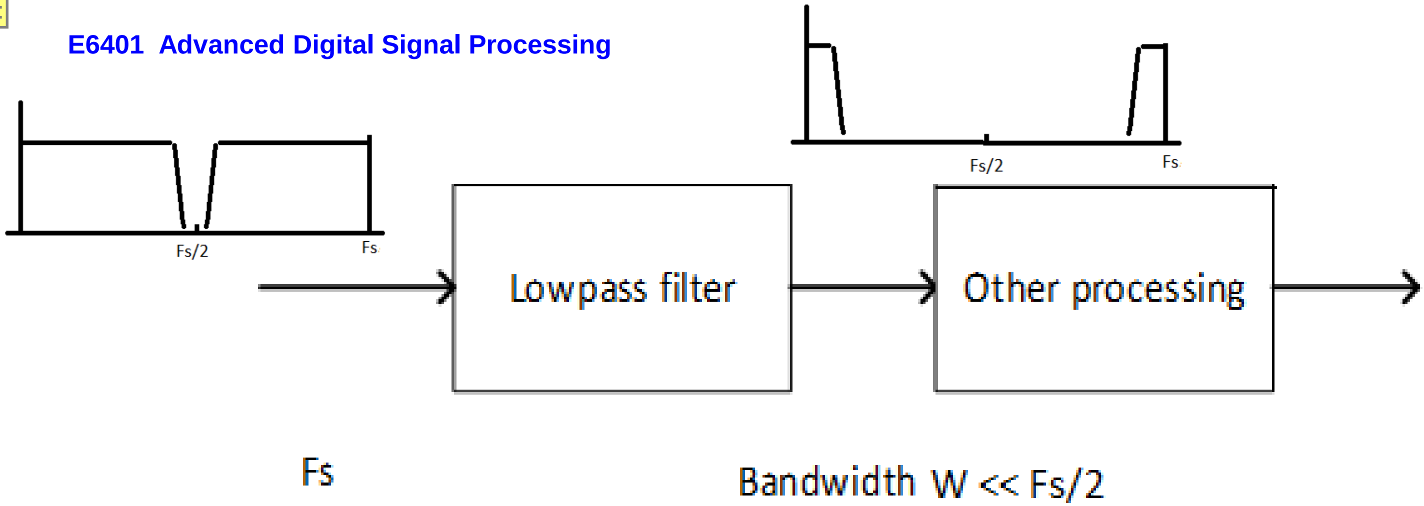
Why and What is Multi-Rate Signal Processing?

Why Multi-Rate Signal Processing?

- One of the most fundamental concepts in DSP is the sampling theory, $F_T \geq 2F_{\max}$, where $F_{\max}$ is the bandwidth or maximum frequency of the signal.
- A suitable sampling rate determines the computation efficiency, and/or accuracy of the processing systems. For a better performance, the sampling rate should be as high as possible and the word length should be as long as possible.
- However, the sampling rate and the word length should be as small as possible whenever and wherever the computational complexity is concerned.
- The input sample rate of many applications has been predetermined and a different sampling rate is usually needed at the outputs of the functional blocks or the system.
- For example, the sampling frequency of 32 kHz is used for broadcasting, 44.1 kHz for digital CD and 48 kHz for digital audio tape, all of them are included in one Hi-Fi system.



- The bandwidth of lowpass filter output is smaller than W.
- It is possible that the other processing system after the filter uses a smaller sampling frequency, i.e., demultiplexing operation in communication system.
- Need to effectively reduce the overall computational complexity by using a suitable sampling frequency for other processing.

- Can we have a digital filter to produce the output with a sampling frequency different from its input sampling frequency?

Why Multi-Rate Signal Processing?

- In the same system, multiple sampling rates are frequently used for minimizing the computational complexity to achieve acceptable performance.
- The process of converting a signal from a given sampling rate to another is called ***sampling rate conversion*** (***SRC***); Systems using multiple sampling rates are called ***multi-rate*** DSP systems;
- Sampling rate conversion can also be realized by passing the signal through a D/A converter, filtering it if necessary, and re-sampling the resulting signal at the desired rate.
- Because the sampling rate conversion is often required in a complex digital signal processing system, the conversion is better performed in the digital domain for better performance and less costs of processing.

Basic Requirements of Sampling Rate Conversion

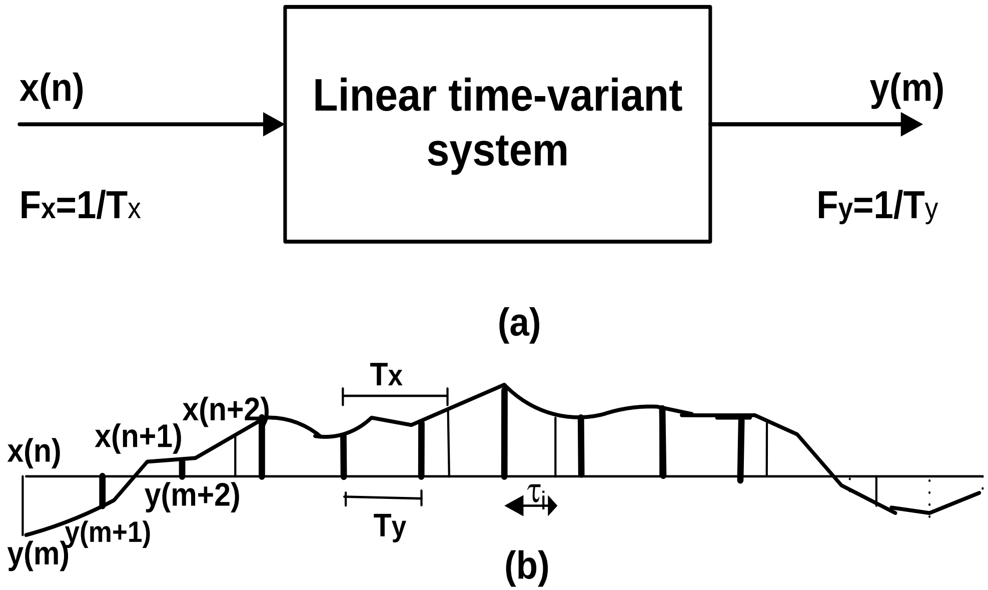


Figure 7 Sample rate conversion viewed as a linear time variant system.

To generate a digital signal with a sampling rate F_y from a given digital signal with a sampling rate F_x,

- We need a ***linear time variant*** system;
- The output $y(m)$ is an estimated version from $x(n)$;
- Note that $y(m)$ and $x(n)$ uses different time indices m and n.

Basic Requirements of Sampling Rate Conversion

- The system provides a flat amplitude response and a time delay t_i for the ith samples of $x(n)$. In particular, t_i is time variant for different i which means that the filter is **time-variant**;
- In general, $y(n)$ should provide the same information as $x(n)$, such that they both are able to reconstruct the same original signal.
- We shall study the **requirements** and **designs** of the system $h(n,m)$ where n and m are the input and output indices in time.
- Multi-rate processing results in several advantages including:
  - less computational requirements;
  - less storage for filter coefficients and signal histories;
  - lower order filter design/implementations;
  - less finite arithmetic effects.
- These advantages must be weighted against the additional complexity in the overhead of the hardware and software implementation to handle the changes in the sampling rates.

Changing Sampling Frequency by a Factor of Integer

Interpolation by a Factor *I*

- Up-sampling process increases the sampling rate by an integer factor of *I*. In the time domain, this is accomplished by inserting *I*-1 *zero valued* samples between each pair of adjacent samples.

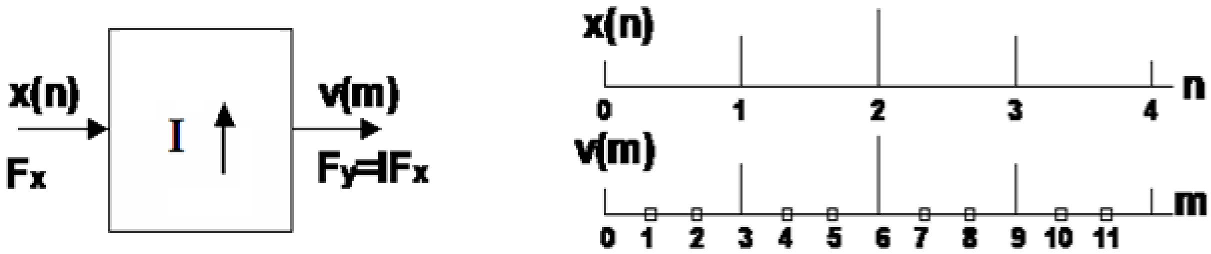


Figure 2.1 Operation of up-sampling (I=3)

- F_x and F_y are the sampling frequencies of $x(n)$ and $v(m)$.
- Let $v(m)$ denote a sequence with a rate $F_y = IF_x$ being obtained from $x(n)$ by adding I-1 zeros between successive values of $x(n)$. Then

$$v(m) = \begin{cases} x(m/I), & m = 0, \pm I, \pm 2I, \ldots \\ 0, & \text{otherwise} \end{cases} \qquad (2.1)$$

which has a z-transform

$$V(z)=\sum_{m=-\infty}^{\infty} v(m)\, z^{-m} = \sum_{m=-\infty}^{\infty} x(m)\, z^{-mI} = X(z^{I}) \tag{2.2}$$

- Let $z=e^{-j\omega y}$ and $z^{I}=e^{-j\omega yI}$ for its spectrum in (2.2) on the unit circle, we have

$$V(\omega_y)=X(\omega_y I) \tag{2.3}$$

where $\omega_y=2\pi F/F_y$ and $F_y=IF_x$ or $\omega_y=2\pi F/(IF_x)=\omega_x/I$, the spectrum of $x(n)$ and $v(m)$ are shown below:

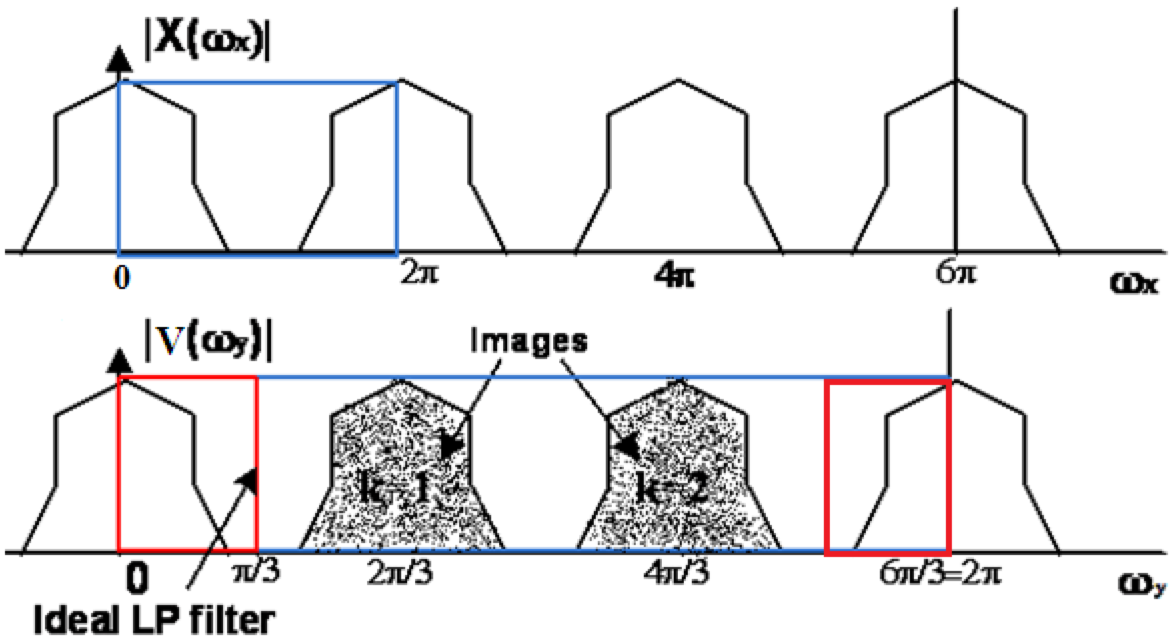


Figure 2.2 Spectra obtained using upsampling (I=3)

- The images of $X(\omega_x)$ in $V(\omega_y)$ above $\omega_y=\pi/I$ should be rejected by passing the up-sampled sequence through a low pass filter $h(m)$ (See Figure 2.3)

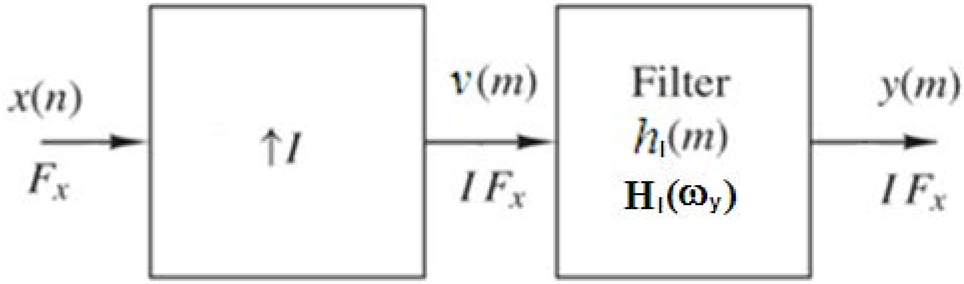


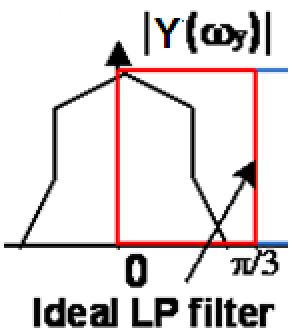
Figure 2.3 Interpolator by a factor of *I*

with frequency response $H_I(\omega_y)$ whose ideal characteristic is

$$H_I(\omega_y)=\begin{cases} C, & 0\le|\omega_y|\le\pi/I \\ 0, & \text{otherwise} \end{cases} \tag{2.4}$$

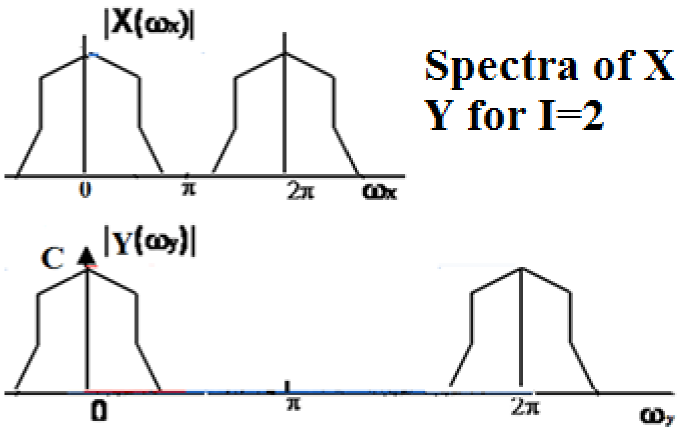
- The output spectrum is

$$Y(\omega_y)=\begin{cases} CX(\omega_y I), & 0\le|\omega_y|\le\pi/I \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$



- Both amplitude and the frequency are scaled in (2.5).

where C is a nonzero scaling constant. It can be proven that when $C = I$, we have $y(m) = x(m/I)$ for $m = 0, \pm I, \pm 2I, \ldots$.



Spectra of X and Y for I=2

- The output sequence $y(m)$ can also be expressed as a convolution of $v(m)$ with the impulse response $h(m)$ of the lowpass filter. For example,

$$y(m) = \sum_{k=-\infty}^{\infty} h(m-k)v(k)$$

since $v(k)=0$ except at multiples of I, i.e., $v(kI) = x(k)$, then

$$y(m) = \sum_{k=-\infty}^{\infty} h(m-kI)x(k) \tag{3.6}$$

which is the mathematical expression of time domain operation for interpolation.

Decimation by a Factor *D*

- Assuming that signal $x(n)$ with spectrum $X(\omega)$ is to be down sampled by an integer factor D. The **down-sampling** process is seen in figure 3.1.

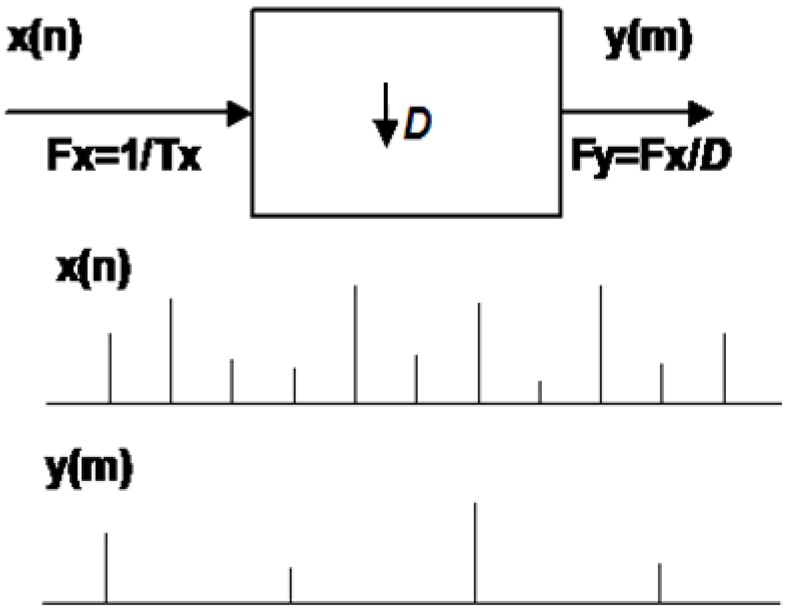


Figure 3.1 (a) Down sampling by a factor D=3

- In the time domain, the down sampler outputs one sample out of every D samples, therefore,

$$y(m) = x(mD);$$

Decimation by a Factor *D*

- In the frequency domain, if $x(n)$ is band limited to π/D. Its spectrum is periodic in ω_x, the period being the normalized sampling rate 2π (Figure 3.1 (b) (*i*));
- The magnitude is reduced by a factor of D;
- If the signal bandwidth is more than π/D, ***overlapping*** between the spectrum of original signal and the replica occurs (Figure 3.1 (b) (*ii*));
- To prevent distortion from spectrum overlapping, anti-aliasing filtering is necessary before decimation (see Figure 3.2).

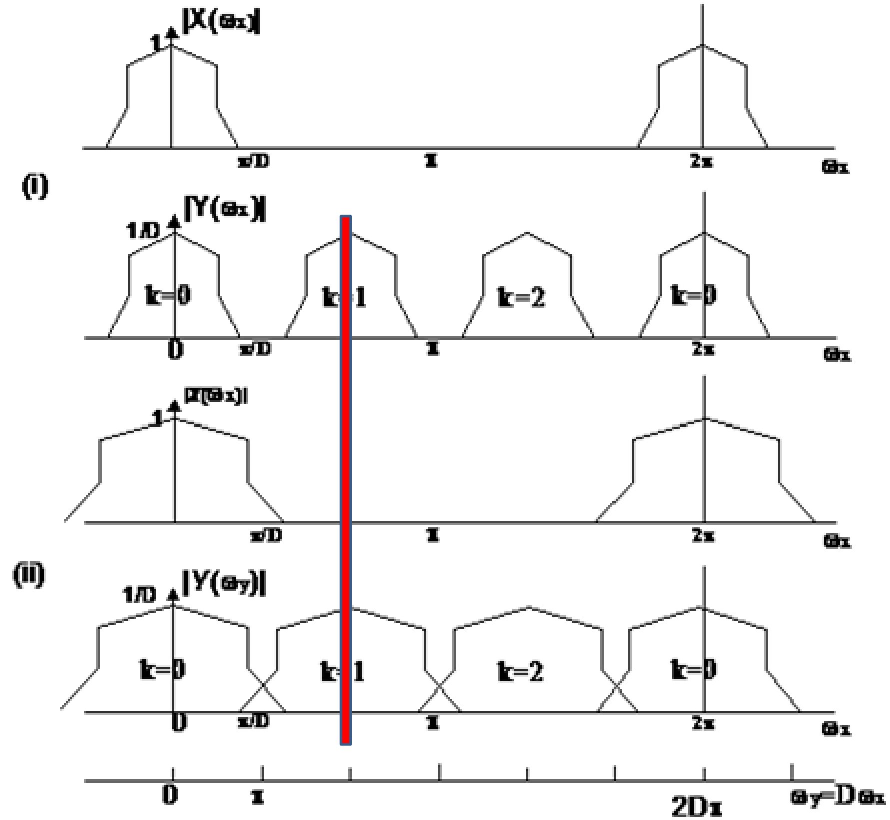


Fig 3.1(b)

Decimation by a Factor *D*

- To prevent spectrum overlapping, anti-aliasing filtering is necessary before decimation (see Figure 3.2).

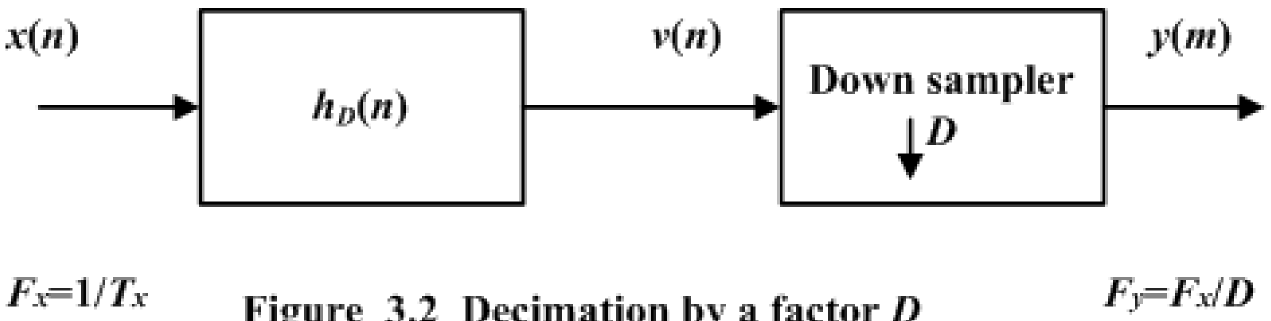
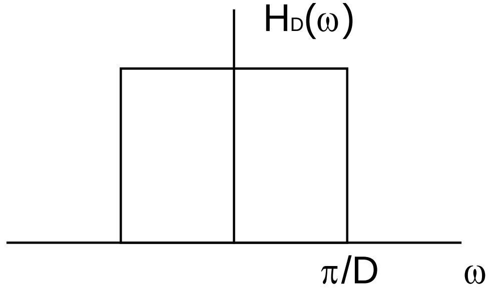


Figure 3.2 Decimation by a factor *D*

- In this case, the output $y(m)$ is distorted by losing high frequency component of the input.
- The frequencies in the low range have been preserved, which generally contain most information of the signal.

Decimation by a Factor *D*

- The input sequence $x(n)$ is passed through a lowpass filter characterized by the impulse response $h_D(n)$ with a frequency response $H_D(\omega)$, which ideally satisfies the condition



$$H_D(\omega) = \begin{cases} 1, & |\omega| \le \pi / D \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

which eliminates the spectrum of $X(\omega)$ in the range of $\pi/D < \omega < \pi$.

- The output of the filter is a sequence $v(n)$ (see Figure 3.2) given as

$$v(n) = \sum_{k=0}^{\infty} h_D(k) x(n-k) \tag{3.2}$$

which is then down sampled by a factor of D to obtain $y(m)$. Thus

$$y(m) = v(mD) = \sum_{k=0}^{\infty} h_D(k)\, x(mD-k) \tag{3.3}$$

Decimation by a Factor *D*

- The combination of filtering and down-sampling is *time variant* because given the fact that $x(n)$ produces $y(m)$, then $x(n-n_0)$ is generally not equal to $y(n-n_0)$.

The frequency domain characteristics

- The mathematical expression for the spectrum of the decimated signals is

$$Y(\omega_y) = \frac{1}{D}\sum_{k=0}^{D-1} H_D\left(\frac{\omega_y - 2\pi k}{D}\right) X\left(\frac{\omega_y - 2\pi k}{D}\right) \qquad (3.4)$$

- With a properly designed filter $H_D(\omega)$ the aliasing is eliminated.

- The spectrum ($k = 0$ in (3.4)) is

$$Y(\omega_y) = \frac{1}{D} H_D\left(\frac{\omega_y}{D}\right) X\left(\frac{\omega_y}{D}\right) = \frac{1}{D} X\left(\frac{\omega_y}{D}\right) \qquad 0 \le |\omega_y| \le \pi \qquad (3.5)$$

Decimation by a Factor *D*

Example:

Observations on Figure 3.3:

- Filter $H_4(\omega)$ avoids overlapping between duplicated spectrum;
- Sample frequency changes from ω_x to ω_y=4ω_x;
- Amplitude of $Y(\omega_y)$ is reduced by a factor of 4.

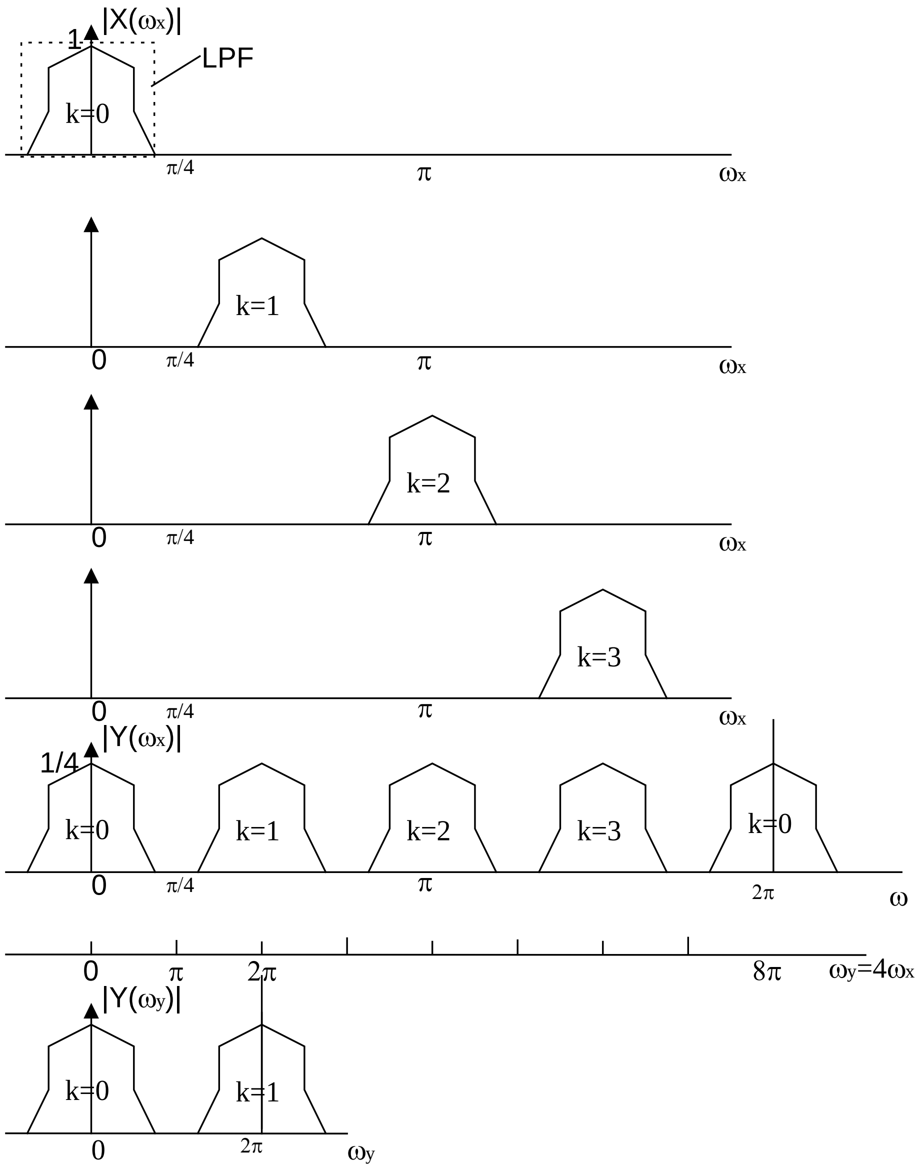
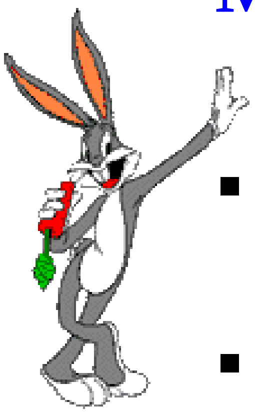


Figure 3.3 Example Illustration of spectra of decimation process (D=4)

Main Points:



- What are the operations in time domain for decimation?
- What are the signal spectrum before and after interpolation/decimation?
- Why anti-imaging/anti-aliasing filter is necessary?
- Be able to correctly plot the spectrum achieved before and after interpolation/decimation

Changing the Sampling Frequency by a Factor of Ratio

A more general case

Sampling Rate Conversion by a Ratio *I*/*D*

- Such a sampling rate conversion is achieved by first performing interpolation by the factor *I* and then decimating the output of the interpolator by factor *D*. The cascaded structure is given below.

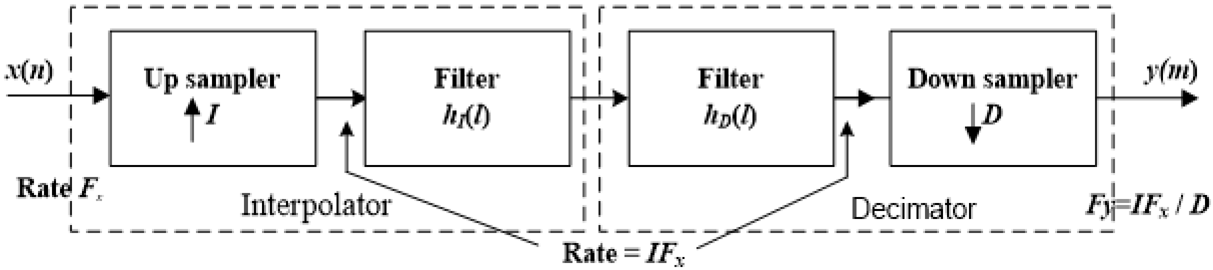


Figure 4.1 Method for sampling rate conversion by a factor *I*/*D*.

- The interpolation **must** be performed **first** and then the decimation to preserve the desired spectral characteristics of $x(n)$
- Both filters h_D and h_I are operated at the same sampling rate $(=IF_x)$ and can be combined into a single low pass filter (see Fig 4.2).
- The filters are operated at the place that has the **highest** sampling rate in the system.

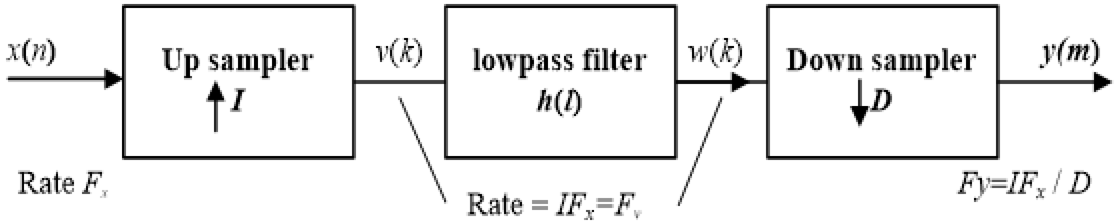


Figure 4.2 Sampling rate conversion by a factor *I/D*.

- The frequency response of the combined filter must incorporate the filtering operations for both interpolation and decimation, and hence it should ideally be the frequency response characteristic

$$H(\omega_v)=\begin{cases} I, & 0\le|\omega_v|\le\min(\pi/D,\pi/I) \\ 0 & \text{Otherwise} \end{cases} \tag{4.1}$$

where $\omega_v = 2\pi F/F_v = 2\pi F/IF_x = \omega_x/I$. In the time domain, the output of the up-sampler is the sequence

$$v(l)=\begin{cases} x(l/I), & l=0,\pm I,\pm 2I\ldots \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

and the output of the linear, time-invariant filter is

$$w(l)=\sum_{k=-\infty}^{\infty}h(l-k)v(k)=\sum_{k=-\infty}^{\infty}h(l-kI)x(k) \tag{4.3}$$

Sampling Rate Conversion by a Ratio *I*/*D*

- Finally the output of the sampling rate converter $y(m)$ is obtained by down sampling the sequence $w(l)$ by a factor of D. Thus

$$y(m) = w(mD) = \sum_{k=-\infty}^{\infty} h(mD - kI)x(k) \qquad (4.4)$$

- Let $k = \left\lfloor \frac{mD}{I} \right\rfloor - n$, and after some manipulation, we have

$$y(m) = \sum_{n=-\infty}^{\infty} h(nI + (mD)_I)\, x(\left\lfloor \frac{mD}{I} \right\rfloor - n), \qquad -\infty < m, n < \infty \qquad (4.6)$$

where $\lfloor r \rfloor$ denotes the largest integer contained in r, (e.g. $\lfloor 6.7 \rfloor = 6$), and $(x)_I$ is the operation of x modulo I, (e.g., $(33)_6 = 3$).

- Equation (4.6) indicates that the output $y(m)$ is obtained by passing the input sequence $x(n)$ through a *time-variant* filter with impulse response

$$g(n, m) = h(nI + (mD)_I), \qquad -\infty < m, n < \infty \qquad (4.7)$$

The filter $h(k)$ is the impulse response of the time-invariant lowpass filter operated at the sampling rate If_x.

Sampling Rate Conversion by a Ratio *I*/*D*

For any integer k,

$$g(n, m+kI) = h[nI+(mD+kDI)_I]) = h[nI+(mD)_I]= g(n, m) \qquad (4.8)$$

which indicates that $g(n, m)$ is **periodic** in terms of the variable m with a period I.

- Then, the time variant filter $g(n, m)$ has I different sets of coefficients, each set is a *time invariant* filter.

Example:

We assume the low pass filter $h(n)$ has 30 coefficients, i.e. n = 0 to 29, and I =5 and D=2. According to (4.7), we have

$$g(n, m) = h(5n+|2m|_5)$$

$g(n,m)$	n=0	1	2	3	4	5
m=0	$h(0)$	$h(5)$	$h(10)$	$h(15)$	$h(20)$	$h(25)$
1	$h(2)$	$h(7)$	$h(12)$	$h(17)$	$h(22)$	$h(27)$
2	$h(4)$	$h(9)$	$h(14)$	$h(19)$	$h(24)$	$h(29)$
3	$h(1)$	$h(6)$	$h(11)$	$h(16)$	$h(21)$	$h(26)$
4	$h(3)$	$h(8)$	$h(13)$	$h(18)$	$h(23)$	$h(28)$

Sampling Rate Conversion by a Ratio *I*/*D*

- When (m mod 5)=0, the first set of coefficients, (the second row in the table) is used in the filtering process. The third row is used when (m mod 5) =1 and so on.

- The spectrum at the filter output with the impulse response $h(l)$ is

$$Y(\omega_y)=\begin{cases} \frac{I}{D}X(\frac{I}{D}\omega_y) & 0\le|\omega_y|\le\min(\pi,\pi D/I) \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

where $\omega_y=D\omega_v$, (see text book p.719) for detailed derivation).

- It is seen that the output spectrum $Y(\omega_y)$ is obtained by scaling the amplitude and ω_x of $X(\omega_x)$.

- Demo Example 

Example 1: see the Figure in next page for the process in the frequency domain.

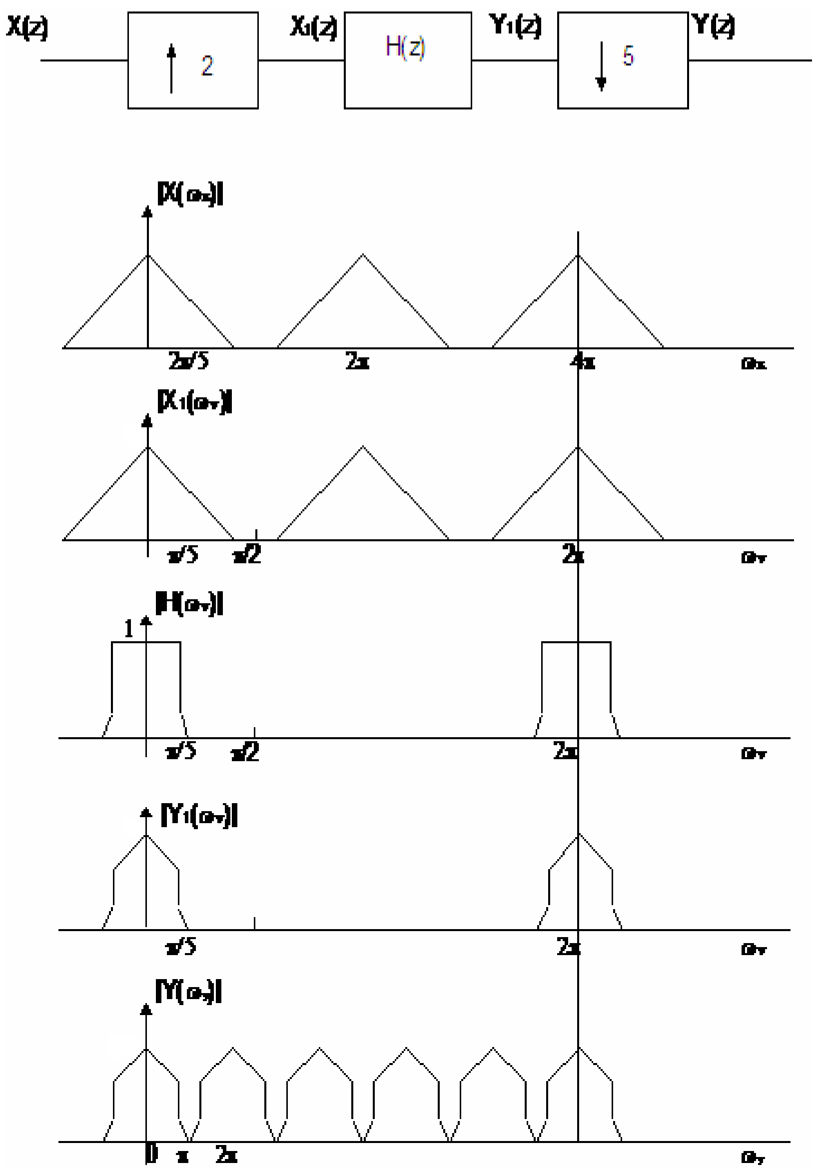


Figure 4.3 A sampling-rate conversion for I = 2 and D = 5.

The change by a constant scaling factor is ignored.

- ← The input spectrum in terms of input sampling rate
- ← Spectrum after upsampling in terms of the increased sampling rate
- ← Spectrum of the filter
- ← Filtered spectrum. Note that the spectrum of input is changed.
- ← The output spectrum in terms of the new sampling rate

Summary

- Three different cases: decimation, interpolation by integer factors and sampling rate conversion by a ratio of integers.
- In the time domain, decimation is achieved by filtering the input before down-sampling; however, interpolation is obtained by up-sampling the input first and then filtering.
- In the frequency domain, the filter used for decimation reduces the bandwidth of the input before down-sampling so that aliasing will not occur; however, the filter used for interpolation is to remove the images produced by the up-sampling process.
- Therefore, the filter specifications are given in terms of input sampling frequency for decimation and of the output sampling frequency for interpolation.
- For conversion by a factor of I/D, the filter specification is in terms of the interpolated frequency, i.e. IF_x and the cutoff frequency is determined by the minimum of the two cutoff frequency requirements for the decimation and interpolation.

How to implement the sampling conversion system?

Basic Structures

Example:

If the input sampling frequency F_T=20 kHz, the decimation filter used in an A/D converter has 151 coefficients, calculate the number of multiplications needed by the decimation process for real time operation.

Assume FIR filter that has a symmetric property, i.e., $h(N-i) = h(i)$, so that each input sample needs about 75 multiplications. The total required number of multiplications is

$$75 \times 20 \times 1{,}000 \times 64 = 9.6 \times 10^7$$

where 64 is the over-sampling factor used in the converter.

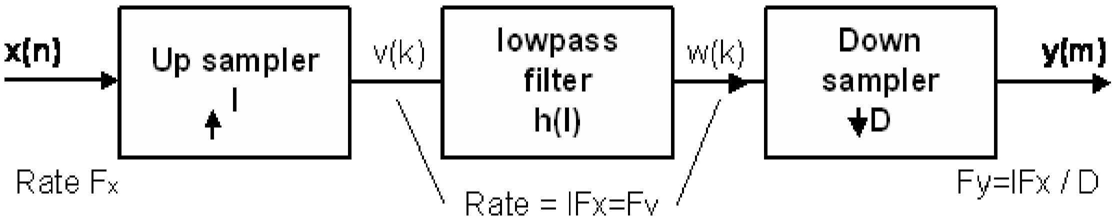
If we assume the DSP chip has a computation cycle of 20 *ns*, or supports 4×10^7 multiplications (assuming 80% utilization).

Therefore, a faster DSP chip is needed. The solution to the above problem is to find *more efficient filter structures* that need less computational complexity.

A DSP designer has always to minimize the cost for the acceptable performances

Filter Design and Implementation

- The conversion by a factor *I/D* shown in the figure below is achieved by



- Increasing the sampling rate by a factor *I*, which is accomplished by inserting *I* - 1 zeros between successive values of input signal *x*(*n*);
- The output sequence is filtered to eliminate the unwanted images;
- Down sampling the filter output by a factor *D*.
- However, such an arrangement is extremely inefficient because:
  - Only one out of *I* input samples is non-zero;
  - Only one out of *D* filtered output samples is used as a valid output;
  - The filter is operated at the highest possible sampling frequency in the system.
- Efficient implementation must be found to reduce the total computational complexity.

Polyphase Filter Structures

- The polyphase structure for an impulse response $h(n)$ is derived from any system function

$$H(z)=\sum h(\mathrm{i})z^{-i}=[\ldots+h(0)+h(M)z^{-M}+\ldots]$$
$$+z^{-1}[\ldots+h(1)+h(M+1)z^{-M}+\ldots]$$
$$+z^{-(M-1)}[\ldots+h(M-1)+h(2M-1)z^{-M}+\ldots]$$

or $$H(z)=\sum_{i=0}^{M-1} z^{-i}P_i(z^M)$$ where $$P_i(z)=\sum_{n=-\infty}^{\infty} h(nM+i)z^{-n}$$

- The polyphase component $p_i(n) = h(nM+i)$, $i= 0, 1, \ldots, M\text{-}1$ is obtained by down sampling a delayed version of the original impulse response $h(n)$.

Polyphase Filter Structures

- For M= 3, the z-transform of the M-components polyphase filter structure is expressed as

$$
\begin{aligned}
Y(z) &= H(z)X(z) \\
&= P_0(z^3)X(z) + z^{-1}P_1(z^3)X(z) + z^{-2}P_2(z^3)X(z) \qquad (11.5.4) \\
&= P_0(z^3)X(z) + z^{-1}\{P_1(z^3)X(z) + z^{-1}[P_2(z^3)X(z)]\} \qquad (11.5.5)
\end{aligned}
$$

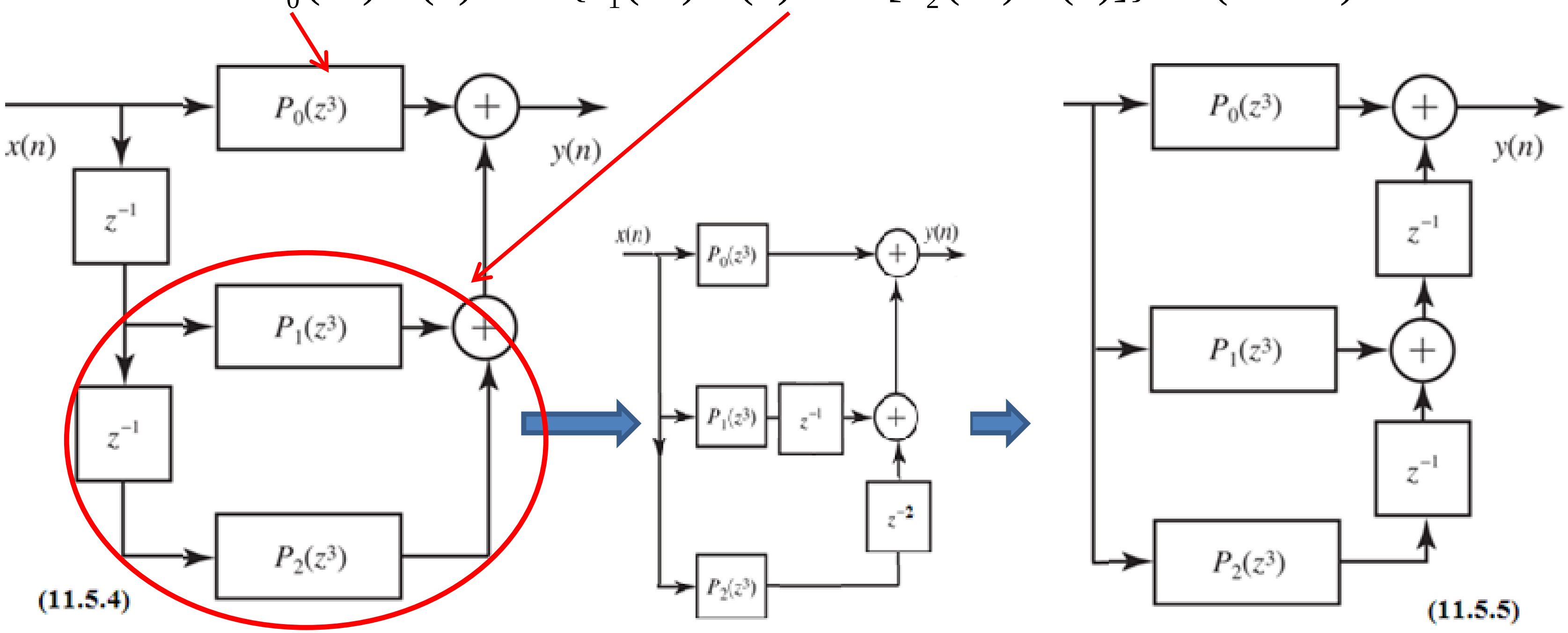
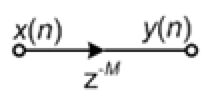
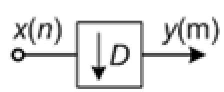
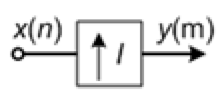
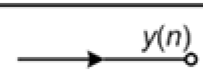


Illustration of transpose polyphase filter structure for $M = 3$.

Signal Flow Graph

- A signal flow graph may be defined in terms of a set of branches and nodes.
- **Branches** define the signal operations in the structure such as delays, gains, sampling rate conversions;
- **Nodes** define the connection points of branches in the structure.
- Typical branch operations are shown in the table on the left.
- *I* and *D* represent the interpolation and decimation factors, respectively.

Operation	Symbol	Time-Domain Description	Frequency Domain Description
M Sample Delay		$y(n)=x(n-M)$	$Y(\omega)=e^{-jM\omega}X(\omega)$
Down Sampling		$y(m)=x(Dm)$	$Y(\omega)=\frac{1}{D}\sum_{l=0}^{D-1}X(\frac{\omega-2\pi l}{D})$
Up Sampling		$y(m)=\begin{cases} x(m/I) & m=0,\pm I,\pm 2I,\ldots \\ 0 & \text{otherwise} \end{cases}$	$Y(\omega)=X(I\omega)$
Input		——	——
Output		——	——

More on signal flow graphs

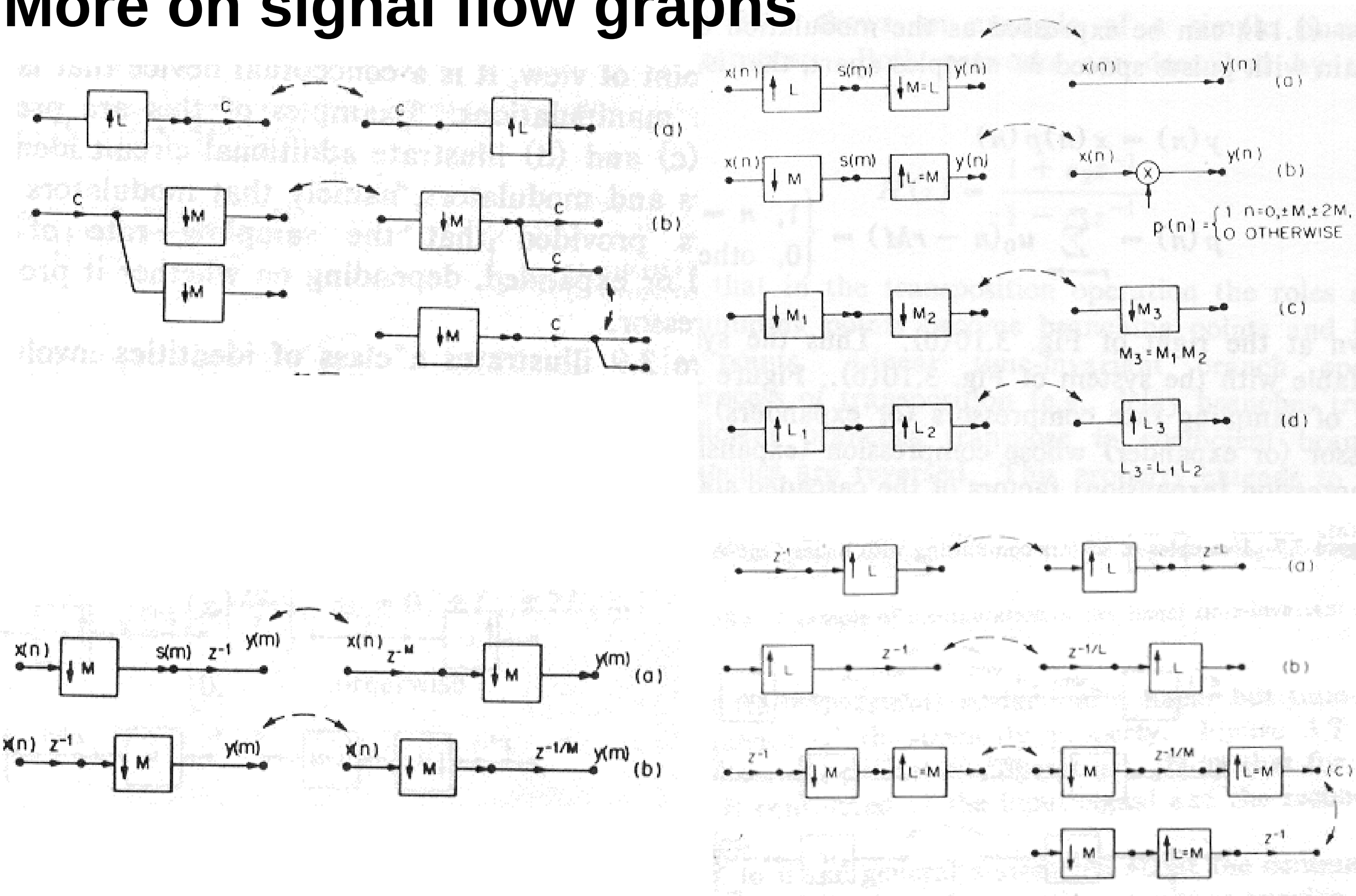


Figure 5. 12 Typical operations for multirate systems

Noble Identity

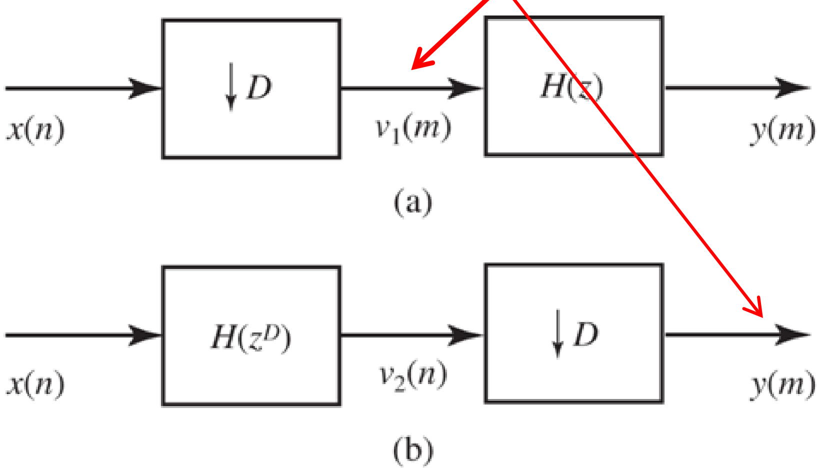
- Proof of the equivalence in the left figure
- It is known the expressions for a down sampling is

$$y(m) = x(nD) \Leftrightarrow Y(z) = \frac{1}{D}\sum_{i=0}^{D-1} X(z^{1/D}W_D^i) \qquad W_D = e^{-j2\pi/D}$$

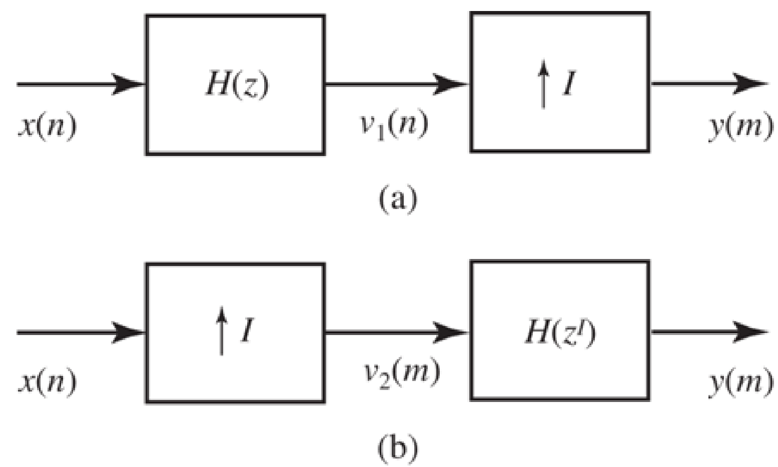
- Figure (b) can be written into

$$Y(z) = \frac{1}{D}\sum_{i=0}^{D-1} V_2(z^{1/D}W_D^i) = \frac{1}{D}\sum_{i=0}^{D-1} H(zW_D^{iD})X(z^{1/D}W_D^i)$$

$$= H(z)\boxed{\frac{1}{D}\sum_{i=0}^{D-1} X(z^{1/D}W_D^i)} = H(z)V_1(z)$$



Two equivalent downsampling systems (first noble identity)



Two equivalent upsampling systems (second noble identity).

- We also observe that the transpose of a decimator is an interpolator, and vice versa, as shown in Figure 5.4;
- We say a decimator is a dual of an interpolator, and vice versa.
- Therefore, there is an interpolator whose structure is the dual of the decimator.
- According to the rules of transformation, it is achieved by changing input into output and D into I in Figure 5.4.

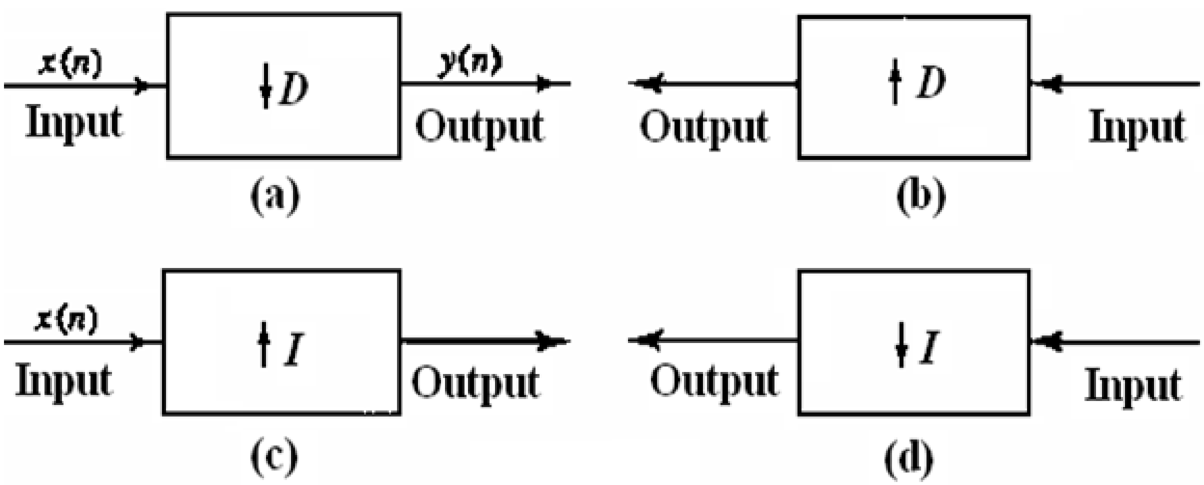
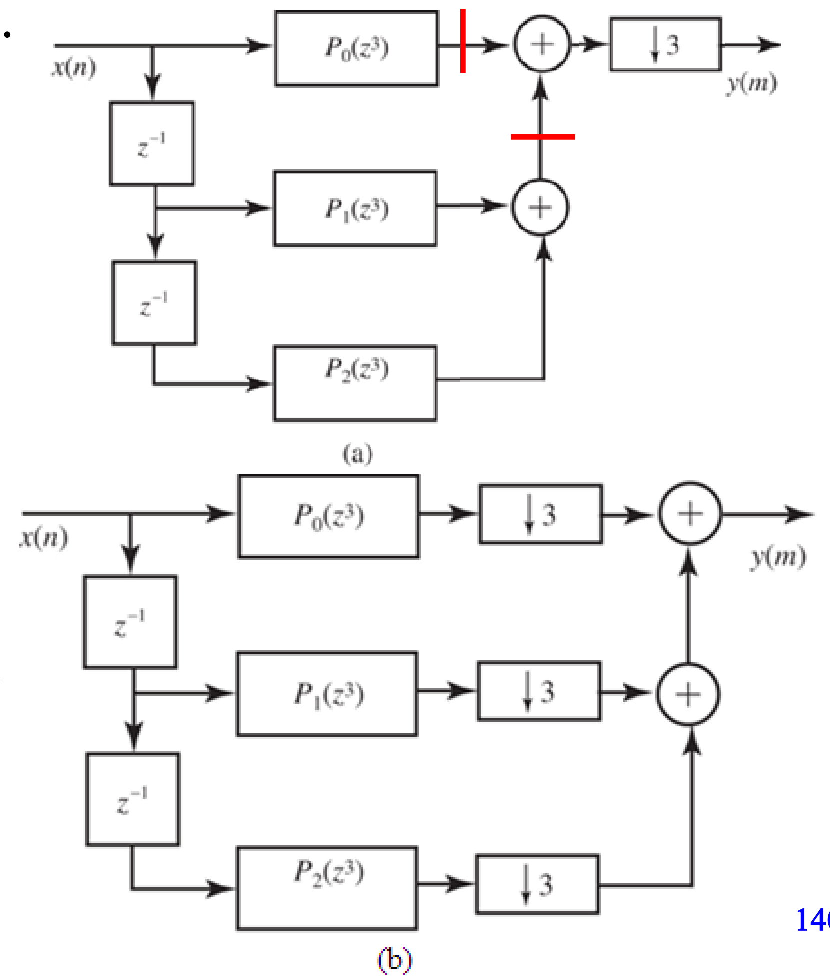
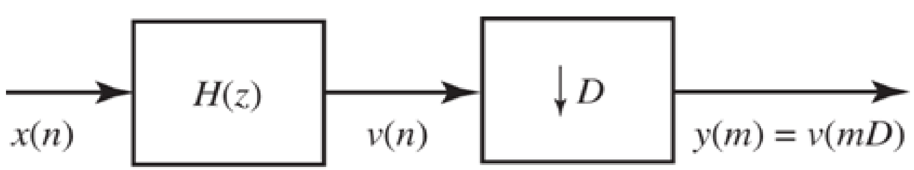


Figure 5.4 Duality of relations obtained through transposition

Polyphase Structures for Decimation

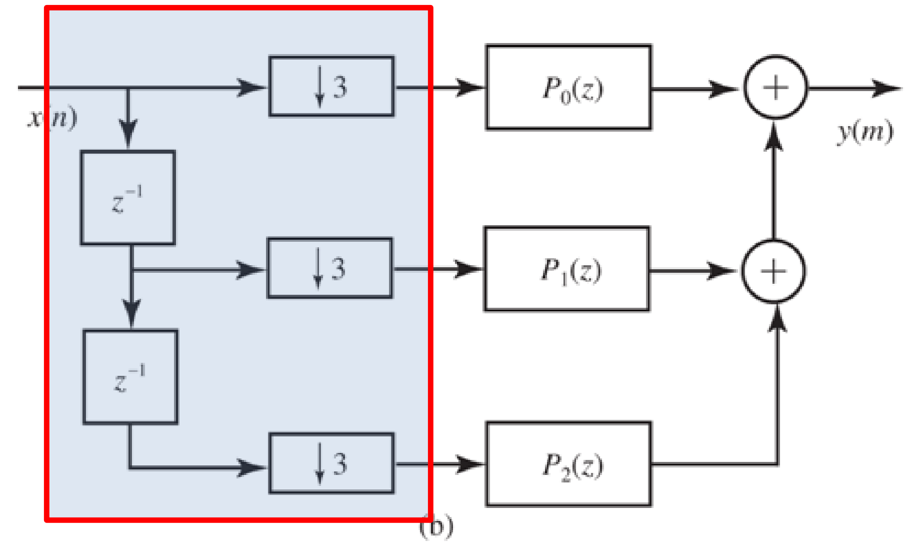
- Decimation is obtained by passing the input signal through an FIR filter before down-sampling by a factor of D
- The filter is operated at the sampling rate F_x which is higher than F_{y},
- Only one data out of D processed data, $v(n)$, is used as a valid output, i.e., wasting computation resource.
- By using the polyphase decomposition concept, the filter is decomposed into D branches
- With the noble identity, the down samplers can be moved before the polyphase filters

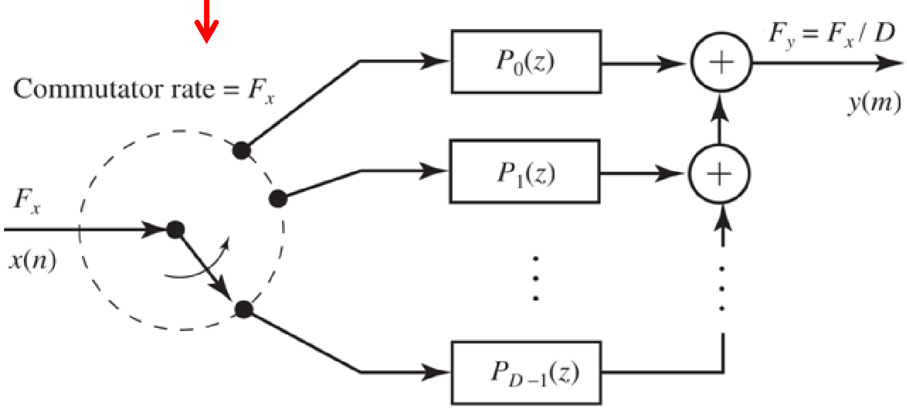




Decimation system

Polyphase Structures for Decimation

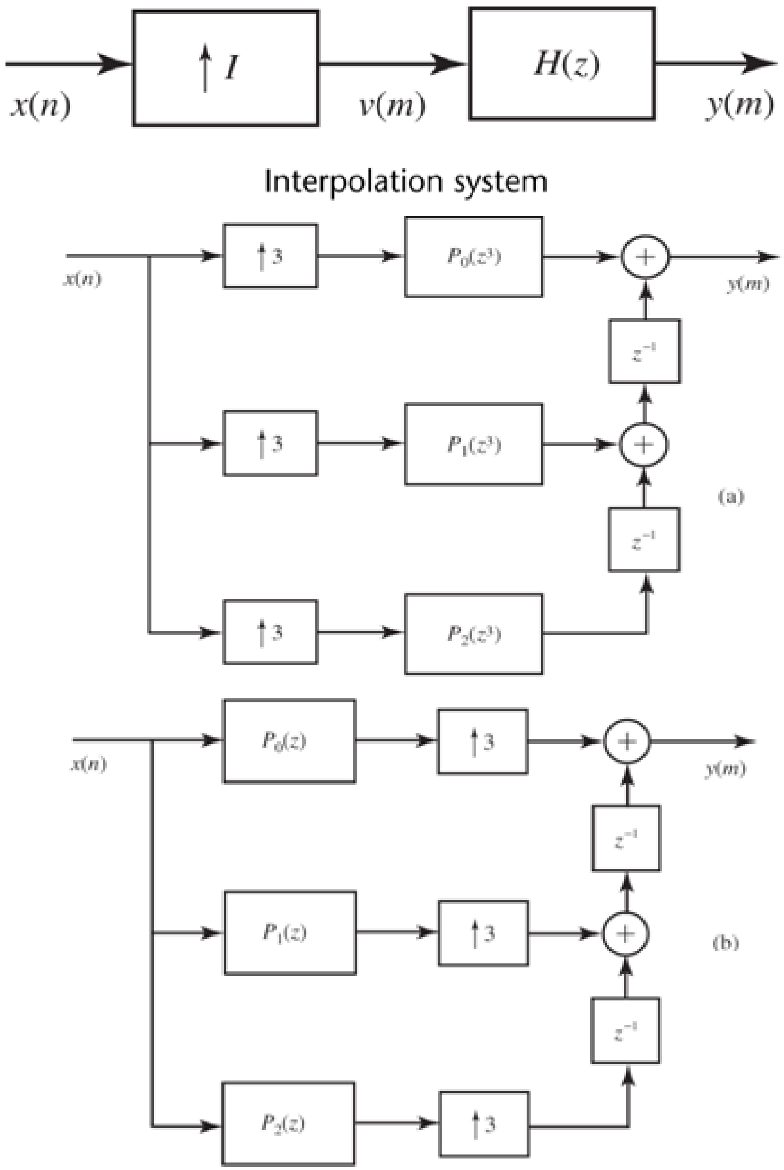




Decimation using a polyphase filter and a commutator

- The filtering process is operated at the lower sampling rate, i.e., after the down sampling process
- All outputs of the polyphase filters are used as valid outputs.
- An alternative implementation uses a commentator to distribute the input data
- Each polyphase filter receives one input data and produces one output within every *D* input sampling periods.

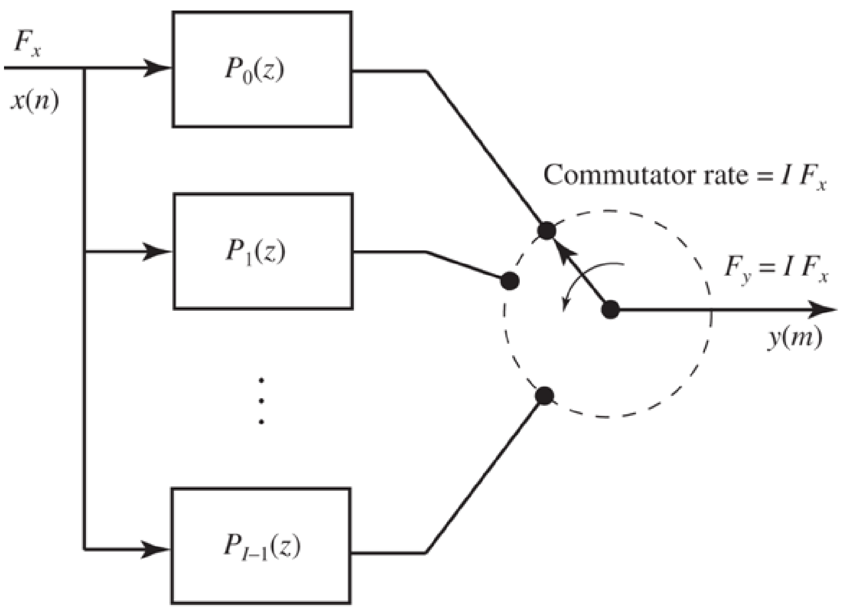
Polyphase Structures for Interpolation



- The up-sampler inserts I-1 zero samples before filtering.
- The filtering process is performed at a higher sampling rate
- The filter $H(z)$ is decomposed into a polyphase structure.
- Using the second noble identity, the up-sampler is moved after the polyphase filters
- The polyphase filters are operated at the input sampling rate and do not process the inserted zeros.

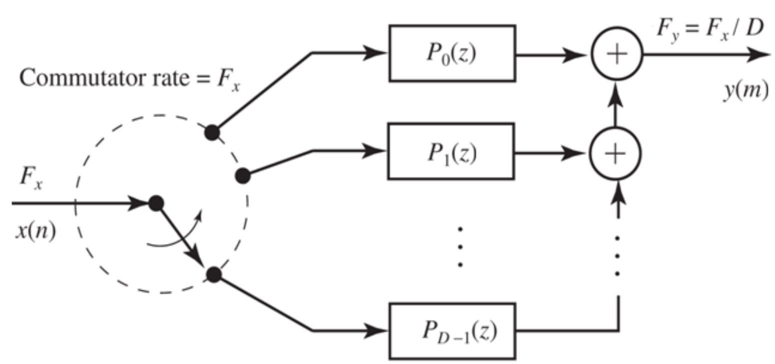
Polyphase Structures for Interpolation

- For each input sample, each polyphase filter produces one output
- The up-sampler in each branch insert I-1 zeros after each output of the polyphase filter.
- Similarly, a commutator is used to collect the output of the polyphase filters



Interpolation using a polyphase filter and a commutator

- It is noted that the polyphase structure for a decimator can be obtained by transposing the polyphase structure for an interpolator.



Decimation using a polyphase filter and a commutator

Polyphase Structures

- If $h(n)$ is flat over the range $0 \le |\omega| \le \pi/I$ (for interpolation), each corresponding polyphase filter is flat over the range $0 \le |\omega| \le \pi$, but with different phases (thus, termed as polyphase).

- In the time domain we have

$$p_k(n) = h(nI + k), \qquad k = 0,1,\ldots I - 1 \quad \text{for interpolator}$$

$$p_k(n) = h(nD + k), \qquad k = 0,1,\ldots D - 1 \quad \text{for decimator}$$

- The polyphase filter can be considered as a set of I subfilters connected to a common delay line, so the kth subfilter generates a time shift of $(k/I)T_x$, k=0,1,…,I – 1, relative to the zeroth subfilter

- Therefore, the phases of the subfilters are

$$p_k(\omega) = e^{j\omega k/I} \qquad k = 0,1,2,\ldots,I - 1.$$

- It is also possible to have polyphase structures for both the interpolator and decimator with commutators rotated in a clockwise direction. The polyphase filters are defined by:

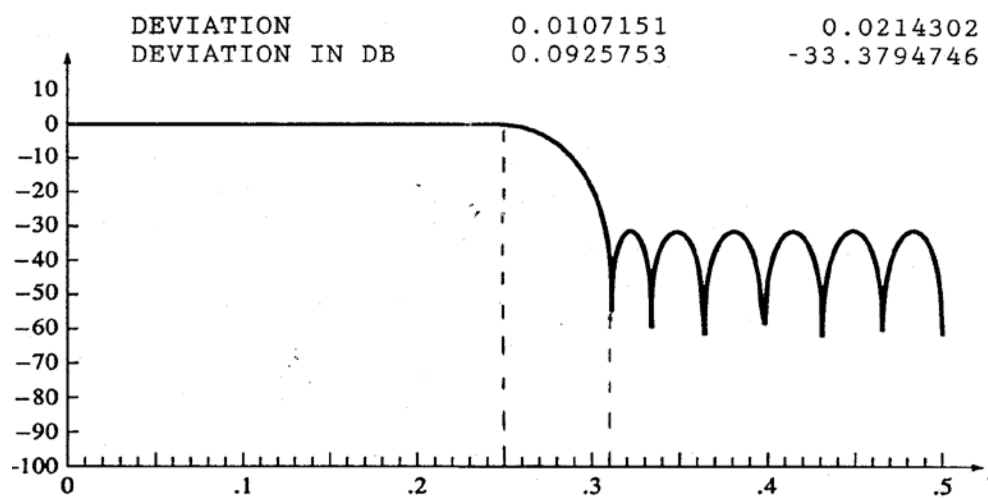
$$p_k(n) = h(nI - k), \qquad k = 0,1,\ldots I - 1 \quad \text{for interpolator}$$

$$p_k(n) = h(nD - k), \qquad k = 0,1,\ldots D - 1 \quad \text{for decimator}$$

Polyphase Structures

- **Example** : Design a decimator that down-samples an input signal $x(n)$ by a factor D = 2. Determine the polyphase filter structure in a decimator realization that employs polyphase filters.
- **Solution**: Assume that the filter has 30 coefficients. Note that the cutoff frequency is $\omega_c = \pi/2$.
- From eqn(2.1)

$$H_D(\omega) = \begin{cases} 1 & |\omega| \le \pi/2 \\ 0 & \text{otherwise} \end{cases}$$



- If passband ripple, transition band, storpband ripple, input sampling frequency, and the desired attenuation in the stop band are given, the filter coefficients can be generated by computer software.

Polyphase Structures

- The polyphase filters obtained from $h(n)$ have impulse responses

$$p_k(n) = h(2n+k) \quad k = 0,1; \quad n = 0,1,\ldots,14$$

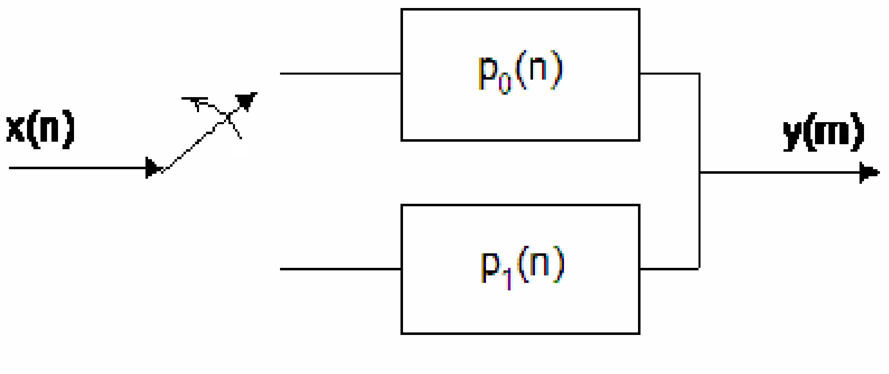


Figure 5.9 A system with two polyphase filters

- Note that $p_0(n) = h(2n)$ and $p_1(n) = h(2n+1)$. Hence one filter consists of the even indexed $h(n)$ and the other consists of the odd-indexed $h(n)$, as shown in Figure 5.9.

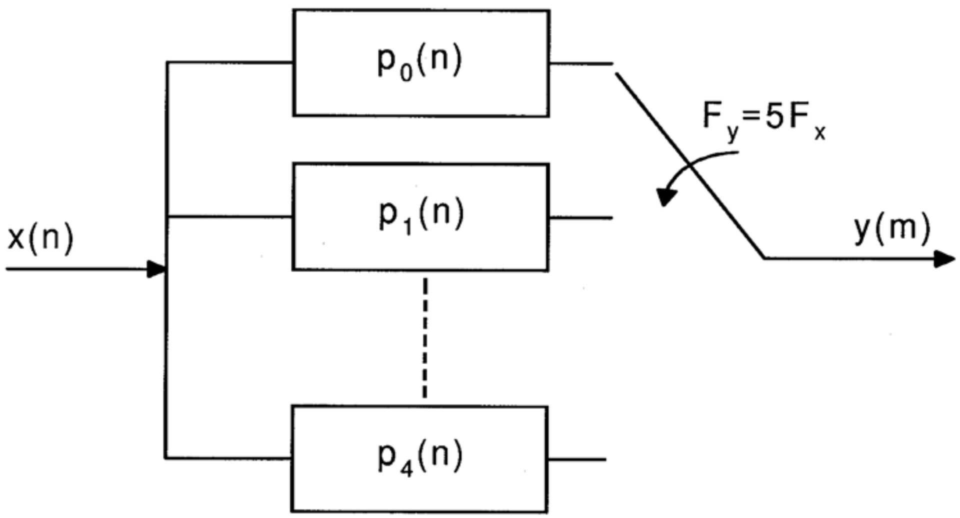
- **Example:** Assume a filter of length N=30 for an interpolator. The cutoff frequency is $\omega_c=\pi/5$. The filter impulse response in the frequency domain is

$$H_I(\omega)=\begin{cases}1 & |\omega|\le\pi/5\\ 0 & \text{otherwise}\end{cases}$$

- The polyphase filters obtained from $h(n)$ have impulse

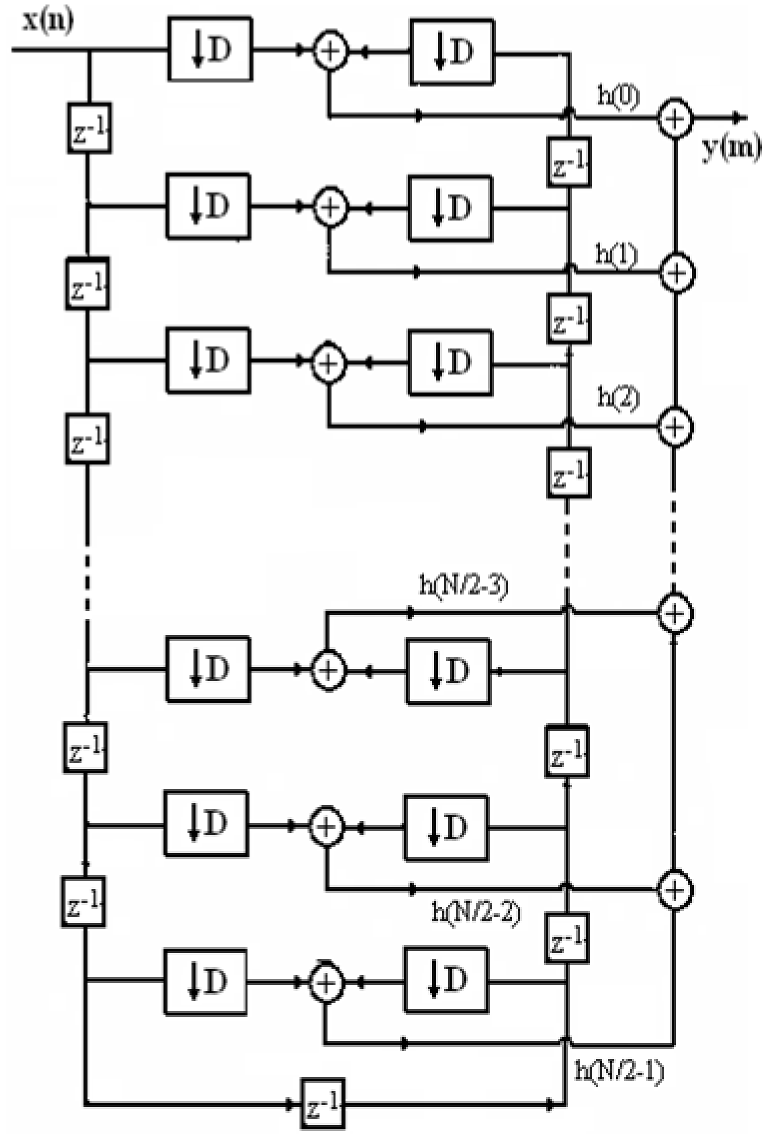
$$p_k(n)=h(5n+k),\quad k=0,1,2,3,4$$

consequently, each subfilter has 6 coefficients.



Using Symmetry Property for Decimation

- We say a decimator is a dual of an interpolator, and vice versa
- An additional saving on computational complexity is to make use the symmetric property of a linear phase filter
- i.e., $h(0) = h(N\text{-}1)$, $h(1) = h(N\text{-}2)$..., (see the Figure) for a realization of a decimator



Efficient realization of decimation by using symmetr in FIR filter

Summary

- Able to make use the properties and signal flow graphs to derive polyphase structures to reduce the computational complexities.
- Why do polyphase structures reduce the computational complexity?
- How to group the filter coefficients for the polyphase structures?