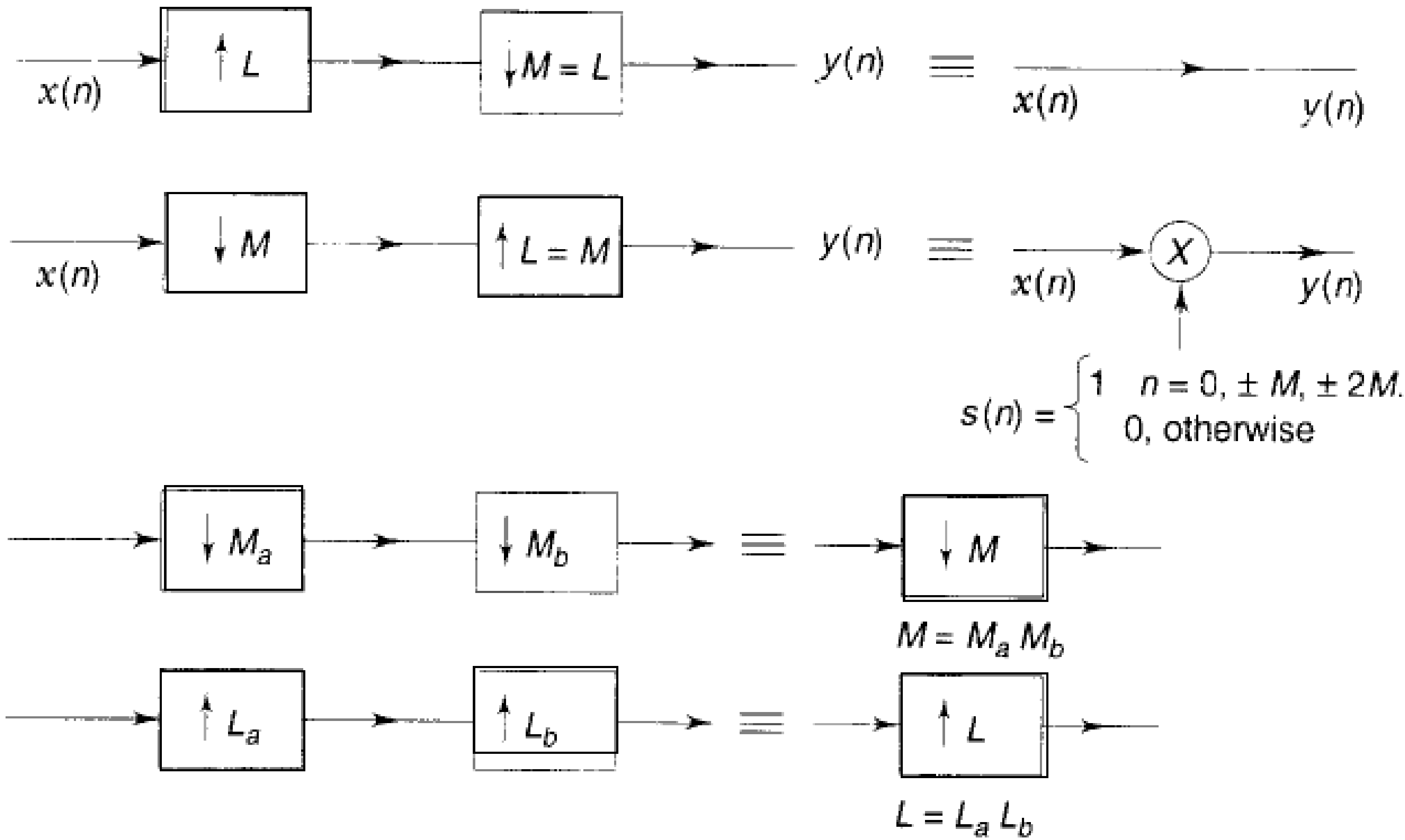
(b) Scalar commutation

Manipulation of Signal Flow Graphs



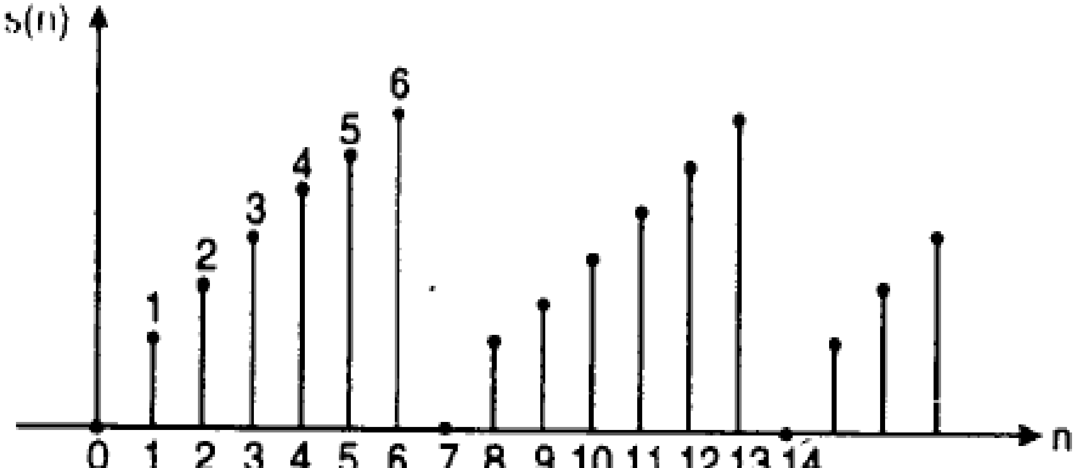
(c) Identites in decimators and interpolators

Manipulation of Signal Flow Graphs



(d) Identifies in cascades of decimators and interpolators

Example 1 Obtain the decimated signal $y(n)$ by a factor 3 from the input signal $s(n)$ shown in Fig.



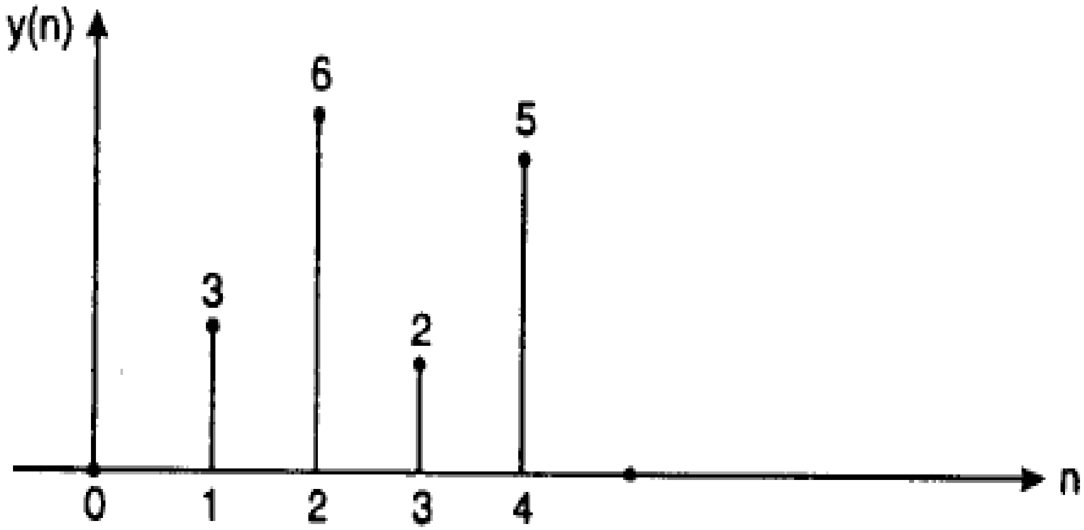
(a)

Example 2 Obtain the two-fold expanded signal $y(n)$ of the input signal $s(n)$.

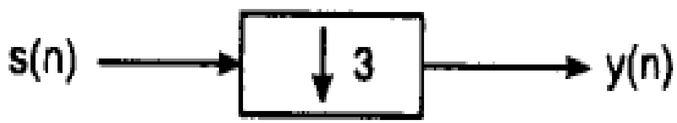
$$s(n) = \begin{cases} n, & n > 0 \\ 0, & \text{otherwise} \end{cases}$$

solution 01:

$y(n) = s(Dn)$, where D is the decimation factor and equal to 3. The decimated signal $y(n)$ is shown in Fig.



(b)



Solution 02:

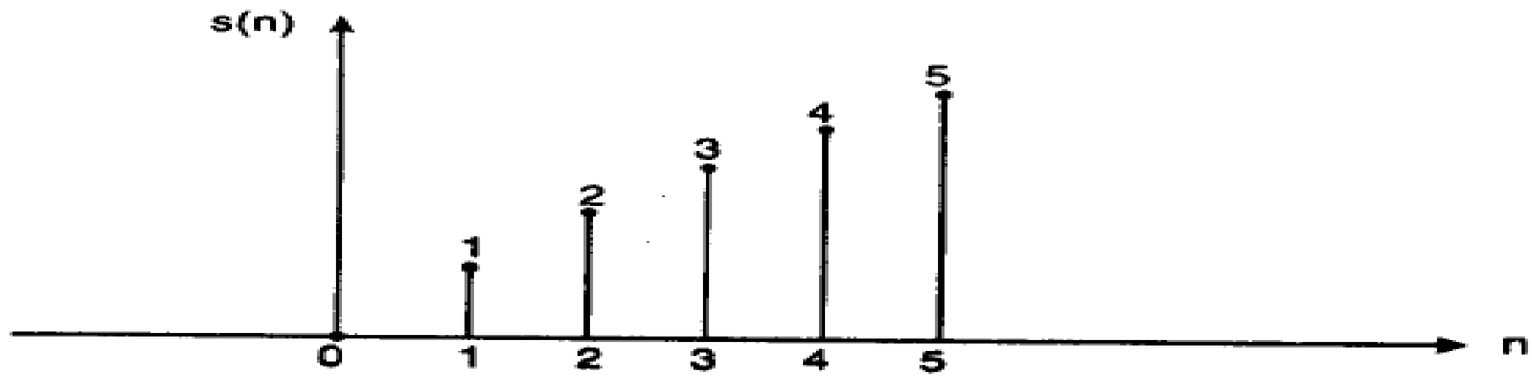
Solution. The output signal $y(n)$ is given by

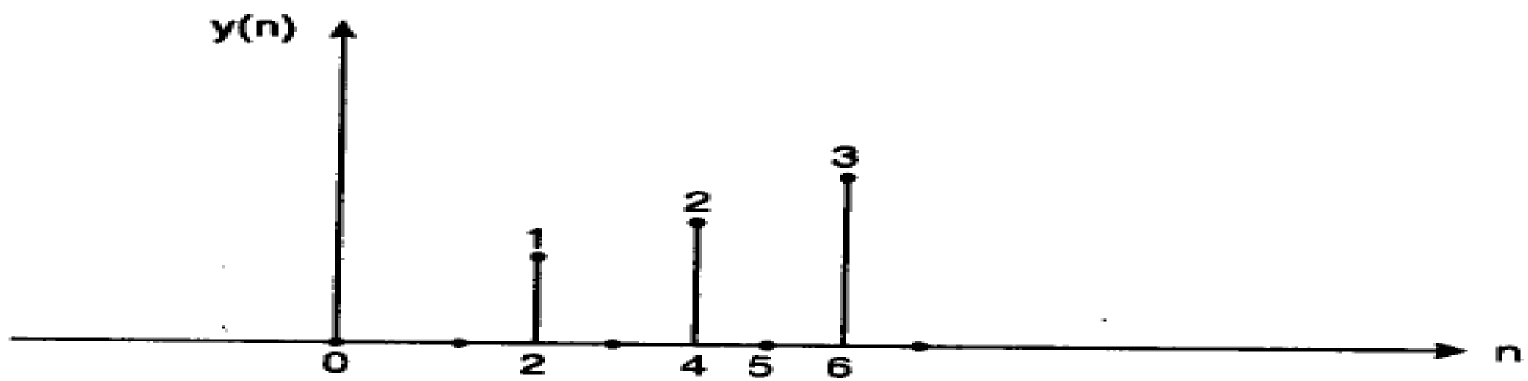
$$y(n) = \begin{cases} s\left(\frac{n}{I}\right), & n = \text{multiples of I} \\ 0, & \text{otherwise} \end{cases}$$

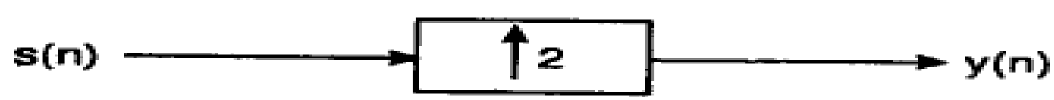
where $I = 2$

$$s(n) = 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ \ldots\ldots .$$
$$y(n) = 0,\ 0,\ 1,\ 0,\ 2,\ 0,\ 3,\ 0,\ 4,\ 0,\ 5,\ 0,\ 6,\ 0,\ \ldots\ldots .$$

In general, to obtain the expanded signal $y(n)$ by a factor I, $(I-1)$ zeros are inserted between the samples of the original signal $s(n)$.

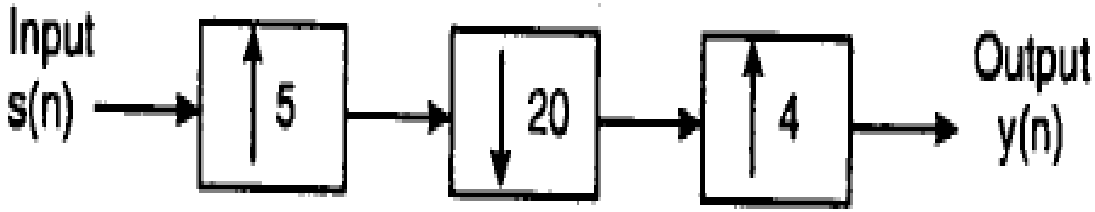






EXAMPLE: 03

Find the expression for the output $y(n)$ in terms of inpu $s(n)$ for the multi sampling rate system given as follows :



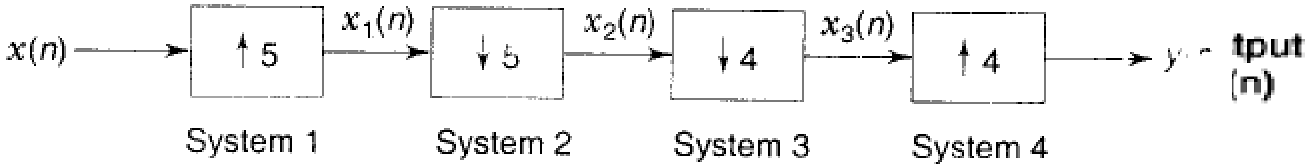
EXAMPLE:04

Find the polyphase decomposition of the *IIR* Digital System with transfer function.

$$H(z) = \frac{1-4z^{-1}}{1+5z^{-1}}$$

Solution:03

Solution. **The decimation with factor 20, can be represented as cascade of two d**‹ system is **‹ given as**



Systems 1 and 2 can be combined. The up-sampler operation of the system 1 is cancelled by the down-sampler operation the system 2.

Therefore, $s_2(n) = s(n)$

Now, Fig. reduces to Fig. **below**

Fig.

Combining systems 3 and 4. The down sampler operation of system 3 is cancelled by the up-sampler operation of system 4.

It means that

$$s(n) = y(n)$$

where
$$s(n) = \begin{cases} 1, & n = 0, \pm 4, \pm 8, \pm 12, \ldots.. \\ 0, & \text{otherwise} \end{cases}$$

Solution:04

$$H(z) = E_0(z^2) + z^{-1} E_1(z^2)$$

where $E_0(z^2)$, $E_1(z^2)$ are polyphase components.

Solution. $$H(z) = H_o(z) + z^{-1} H_1(z) \qquad \text{...(1)}$$

Where $H_o(z)$ and $H_1(z)$ are polyphase components of the IIR Digital system $H(z)$.

$$H(z) = \frac{(1-4z^{-1})}{(1+5z^{-1})} = \frac{(1-4z^{-1})}{(1+5z^{-1})}\frac{(1-5z^{-1})}{(1-5z^{-1})}$$

$$= \frac{1-9z^{-1}+20z^{-2}}{1-25z^{-2}}$$

$$= \left(\frac{1+20z^{-2}}{1-25z^{-2}}\right) + z^{-1}\left(\frac{-9}{1-25z^{-2}}\right) \qquad \text{...(2)}$$

By comparing Eqns. (1) and (2), we get polyphase components of $H(z)$.

$$H_o(z) = \frac{1+20z^{-2}}{1-25z^{-2}}$$

$$H_1(z) = \frac{-9}{1-25z^{-2}}$$

agk-rcet-bhilai

FILTER DESIGN AND IMPLEMENTATION FOR SAMPLING RATE ALTERNATION OR CONVERSION

Sampling rate alternation by a factor $\left(\frac{I}{D}\right)$ can be achieved by first increasing the sampling rate by integer factor I then down-sampling the filtered signal by the integer factor D. Interpolation is accomplished by inserting I-1 zeros between successive values of the input signal $S(n)$. Before down-sampling, interpolated signal is linearly filtered to eliminate the unwanted images of $S(\omega)$. Here, we discuss the design and implementation of the linear filter. We discuss following types of linear filters :

1. Direct-form FIR Digital Filter Structures.
2. Poly-phase Digital Filter Structures.
3. Time-varying Digital Filter Structures.

Direct-form FIR Digital Filter Structures

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

Thus we will have the filter parameters $\{h(k)\}$. These filter parameters allow us to implement the FIR digital filter directly. It is shown in

Although this realization is simple but it is also very inefficient. The inefficiency results from the fact that the Up-sampling process introduces $(I-1)$ zeros between successive points of the input signal $s(n)$

If I is large, most of the signal components in the FIR digital filter are zero. Consequently, most of the multiplications and additions result in zeros. Furthermore, the down-sampling process at the output of the filter implies that only one out of every D output samples is required at the output of the filter. Consequently, only one out of every D possible values at the output of the filter should be computed.

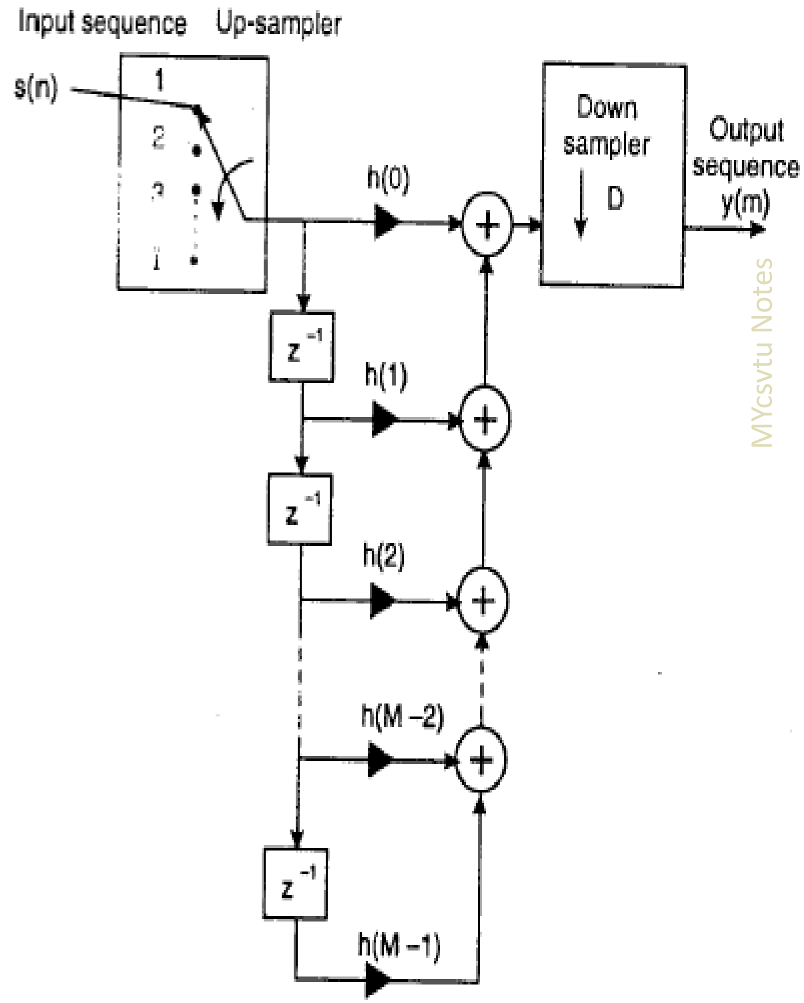
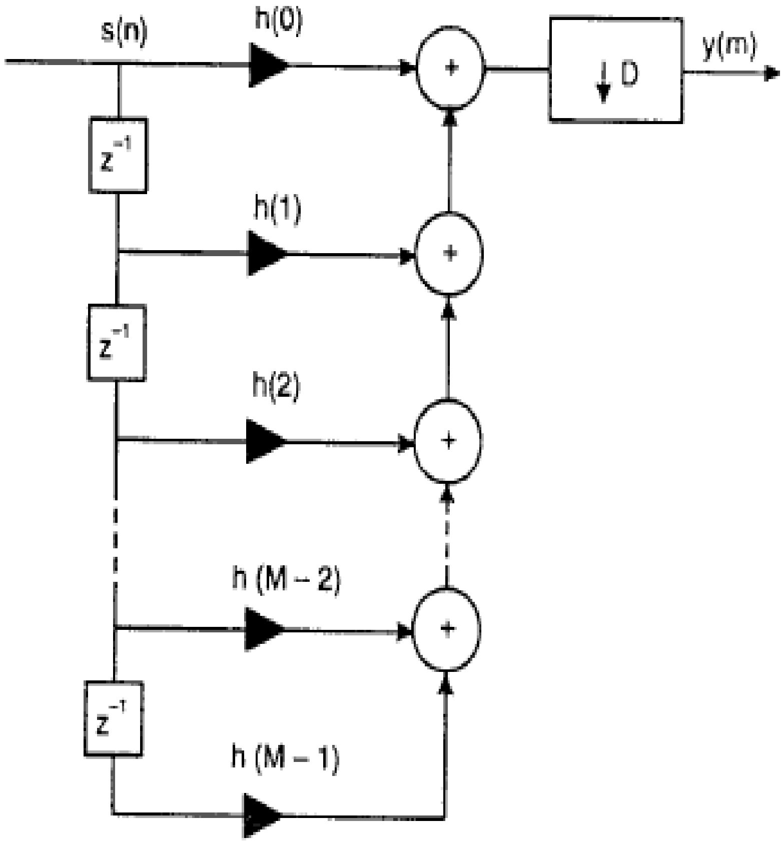
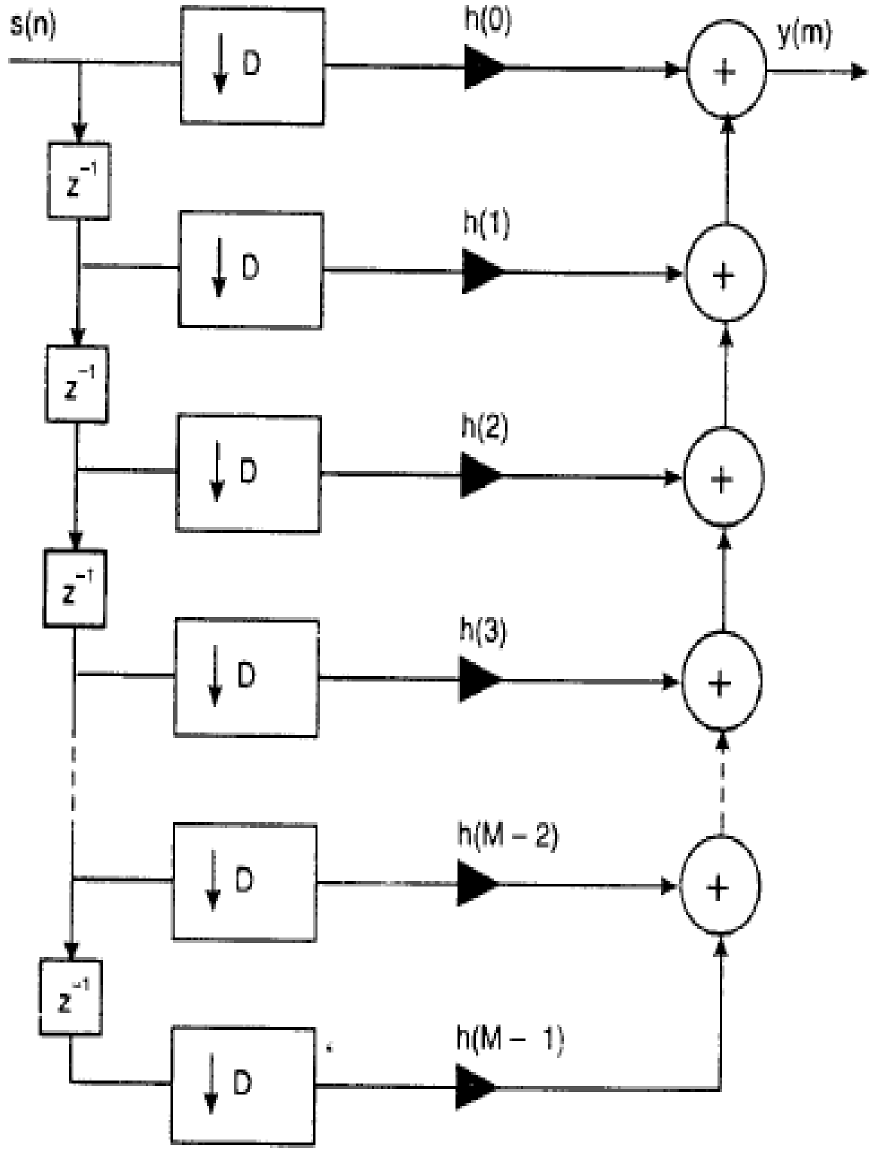


Fig. *Direct-form realization of FIR Digital Filter in Sampling rate conversion by factor* $\frac{I}{D}$.

Decimation of a signal by a factor D.

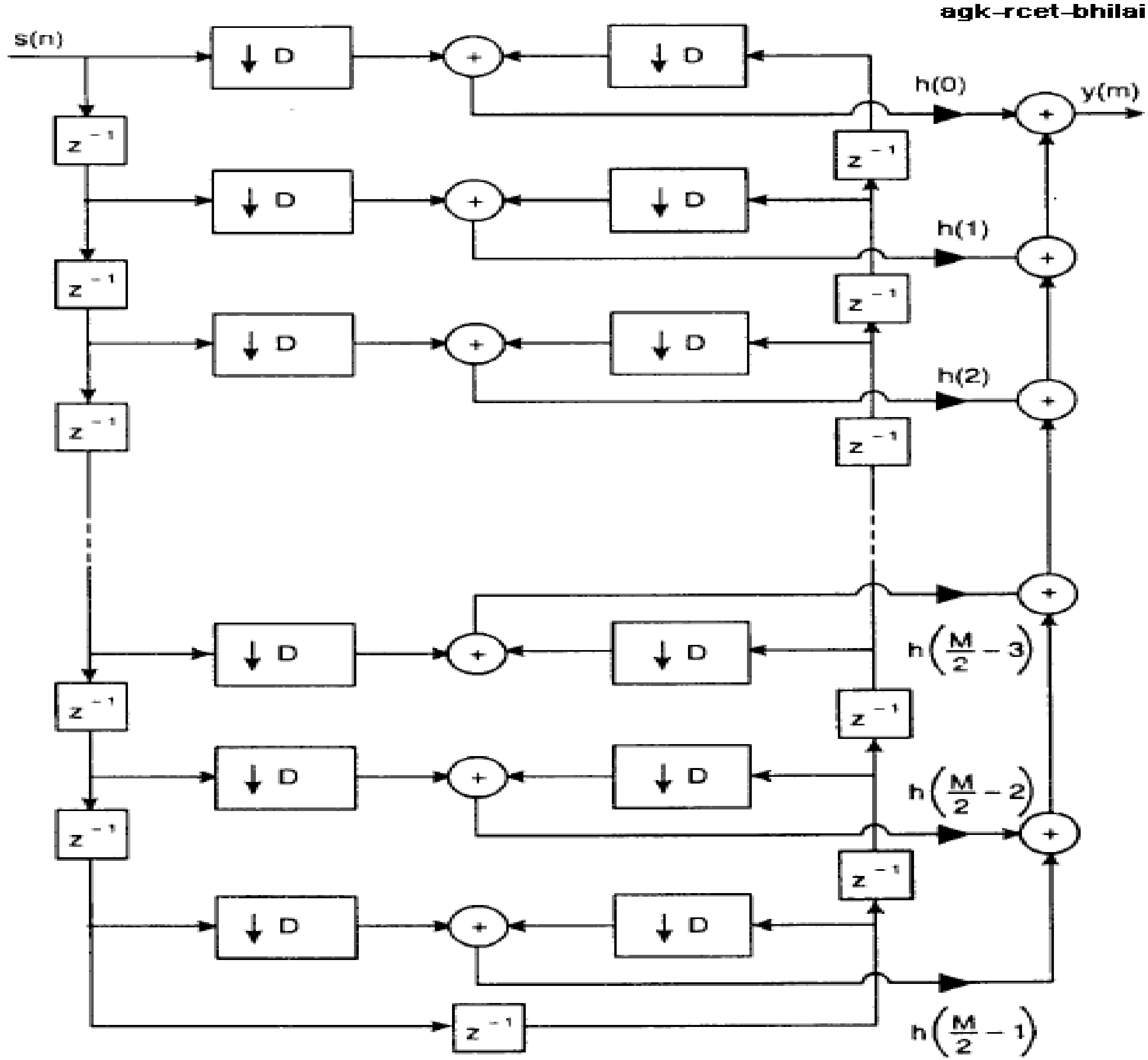


(a)



(b)

agk-rcet-bhilai



Efficient realisation of a decimator. This decimator exploits the property of symmetry in the FIR digital filter.

Efficient Implementation of an Interpolator

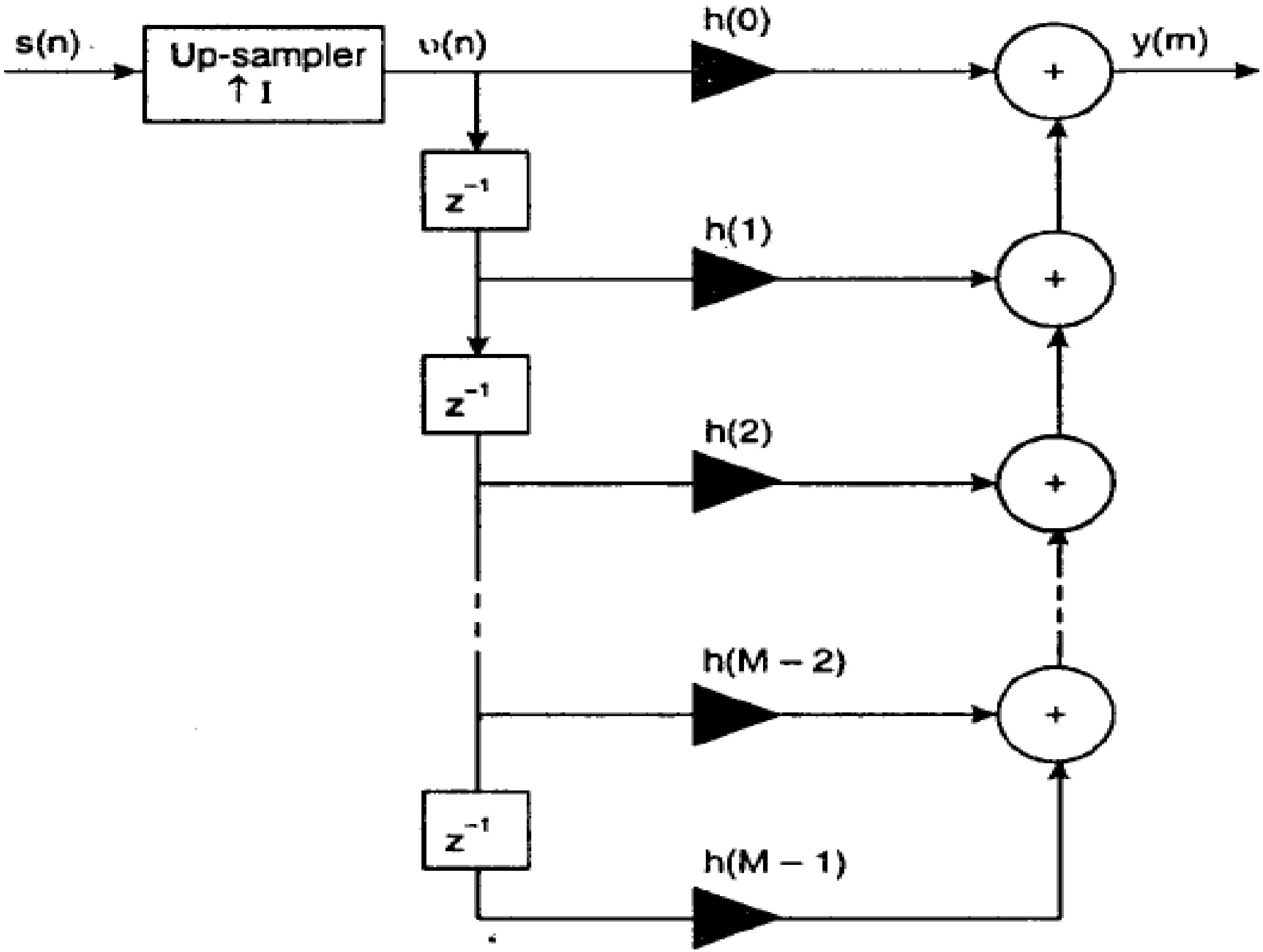
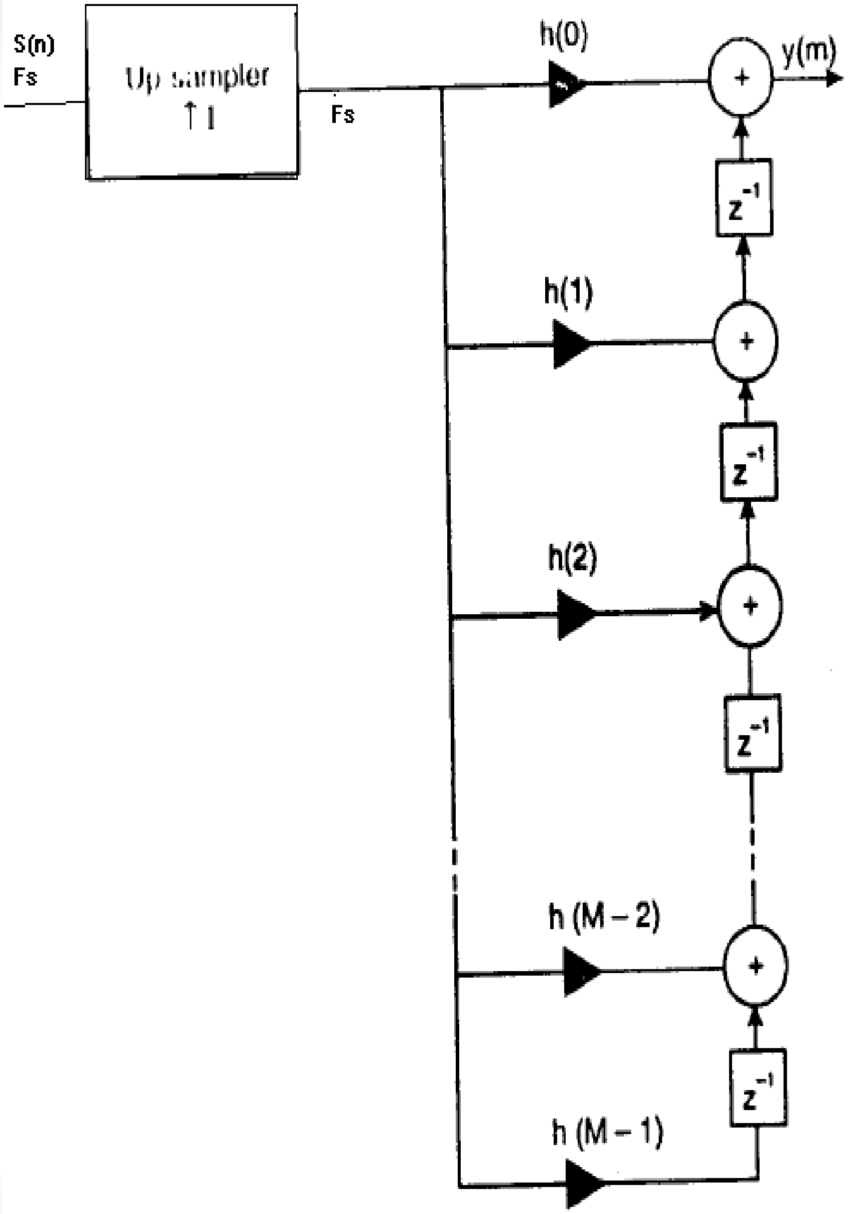


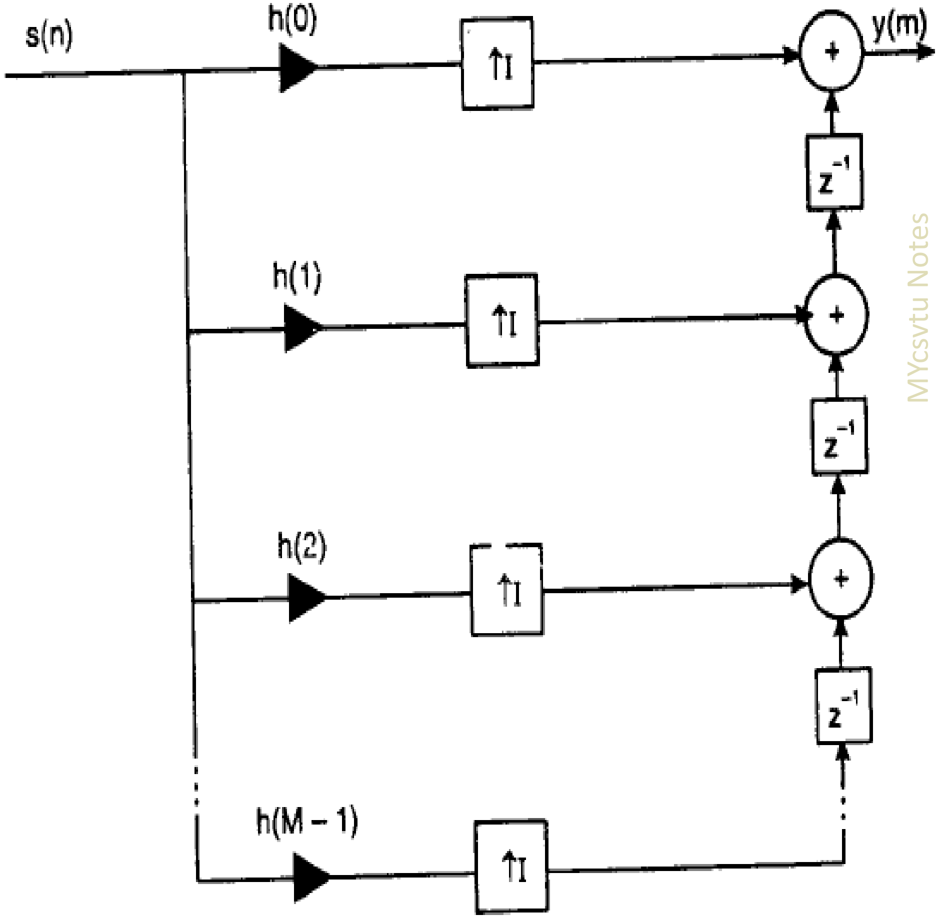
Fig. *Direct Form Realisation of FIR filter in interpolation by a factor I.*

agk–rcet–bhilai

Efficient Realisation of Interpolator.



(a)



(b)

agk-rcet-bhilai

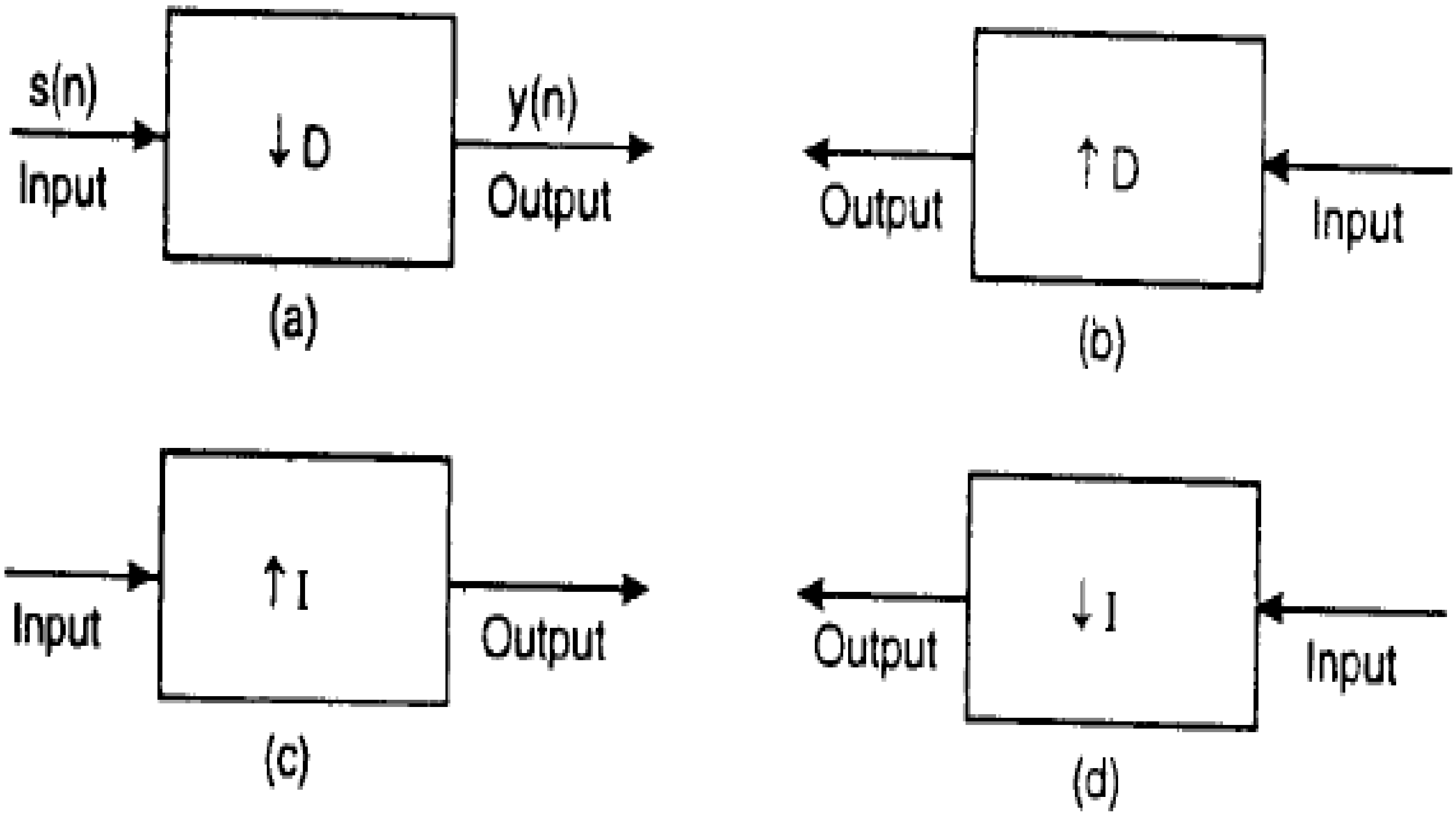


Fig. *Duality relationships obtained through transposition.*

agk-rcet-bhilai

Polyphase Filter Structure

The computational efficiency of the filter structure given in Fig. can also be achieved by reducing the large *FIR* digital filter of length M into a set of smaller filters of length $K = \frac{M}{I}$, where M is selected to be a multiple of I.

$$p_k(n) = h(k + nI),\ k = 0, 1, 2, \ldots, I - 1$$
$$n = 0, 1, 2, \ldots, K - 1$$

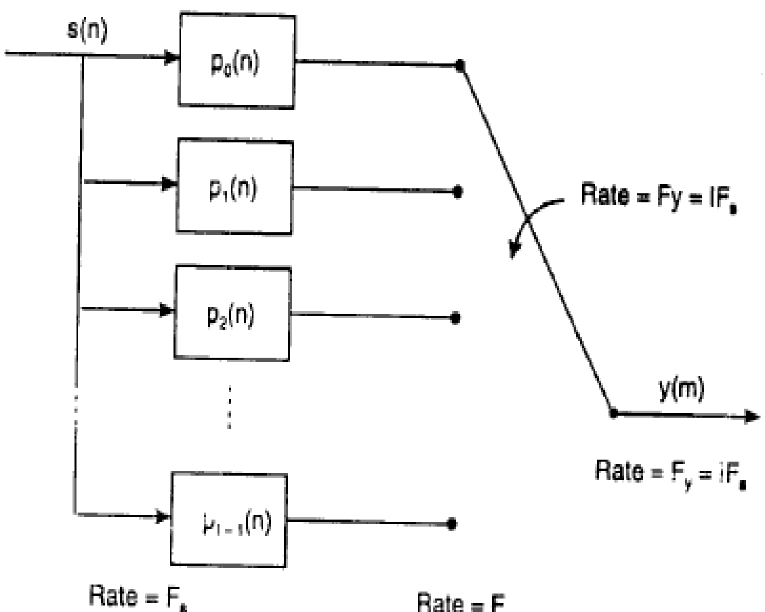


Fig. (a) *Interpolation by use of polyphase filters.*

$$p_k(n) = h(k + nD),\ k = 0, 1, 2, \ldots, D - 1$$
$$n = 0, 1, 2, \ldots, K - 1$$

Where $K = \frac{M}{D}$ is an integer when M is selected to be a multiple of D.

$$p_k(n) = h(nI - k),\ k = 0, 1, 2, \ldots, I - 1$$
$$p_k(n) = h(nD - k),\ k = 0, 1, 2, \ldots, D - 1$$

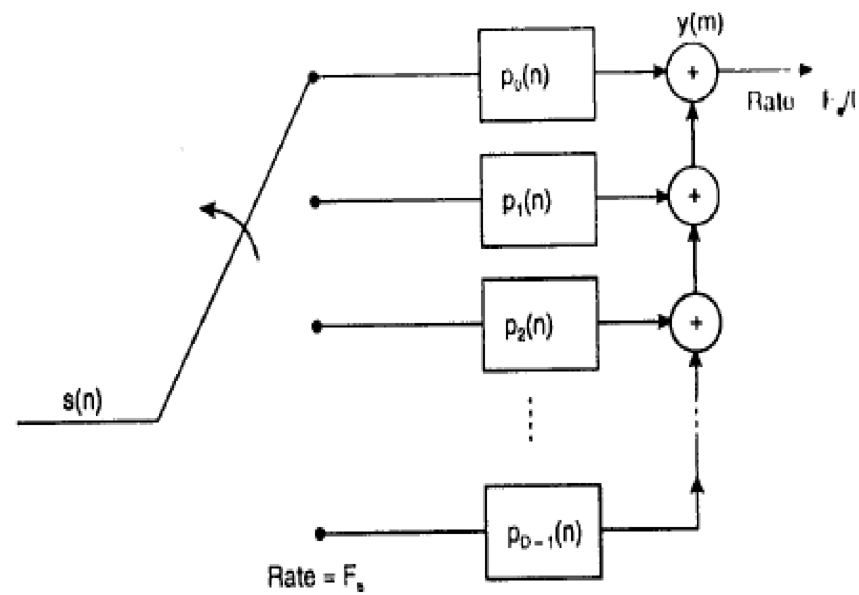


Fig. (b) *Decimation process by use of polyphase filters.*

POLYPHASE DECOMPOSITION

The z–transform of a filter with impulse response $h(n)$ is given by

$$H(z) = h(0) + z^{-1} h(1) + z^{-2} h(2) + \ldots$$

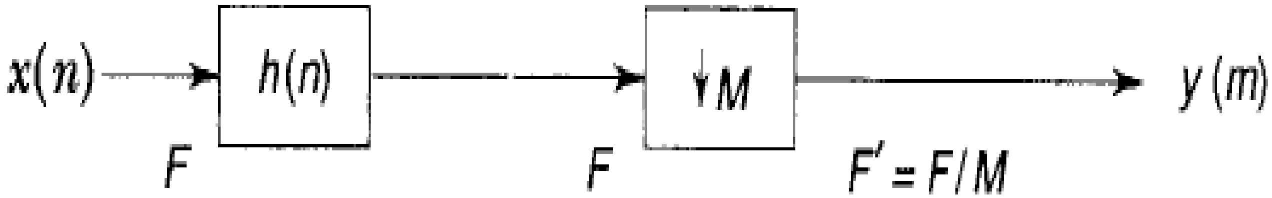
Rearranging the above equation we get,

$$H(z) = h(0) + z^{-2} h(2) + z^{-4} h(4) + \ldots$$
$$+ z^{-1} (h(1)) + z^{-2} h(3) + z^{-4} h(5) + \ldots)$$

Type I Polyphase Decomposition

$$H(z) = E_0(z^2) + z^{-1} E_1(z^2)$$

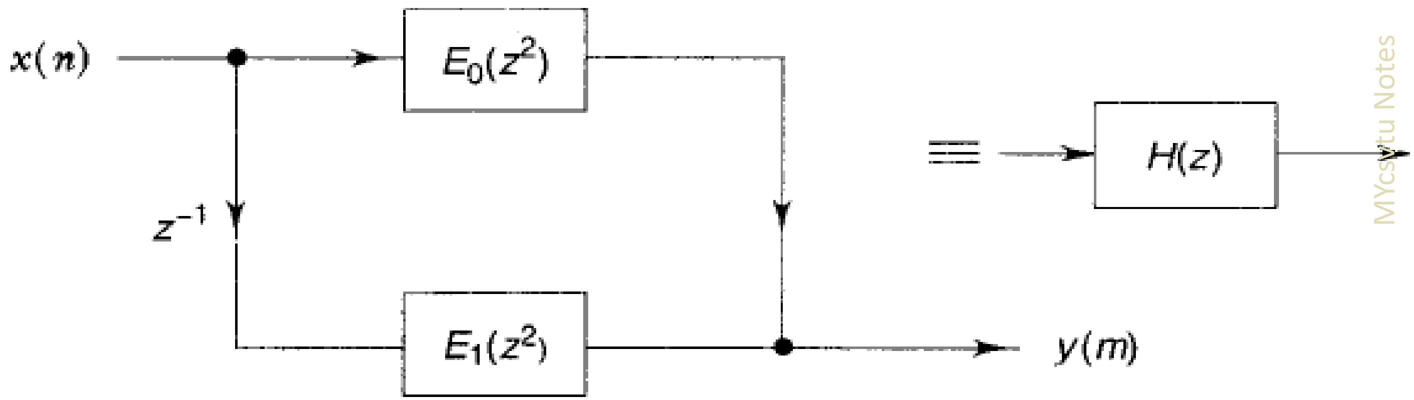
where $E_0(z^2)$ and $E_1(z^2)$ are polyphase components for a factor of



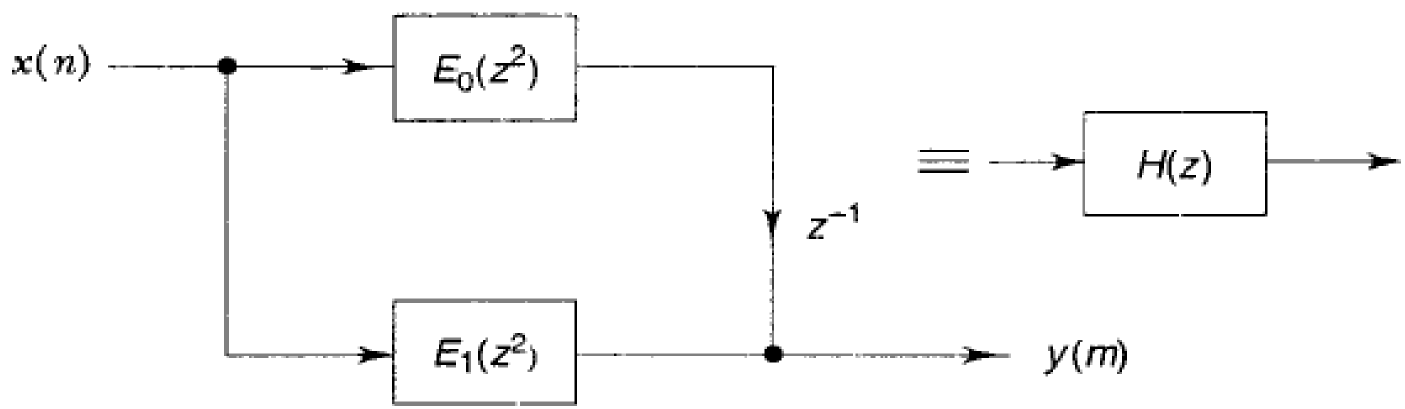
Type II Polyphase Decomposition

$$H(z) = z^{-1} R_0(z^2) + R_1(z^2)$$

where $R_0(z^2)$ and $R_1(z^2)$ are polyphase components for a factor two.



(b) Type 2: Polyphase decomposition



Time-varying Digital Filter Structures

$$g(n, m) = h[nI - (mD)_I]$$

$$y(m) = \sum_{n=0}^{K-1} g\left[n, m - \left\lfloor \frac{m}{I} \right\rfloor I\right] S\left[\left\lfloor \frac{mD}{I} \right\rfloor - n\right]$$

$$g\left[n, m - \left\lfloor \frac{m}{I} \right\rfloor I\right], n = 0, 1, 2, \ldots, K-1.$$

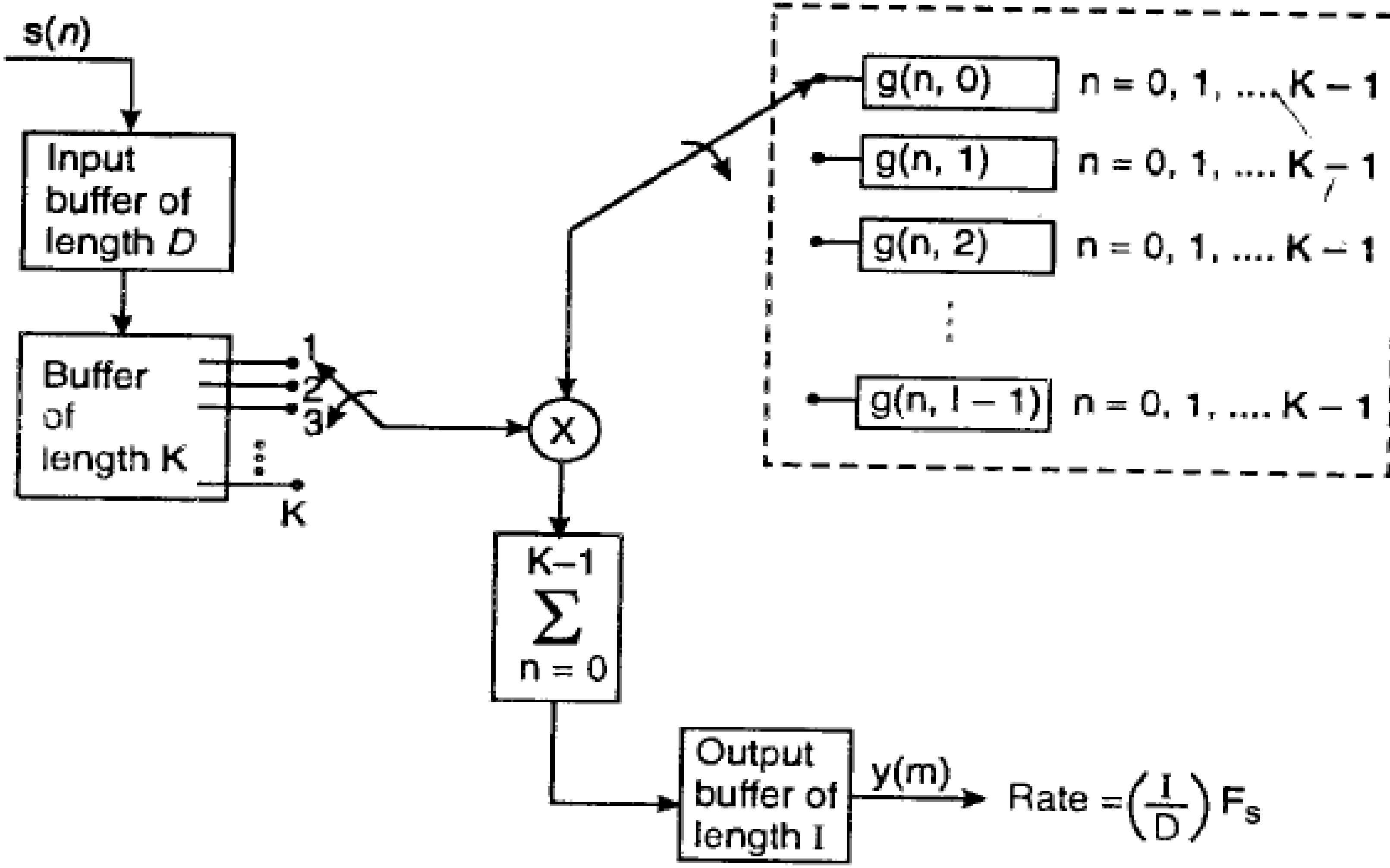


Fig. *Efficient Implementation of Sampling Rate conversion by block processing.*

agk-rcet-bhilai

Frequency response of above filter is given by

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

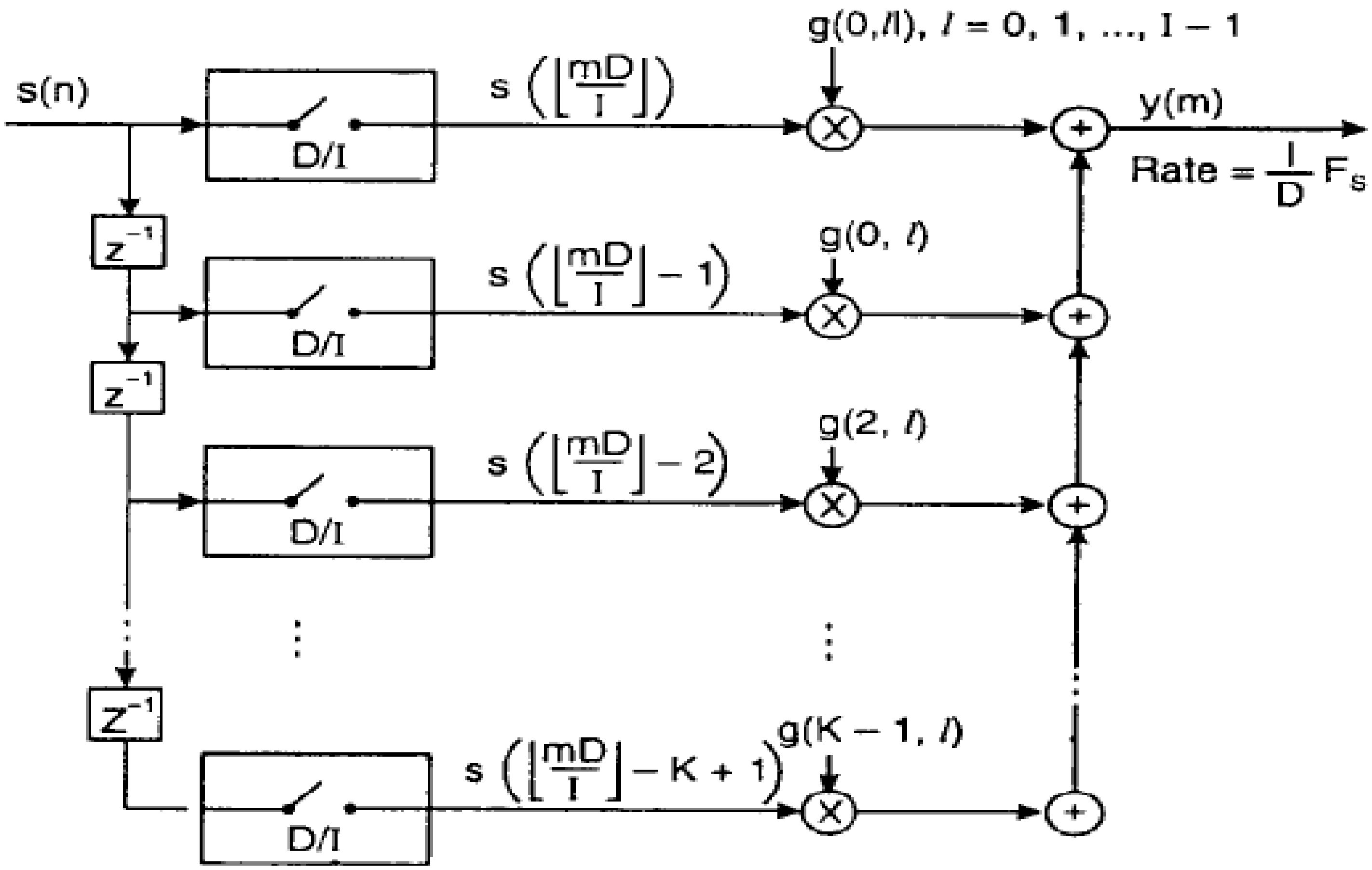


Fig Efficient realization of sampling rate coversion by a factor $\left(\frac{I}{D}\right)$

SAMPLING-RATE CONVERSION BY AN ARBITRARY FACTOR

Case - I : We need to perform sampling rate conversion by a rational numbers $\left(\frac{I}{D}\right)$, , where I is large integer $\left[\text{For example, } I = 1023 \text{ and } D = 511, \textit{i.e.}, \frac{I}{D} = \frac{1023}{511}\right]$. Although we can achieve exact sampling rate conversion by this number, we would use a polyphase filter with $I = 1023$ subfilters, such an exact implementation is obviously inefficient in memory usage because we need to store a large number of filter coefficients.

Case II : In some applications, the exact conversion sampling rate is not known when we design the sampling rate convertor, or sampling rate is continuously changing during the conversion process. For example, we may counter the situation where the input and output samples are controlled by two independent clocks. It is possible to define a nominal conversion rate that is a rational number, the actual sampling rate would be slightly difficult. Sampling rates depend on the frequency difference between the two clocks obviously, it is not possible to design an exact sampling rate converter in this case.

agk-rcet-bhilai

First-order Approximation Method

In general case, we can express $\frac{1}{R_a}$ as

$$\frac{1}{R_a} = \frac{k}{I} + \beta$$

where k and I are positive integers and β is a number in the range.

$$0 < \beta < \frac{1}{I}$$

Now, the boundaries of $\frac{1}{R_a}$ is given as

$$\frac{k}{I} < \frac{1}{R_a} < \frac{k+1}{I}$$

Where I corresponds to interpolation factor. Interpolation factor I will be determined to satisfy the specification on the amount of tolerable distortion introduced by sampling rate converter. Also, I is equal to the number of polyphase filters.

For example

Suppose that $R_a = 2.2$ and that we have determined, as we will demonstrate, that $I = 6$ polyphase filters are required to meet the distortion specification. Then

$$\frac{k}{I} \equiv \frac{2}{6} < \frac{1}{R_a} < \frac{3}{6} \equiv \frac{k+1}{I}$$

So that $k = 2$.

The time spacing between samples of the interpolated sequence is $\frac{T_s}{I}$. However, the desired conversion rate $R_s = 2.2$ for $I = 6$ corresponds to decimation factor of 2.727, which falls between $k = 2$ and $k = 3$.

In the first-order approximation method, we achieve the desired decimation rate by selecting the output-sample from the polyphase filter closest in time to the desired sampling time. It is illustrated in Fig. for $I = 6$.

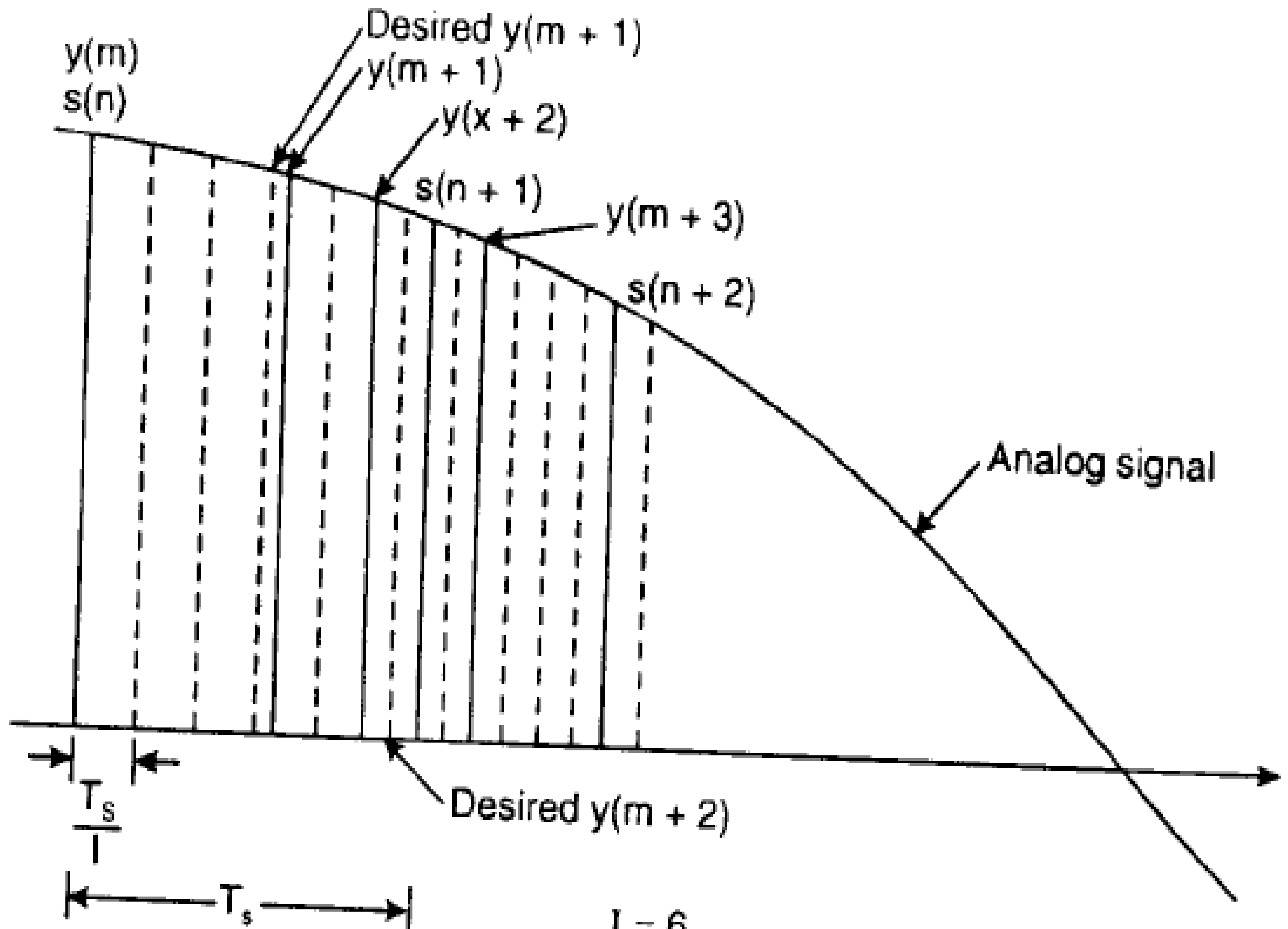


Fig. *Sampling rate rconversion by use of first order approximations.*

agk-rcet-bhilai

$$P = \frac{1}{2\pi}\int_{-\omega_s}^{\omega_s} |S(\omega)|^2 \, d\omega = \frac{A^2\omega_s}{\pi}$$

$$e^{j\omega\tau} - e^{j\omega(\tau - t_m)} = e^{j\omega\tau}(1 - e^{-j\omega t_m})$$

$$= e^{j\omega\tau}[1 - \cos\omega t_m + j\sin\omega t_m] \approx je^{j\omega\tau}\omega t_m$$

By using the bound $|t_m| \le \frac{0.5}{I}$, we obtain an upper bound for the total error power as

$$P_e = \frac{1}{2\pi}\int_{-\omega_s}^{\omega_s} \left| S(\omega)e^{j\omega\tau} - S(\omega)e^{j\omega(\tau - t_m)} \right|^2 d\omega$$

$$\approx \frac{1}{2\pi}\int_{-\omega_s}^{\omega_s} \left| S(\omega)\, je^{j\omega\tau}\omega t_m \right|^2 d\omega$$

$$\le \frac{1}{2\pi}\int_{-\omega_s}^{\omega_s} A^2\left(\frac{0.5}{I}\right)^2 \omega^2 d\omega$$

$$= \frac{A^2\omega_s^2}{12\pi I^2}$$

$$SDR_1 = \frac{P}{P_e} \ge \frac{12I^2}{\omega_s^2}$$

Second-order Approximation Method

The disadvantage of the first order approximation method is the large number of subfilters needed to achieve a specified distortion requirement.

The implementation of the linear interpolation method is very similar to the first-order approximation. In linear interpolation, we compute two adjacent samples with the desired sampling time falling between their sampling times. It is illustrated in Fig. (12.22). But in first-order approximation, we use the sample from the interpolating filter closest to the desired conversion output as the approximation. The normalised time spacing between these two samples is $\frac{1}{I}$.

We assume that the sampling time of the first sample lags the desired sampling time by t_m, the sampling time of the second sample is then leading the desired sampling time by $\left(\frac{1}{I}\right) - t_m$. If the two samples are denoted by $y_1(m)$ and $y_2(m)$ then by using linear interpolation, we can compute the approximation to the desired output as

$$y(m) = (1-\alpha_m)\, y_1(m) + \alpha_m y_2(m)$$

where $\alpha_m = It_m$.

Note that $0 \le \alpha_m \le 1$

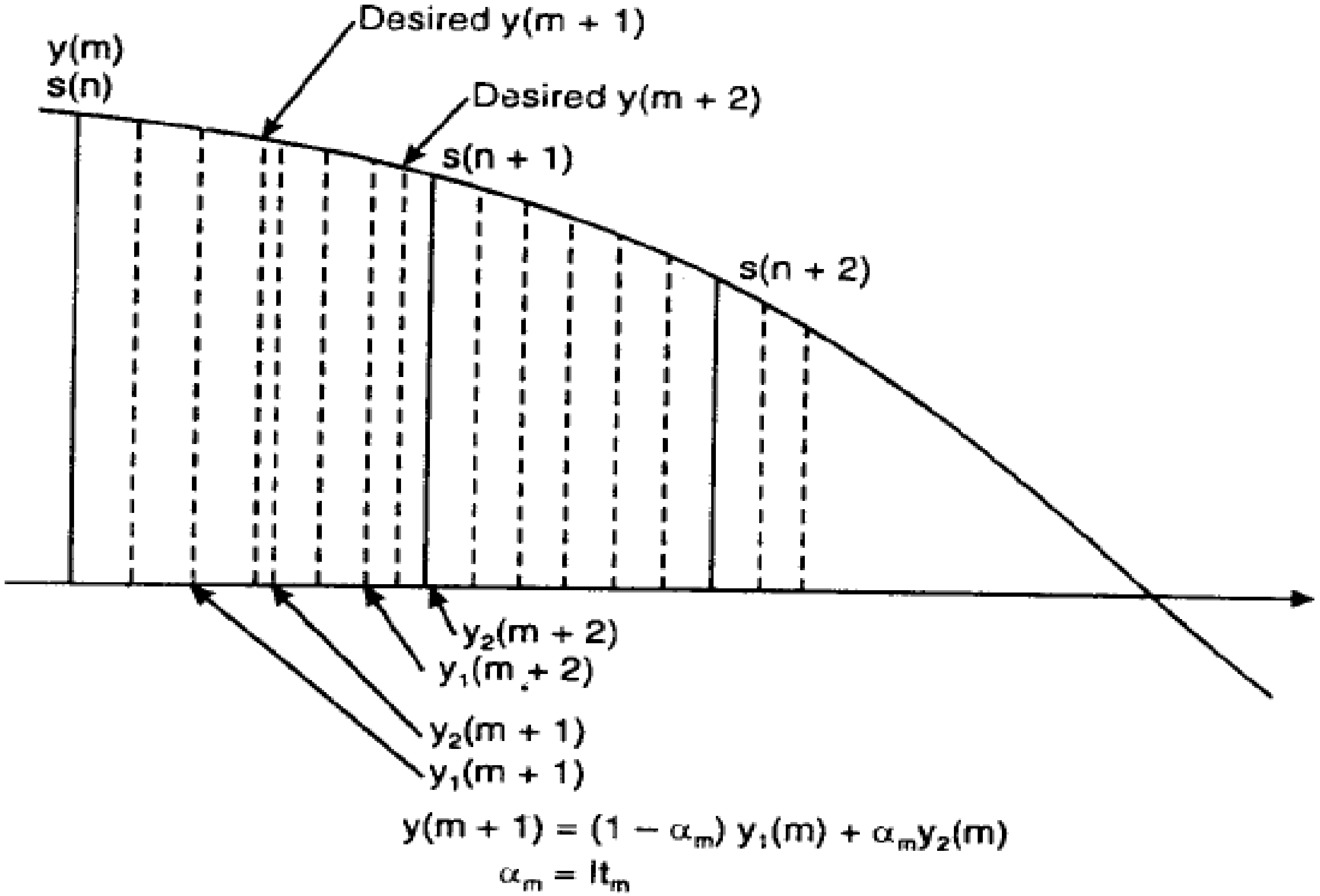


Fig. *Sampling rate conversion by use of linear interpolation or IInd-order approximation.*

PROVE THAT

$$P_e \leq \frac{1}{2\pi}\int_{-\omega_s}^{\omega_s} A^2\left(\frac{0.25}{I^2}\right)^2 \omega^4 d\omega = \frac{A^2(\omega_s)^5}{80\,\pi\, I^4}$$

The frequency responses of the desired filter, first-subfilter and second subfilter is $e^{j\omega\tau}$, $e^{j\omega(\tau-t_m)}$, and $e^{j\omega(t-t_m+\frac{1}{I})}$, respectively. Because linear interpolation is a linear operation, we can also use linear interpolation to compute the frequency response of the filter that generates $y(m)$ as

$$(1-It_m)\, e^{j\omega(\tau-t_m)} + It_m e^{j\omega(\tau-t_m+\frac{1}{I})}$$

$$= e^{j\omega\tau}\left[(1-\alpha_m)\, e^{-j\omega t_m} + \alpha_m e^{j\omega(-t_m+\frac{1}{I})}\right.$$

$$= e^{j\omega\tau}[1-\alpha_m][\cos\omega t_m - 1\sin\omega t_m]$$

$$+ e^{j\omega\tau}\alpha_m\left[\cos\omega\left(\frac{1}{I}-t_m\right) + j\sin\omega\left(\frac{1}{I}-t_m\right)\right].$$

By ignoring higher order errors, we can write Eqn. as

By ignoring higher order errors, we can write Eqn. as

$$e^{j\omega\tau} - (1-\alpha_m)e^{j\omega(\tau - t_m)} - \alpha_m\, e^{j\omega(t - t_m + \frac{1}{I})}$$

$$= e^{j\omega\tau}\left\{\left[1 - (1-\alpha_m)\cos\omega t_m - \alpha_m \cos\omega\left(\frac{1}{I} - t_m\right)\right]\right.$$

$$\left. + j\left[(1-\alpha_m)\sin\omega t_m - \omega_m \sin\omega t_m - \alpha_m \sin\omega\left(\frac{1}{I} - t_m\right)\right]\right\}$$

$$\approx e^{j\omega\tau}\left[\omega^2(1-\alpha_m)\frac{\alpha_m}{r2}\right] \quad \ldots$$

$$P_e = \frac{1}{2\pi}\int_{-\omega_s}^{\omega_s} / S(\omega)\left[e^{j\omega\tau} - (1-\alpha_m)e^{j\omega(\tau - t_m)} - \alpha_m e^{j\omega(\tau - t_m + \frac{1}{I})}\right]/^2\, d\omega \quad \text{error}$$

$$\approx \frac{1}{2\pi}\int_{-\omega_s}^{\omega_s} /S(\omega)e^{j\omega\tau}\left[\omega^2(1-\alpha_m)\,\frac{\alpha_m}{I_2}\right]/^2\, d\omega$$

$$\leq \frac{1}{2\pi}\int_{-\omega_s}^{\omega_s} A^2\left(\frac{0.25}{I^2}\right)^2 \omega^4 d\omega = \frac{A^2(\omega_s)^5}{80\,\pi\, I^4}$$

$$SDR_2 = \frac{P}{P_e} \geq \frac{80I^4}{\omega_s^4}$$

APPLICATION OF MULTIRATE DIGITAL SIGNAL PROCESSING

Multirate Digital signal processing has following applications :

1. Design of Phase Shifters
2. Interfacing of Digital Systems with different sampling rates
3. Implementation of Narrow Band Low Pass filters (NB-LPF)
4. Implementation of Digital Filter Banks.
5. Subband coding of speech signals
6. Quadrature Mirror Filters.
7. Transmultiplexers
8. Oversampling A/D and D/A Conversion.

Example 11.7 Consider a multirate system shown in Fig. Find $y(n)$ as a function of $x(n)$.

Solution From Fig the outputs of the down sampler are

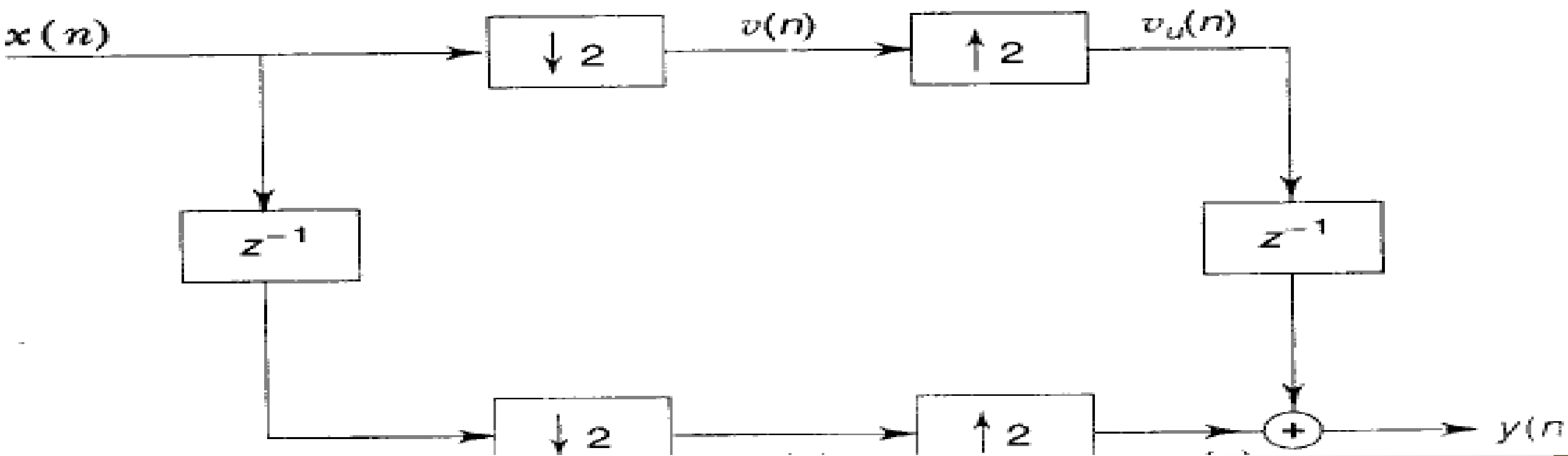
$$V(Z) = \frac{1}{2} X\left(z^{\frac{1}{2}}\right) + \frac{1}{2} X\left(-z^{\frac{1}{2}}\right)$$

$$W(Z) = \frac{z^{\frac{-1}{2}}}{2} X\left(z^{\frac{1}{2}}\right) - \frac{z^{\frac{-1}{2}}}{2} X\left(-z^{\frac{1}{2}}\right)$$

The outputs of the up sampler are

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

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



$Y(z)$ is given by

$$Y(z) = z^{-1} V_u(z) + W_u(z)$$

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

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

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

Example Implement a two-stage decimator for the following specifications:

Sampling rate of the Input signal $s(n)$

$$F_s = 20{,}000 \text{ Hz}$$

Decimation Factor, $D = 100$

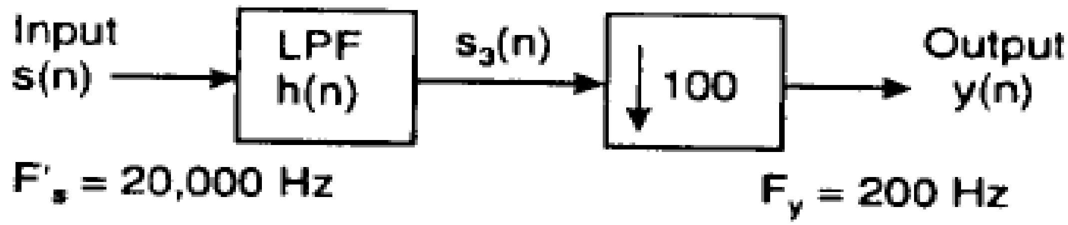
PassBand = 0 to 40 Hz

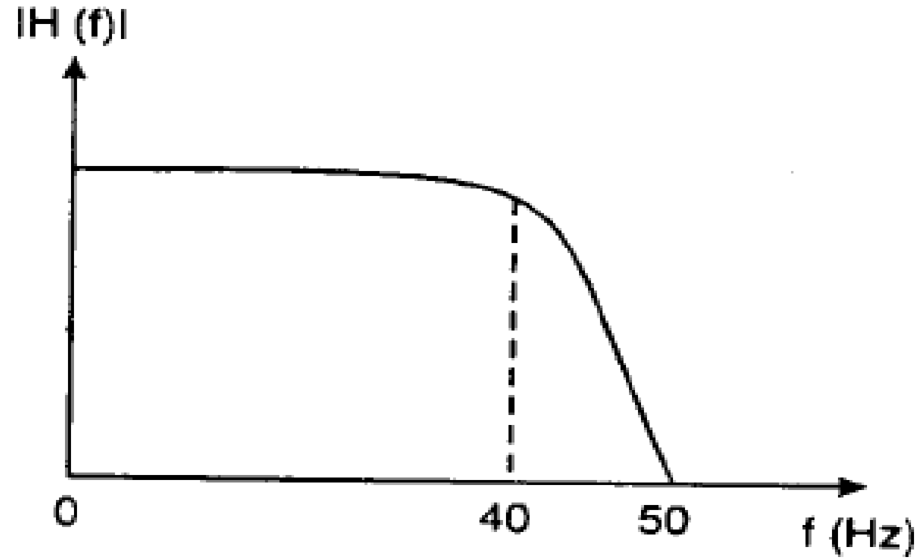
Transition Band = 40 to 50 Hz

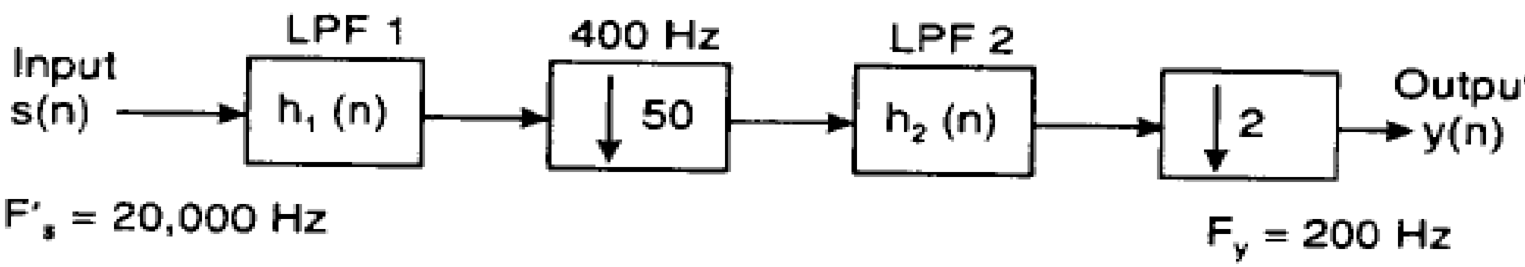
Passband ripple = 0.01

Stopband ripple = 0.002

Solution.


Input
s(n)
LPF
h(n)
$s_3(n)$
↓100
Output
y(n)
F'_s = 20,000 Hz
F_y = 200 Hz


|H (f)|
0
40
50
f (Hz)


Input
s(n)
LPF 1
h_1 (n)
400 Hz
↓50
LPF 2
h_2 (n)
↓2
Output
y(n)
F'_s = 20,000 Hz
F_y = 200 Hz

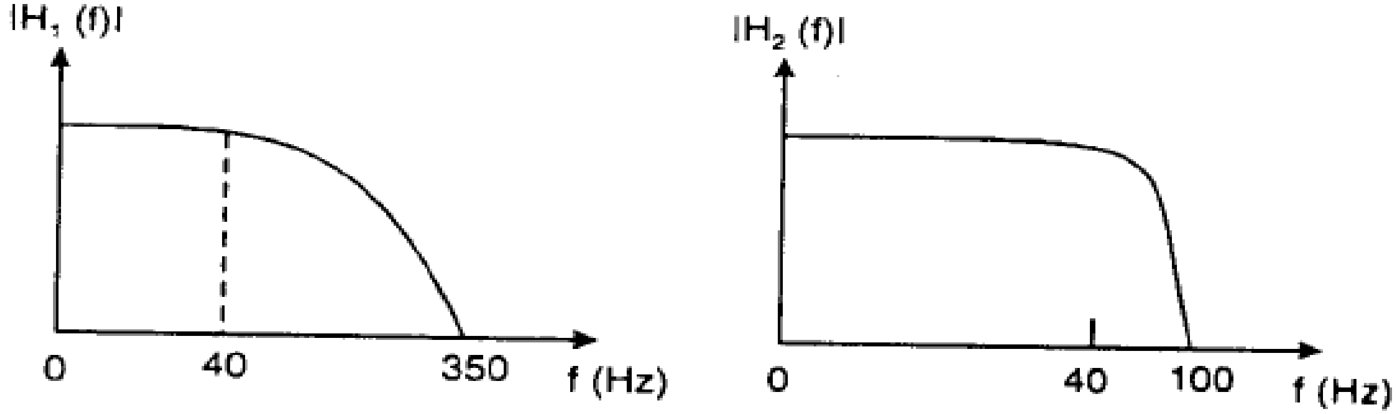


Fig *Illustration of single- stage and two -stage network for decimator,*
The implantation of the system in shown in Fig.

Upper limit of passband $F_p = 40$ Hz
limit of stopband $F_s = 50$ Hz

Pass band ripple $\delta_p = 0.01$

Stopband ripple $\delta_p = 0.002$

D = Decimation factor = 100

Sampling rate of the input signal $s(n)$

$= F_s^1 = 20{,}000$ Hz

$= 20$ kHz

Normalised transition Band-width

$$\Delta f = \frac{F_s - F_p}{F_s'}$$

$$= \frac{50-40}{20{,}000} = \frac{10}{20{,}000} = 5 \times 10^{-4}$$

For an equiripple linear phase *FIR* digital filter, the length N is given by

$$N = \frac{-20\log_{10}\sqrt{\delta_p \delta_s} - 13}{14.6\ \Delta f}$$

$$= \frac{-20\log_{10}\sqrt{(0.01)\ (0.002)} - 13}{14.6\ (5 \times 10^{-4})}$$

$$= 4656$$

In the single stage implementation, the number of multiplication per second is,

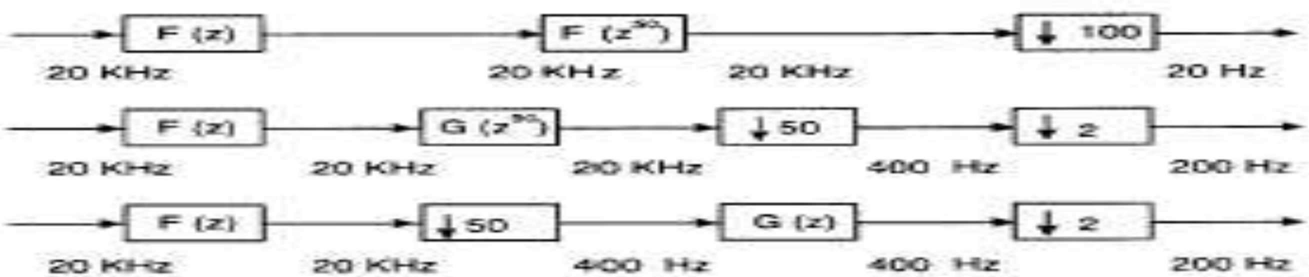
$$N_{M.H} = N\frac{F_s'}{D}$$

$$= \frac{4656 \times 20{,}000}{100}$$

$$= 931200$$

Two-stage Realisation

$H(z)$ can be implemented as a cascade realisation in the form $G(z^{50})\,F(z)$. The steps in the tow-stage realisation of the decimator structure is shown in Fig. and the magnitude response are shown in Fig.



Two stage realisation of the decimators structure.

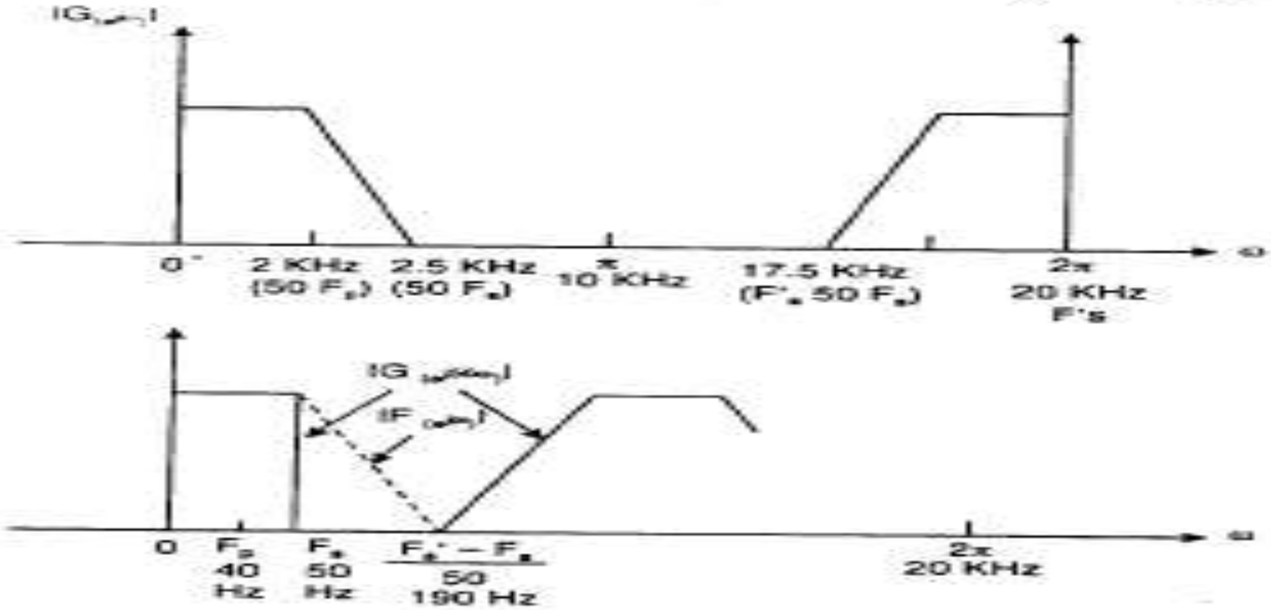


Fig. *Magnitude Response for a two stage decimeter.*

For a cascade realisation, the overall ripple is the sum of the passband ripples of $F(z)$ and $G(z^{50})$. To maintain the stop band ripple atleast as good as $F(z)$ or $G(z^{50})$, δ_s for both can be 0.002. The specification for the interpolated *FIR* Digital filters is given by

For $G(z)$, $\delta_p = 0.005$, $\delta_s = 0.002$

$$\Delta f = \frac{500}{20{,}000} = 2.5 \times 10^{-2}$$

For $F(z)$, $\delta_p = 0.005$, $\delta_s = 0.002$

$$\Delta f = \frac{150}{20{,}000} = 7.5 \times 10^{-3}$$

The filter lengths are calculated as follows :
For $G(z)$,

$$N = \frac{-20 \log_{10} \sqrt{(0.005)\,(0.002)} - 13}{14.6 \left(\frac{2.5 \times 10^3 - 2 \times 10^3}{20 \times 10^3} \right)}$$

$$= 101$$

For $F(z)$,

$$N = \frac{-20\log_{10}\sqrt{(0.005)(0.002)} - 13}{14.6\left(\frac{190-40}{20\times 10^3}\right)}$$

$$= 337$$

The length of the overall filter in cascade is given by

$$337 + (50 + 101) + 2$$

$$= 5389$$

The filter length in cascade realisation has increased but the number of multiplication per second can be reduced.

$$N_{M,G} \frac{101\times 400}{2} = 20{,}200$$

$$N_{M,F} \frac{337\times 20000}{50} = 134800$$

Total number of multiplication per second is given by

$$N_{M,G} + N_{M,F} = 20{,}200 + 1{,}34{,}800$$

$$= 1{,}55{,}000$$

Example Compare the single-stage, two-stage, three-stage and multistage realisation of the decimator with the following specifications.

Sampling rate of a signal has to be reduced from 10 kHz to 500 Hz. The decimation filter H(z) has the passband edge (F_p) to be 150 Hz, stopband edge (F_s) to be 180 Hz.

Passband ripple $\delta_p = 0.002$

Stopband ripple $\delta_s = 0.001$

Solution. The length N of an equiripple linear phase *FIR* digital filter in given by

$$N = \frac{-20\log_{10}\sqrt{\delta_p\delta_s} - 13}{14.6\,\Delta f}$$

where Δf = Normalised transition band-width

$$= \frac{F_s - F_p}{F_s'}$$

Given $F_s' = 10\text{ kHz}$

$$N = \frac{-20\log_{10}\sqrt{(0.002)\,(0.001)} - 13}{14.6\left(\frac{180-150}{10{,}000}\right)}$$

$$\approx 1004$$

For the single-stage implementation of the decimator with a decimation factor of 20, the number of multiplications per second is given by

$$N_{M,H} = \frac{N\ F_s'}{D}$$

$$= \frac{1004 \times 10{,}000}{20} = 50{,}2000$$

Two-stage Realisation

$H(z)$ can be implemented as a cascade realisation in the form of $G(z$ $F(z)$. The steps in the two-stage realisation of the decimator structure are shown in Fig. and response is shown in Fig.

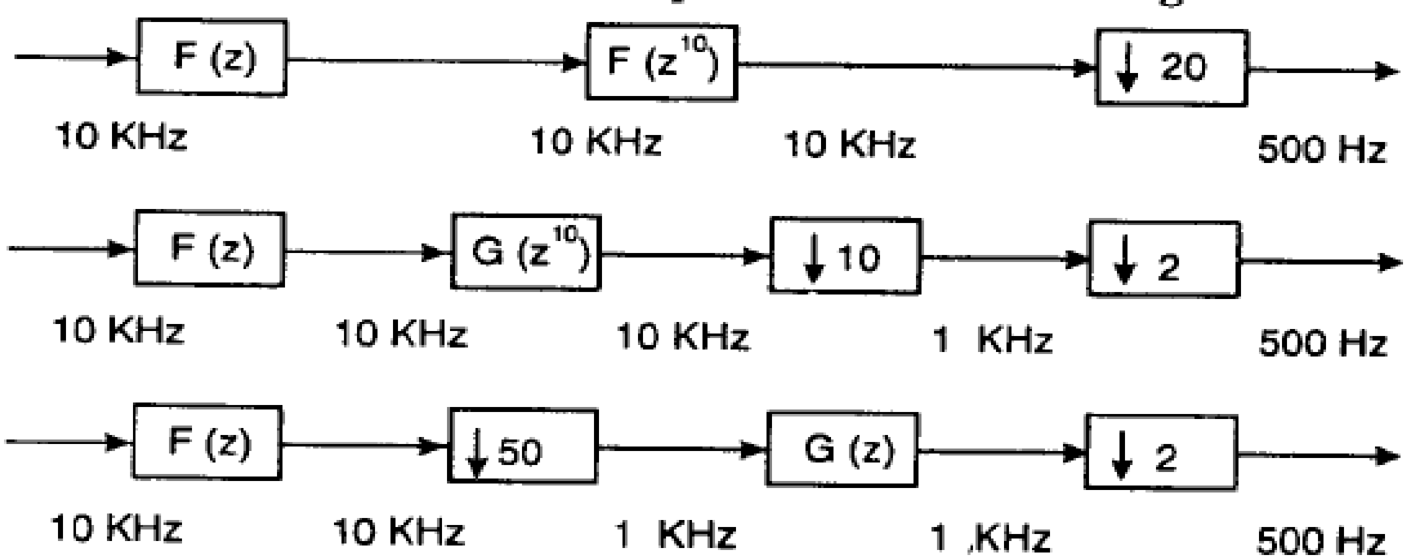


Fig. *(a) Two-stage realisation of the decimators structure.*

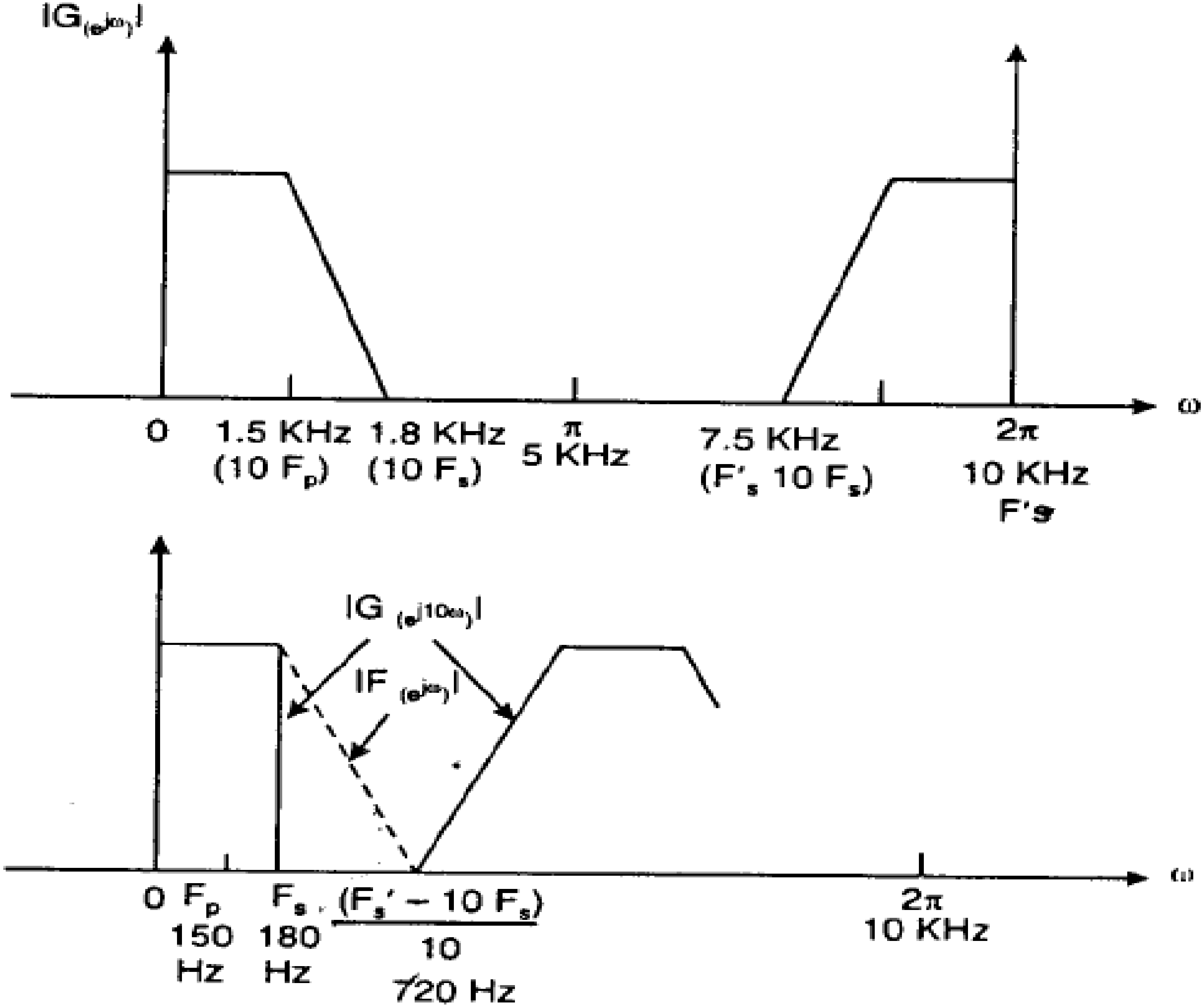


Fig. *(b) Magnitude response for a two stage decimator.*

For the cascade realisation, the overall ripple is the sum of the passband ripples of $F(z)$ and $G(z^{10})$. To maintain the stop band ripple atleast as good as $F(z)$ or $G(z^{10})$, δ_s for both can be 0.001. The specification for the interpolated *FIR* Digital filters is given by

For $G(z)$, $\delta_p = 0.001$

$$\delta_s = 0.001$$

$$\Delta f = \frac{300}{10,000}$$

For $F(z)$, $\delta_p = 0.001$

$$\delta_s = 0.001$$

$$\Delta f = \frac{570}{10,000}$$

The filter length N can be determined as follows :

For $G(z)$,

$$N = \frac{-20\log_{10}\sqrt{\delta_p\delta_s} - 13}{14.6\,(\Delta f)}$$

$$= \frac{-20\log_{10}\sqrt{\delta_p \delta_s} - 13}{14.6\left[\frac{F_s - F_p}{F_s'}\right]}$$

$$= \frac{-20\log_{10}\sqrt{(0.001)\,(0.001)} - 13}{14.6\left(\frac{18\times10^3 - 15\times10^3}{10\times10^3}\right)}$$

$$= 107$$

For $F(z)$,

$$N = \frac{-20\log_{10}\sqrt{(0.001)\,(0.001)} - 13}{14.6\left(\frac{720 - 150}{10\times10^3}\right)}$$

$$= 56$$

The length of the overall filter in cascade is given by

$$56 + (10 \times 107) + 2 = 1128$$

The filter length in cascade realisation has increased but the number of multiplications per second can be reduced.

$$N_{M,G} = 107\,\frac{1000}{2} = 53{,}500$$

$$N_{M,F} = 56 \frac{10,000}{10} = 56,000$$

Total number of multiplication per second is

$$N_{M,G} + N_{M,F}$$

$$= 53,500 + 56,000$$

$$= 1,09,500$$

Three-stage Realisation :

The decimation filter $F(z)$ can be realised in the cascade form $P(z)\,Q(Z^5)$.

The specifications are given as follows :

For $G(z)$, $\delta_p = 0.0005$

$$\delta_s = 0.001$$

$$\delta f = \frac{570}{10,000} \times 5 = 0.285$$

$$N = 12$$

For $P(z)$, $\delta_p = 0.0005$

$$\delta_s = 0.001$$

$$\delta f = \frac{1130}{10,000} = 0.113$$

$$N = 30$$

The three stage realisation is shown in Fig

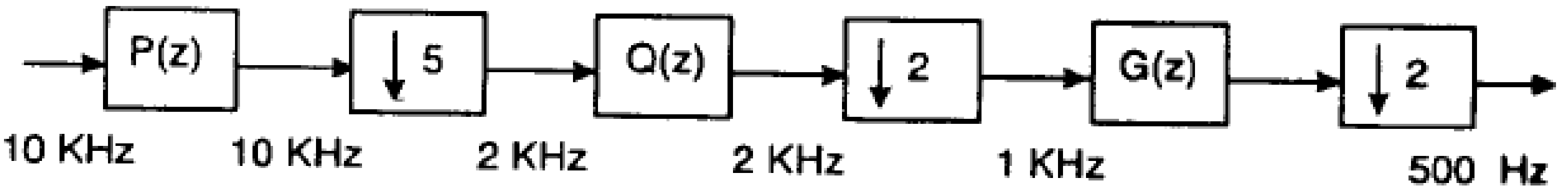


Fig. *Frequency Response for a three-stage Decimation.*

The number of multiplications per second is given by

$$N_{M,Q} = \frac{12 \times 2000}{2} = 12{,}000$$

$$N_{M,P} = \frac{30 \times 1000}{2} = 60{,}000$$

The overall number of multiplications per second for a three stage realisation is given by

$$N_{M,G} + N_{M,Q} + N_{M,P} = 53{,}500 + 12{,}000 + 60{,}000$$

$$= 1{,}25{,}500$$

The number of multiplications per second for a three-stage realisation is more than that of a two-stage realisation. Hence higher than two-stage realisation may not lead to an efficient realisation.

Example We have given a muiti sampling rate system shown in Fig. Find $y(n)$ as a function of $s(n)$.

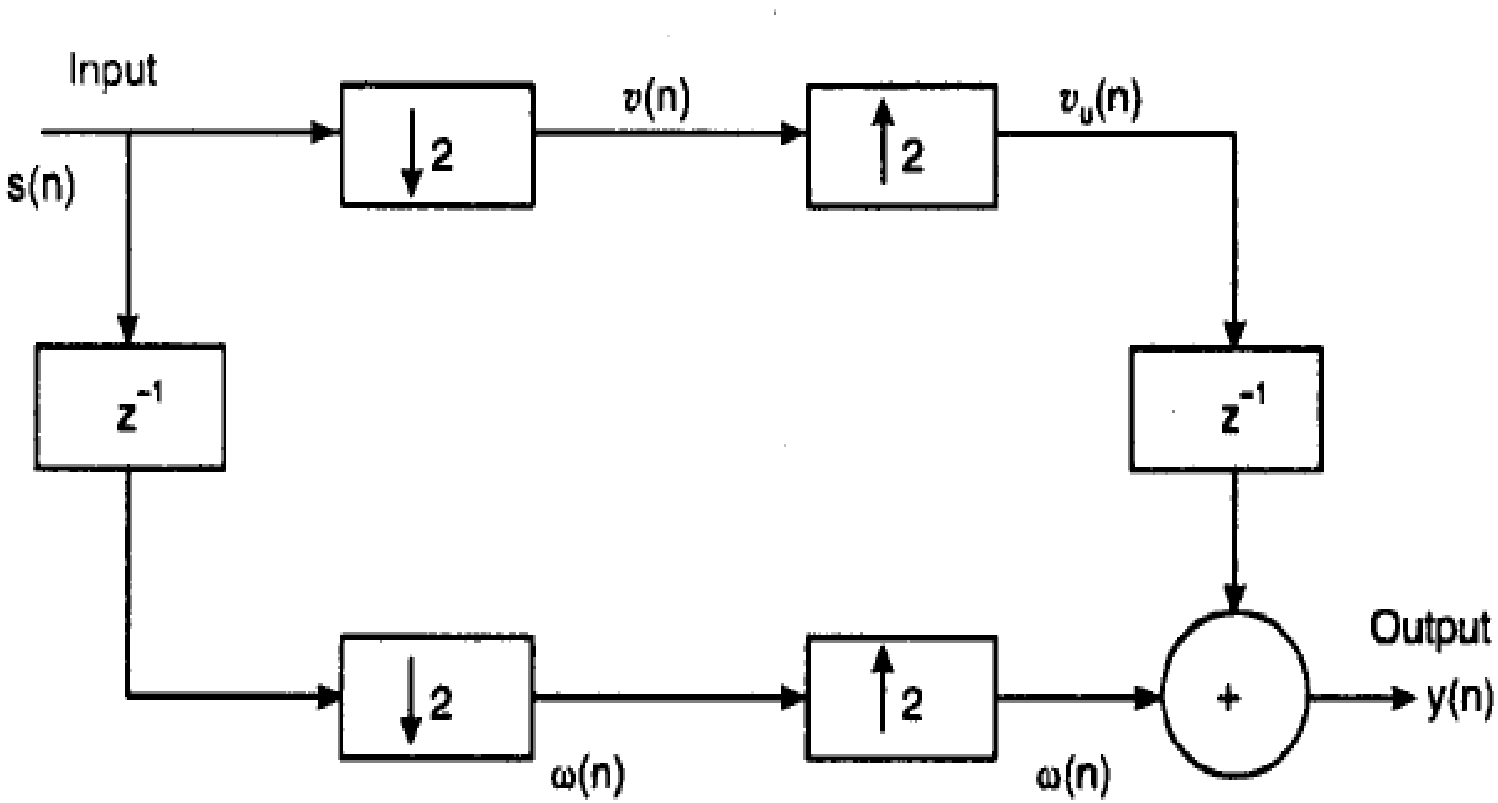


Fig. *Multisampling rate system.*

Solution. **From above Fig.   , the outputs of the down-sampler are given as**

$$V(z) = \frac{1}{2}\, S(z^{\frac{1}{2}}) + \frac{1}{2}\, S(-z^{\frac{1}{2}}) \qquad \ldots(1)$$

$$W(z) = \frac{z^{\frac{-1}{2}}}{2}\, S(z^{\frac{1}{2}}) - \frac{Z^{\frac{-1}{2}}}{2}\, S(-z^{\frac{-1}{2}}) \qquad \ldots(2)$$

The outputs of the up-sampler are

$$V_u(z) = \frac{1}{2}\, S(z) + \frac{1}{2}\, S(-z) \qquad \ldots(3)$$

$$W_u(z) = \frac{Z^{-1}}{2}\, S(z) \frac{-Z^{-1}}{2}\, S(-z) \qquad \ldots(4)$$

$Y(z)$ is given by

$$Y(z) = z^{-1}\, V_u(z) + W_u(z) \qquad \ldots(5)$$

Substituting Eqns. (3) and (4) in Eqn. (5), we get

$$Y(z) = Z^{-1}\left[\frac{1}{2}\, S(z) + \frac{1}{2}\, S(-z)\right] + \left[\frac{z^{-1}}{z}\, S(z) - \frac{z^{-1}}{2}\, S(-z)\right]$$

or $$Y(z) = Z^{-1}\, S(z) \qquad \ldots(6)$$

Taking the inverse z-transform of both sides of Eqn. (6), we get

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

REVIEW QUESTIONS

1. What is Multirate Digital Signal Processing (MDSP) ?
2. What is the need for Multirate Digital signal processing ?
3. Give some examples of Multirate Digital Systems.
4. Write short notes on the following topics :
   (*a*) MDSP (*b*) Decimator (*c*) Decimation filter (*d*) Interpolator
   (*e*) Interpolation filter
5. Explain the interpolation process for an integer factor I with an example.
6. Explain the Decimation process for an integer factor D with an example.
7. The signal $s(n)$ is defined by

$$s(n) = \begin{cases} A^n, & n > 0 \\ 0, & \text{otherwise} \end{cases}$$

   (*a*) Obtain the decimated signal with a factor of 3
   (*b*) Obtain the interpolated signal with a factor of 3.
8. Explain polyphase decomposition process.
9. Describe the sampling rate conversion by a rational factor $\left(\frac{I}{D}\right)$.
10. Obtain the polyphase structure of the filter with the transfer function.

$$H(z) = \frac{1 - 3z^{-1}}{1 + 4z^{-1}}$$

11. Give the name of some areas where *MDSP* systems are used.
12. Give the advantages of using *MDSP* systems.
13. Discuss filter design and implementation for sampling-rate conversion.
14. Describe and draw Direct-form *FIR* Digital filter structure.
15. Write short notes on following :
    (*a*) Polyphase Digital filter structure
    (*b*) Time-varying Digital filter structure
16. Describe the sampling-Rate Conversion by an arbitrary factor.
17. Write short notes on the following :
    (*a*) Sampling rate conversion by use of first-order approximation method.
    (*b*) Sampling rate conversion by use of second-order (Linear) approximation method.
18. List some applications of *MDSP*.

THANK YOU
END OF FOURTH UNIT