Chapter 2
Multirate Signal Processing

Digital signals are obtained from continuous time signals via sampling operation. Continuous time signals can be considered as digital signals having infinite number of samples. Sampling is nothing but selecting some of these samples and forming a mathematical sequence called digital signal. And these digital signals can be in periodic or non-periodic forms. The number of samples taken from a continuous time signal per-second is determined by sampling frequency. As the sampling frequency increases, the number of samples taken from a continuous time signal per-second increases, as well. As the technology improves, new and better electronic devices are being produced. This also brings the compatibility problem between old and new devices. One such problem is the speed issue of the devices. Consider a communication device transmitting digital samples taken from a continuous time signal at a high speed. This means high sampling frequency, as well. If the speed of the receiver device is not as high as the speed of the transmitter device, then the receiver device cannot accommodate the samples taken from the transmitter. This results in communication error. Hence, we should be able to change the sampling frequency according to our needs.

We should be able to increase or decrease the sampling frequency without changing the hardware. We can do this using additional hardware components at the output of the sampling devices. One way of decreasing the sampling frequency is the elimination of some of the samples of the digital signal. This is also called sampling of digital signals, or decimation of digital signals, or compression of digital signals. On the other hand, after digital transmission, at the receiver side before digital to analog conversion operation, we can increase the number of samples. This is called upsampling, or increasing sampling rate, or increasing sampling frequency. If we have more samples for a continuous time signal, when it is reconstructed from its samples, we obtain a better continuous time signal. In this chapter, we will learn how to manipulate digital signals, which means, changing their sampling rates, reconstruction of a long digital sequence from a short version of it, de-multiplexing and multiplexing of digital signals via hardware units etc.

O. Gazi, *Understanding Digital Signal Processing*, Springer Topics
in Signal Processing 13, DOI 10.1007/978-981-10-4962-0_2

2.1 Sampling Rate Reduction by an Integer Factor (Downsampling, Compression)

To represent a continuous time by digital sequences, we take samples from the continuous time signal according a sampling frequency and form a mathematical sequence. If the mathematical sequence contains too many samples, we can omit some of these samples and keep the rest of the samples for transmission, storage, processing etc.

Let's give another example from real life. Assume that you want to send 500 students to a university in a foreign country. The selected students represent your university and from each department 10 students were selected. Later on you think that the travel cost of 500 students is too much and decide on reducing the number of selected students.

A continuous time signal can be considered as a digital signal containing infinite number of samples for any time interval. Sampling of analog signals is nothing but selecting a finite number of samples from the infinite sample sets of the analog signals for the given time interval. The downsampling operation can be considered as the sampling of digital signals. In this case a digital signal containing a number of samples for a given time interval is considered and for the given interval, some of the samples of the digital signal are selected and a new digital signal is formed. This operation is called downsampling. During the downsampling some of the samples of a digital signal are selected and the remaining samples are omitted.

The downsampling operation is illustrated in Fig. 2.1 where $x[n]$ is the signal to be downsampled and $y[n]$ is the signal obtained after downsampling $x[n]$, i.e., after omitting sampled from $x[n]$, and M is the downsampling factor.

Given $x[n]$ to find the compressed signal, i.e., downsampled signal, $y[n]$, we divide the time axis of $x[n]$ by M and keep only integer division results and omit all non-integer division results. Let's illustrate this operation by an example.

Example 2.1 A digital signal expressed as a mathematical sequence is given as

$$x[n] = [3.3 \quad -2.5 \quad 1.2 \quad 4.5 \quad 5.5 \quad -2.3 \quad \underbrace{5.0}_{n=0} \quad 6.2 \quad 3.4 \quad 2.3 \quad -4.4 \quad 3.2 \quad 2.0]$$

Find the downsampled $y[n] = x[3n]$.

$$x[n] \longrightarrow \boxed{\downarrow M} \longrightarrow y[n] = x[Mn]$$

Fig. 2.1 Downsampling operation

Solution 2.1 Let's write the time index values of the signal, $x[n]$ explicitly follows

$$x[n] = [\underbrace{3.3}_{n=-6} \quad \underbrace{-2.5}_{n=-5} \quad \underbrace{-1.2}_{n=-4} \quad \underbrace{4.5}_{n=-3} \quad \underbrace{5.5}_{n=-2} \quad \underbrace{-2.3}_{n=-1} \quad \underbrace{5.0}_{n=-0} \quad \underbrace{6.2}_{n=1} \quad \underbrace{3.4}_{n=2} \quad \underbrace{2.3}_{n=3} \quad \underbrace{-4.4}_{n=4} \quad \underbrace{3.2}_{n=5} \quad \underbrace{2.0}_{n=6}].$$

In the second step, we divide the time axis of $x[n]$ by 3, this is illustrated in

$$[\underbrace{\mathbf{3.3}}_{\mathbf{n=-\frac{6}{3}}} \quad \underbrace{-2.5}_{n=-\frac{5}{3}} \quad \underbrace{-1.2}_{n=-\frac{4}{3}} \quad \underbrace{\mathbf{4.5}}_{\mathbf{n=-\frac{3}{3}}} \quad \underbrace{5.5}_{n=-\frac{2}{3}} \quad \underbrace{-2.3}_{n=-\frac{1}{3}} \quad \underbrace{\mathbf{5.0}}_{\mathbf{n=-\frac{0}{3}}} \quad \underbrace{6.2}_{n=\frac{1}{3}} \quad \underbrace{3.4}_{n=\frac{2}{3}} \quad \underbrace{\mathbf{2.3}}_{\mathbf{n=\frac{3}{3}}} \quad \underbrace{-4.4}_{n=\frac{4}{3}} \quad \underbrace{3.2}_{n=\frac{5}{3}} \quad \underbrace{\mathbf{2.0}}_{\mathbf{n=\frac{6}{3}}}].$$

where divisions' yielding integer results are shown in bold numbers and these divisions are given alone as follows

$$[\underbrace{3.3}_{n=-\frac{6}{3}} \quad \underbrace{4.5}_{n=-\frac{3}{3}} \quad \underbrace{5.0}_{n=-\frac{0}{3}} \quad \underbrace{2.3}_{n=-\frac{3}{3}} \quad \underbrace{2.0}_{n=\frac{6}{3}}]$$

and when the divisions are done, we obtain the downsampled signal as

$$y[n] = [\underbrace{3.3}_{n=-2} \quad \underbrace{4.5}_{n=-1} \quad \underbrace{5.0}_{n=0} \quad \underbrace{2.3}_{n=1} \quad \underbrace{2.0}_{n=1}]$$

As it is seen from the previous example, downsampling a digital signal by M means that from every M samples of the digital signal only one of them is selected and the rest of them are eliminated. As an example, if $y[n] = x[6n]$, then from every 6 samples of $x[n]$ only one of them is kept and the other 5 samples are omitted.

Now we ask the question, if sampling frequency is f_s and downsampling factor is M, after downsampling operation how many samples per-second are available at the downsampler output? The answer is given in the block diagram in Fig. 2.2.

Where $\lceil . \rceil$ is the upper-floor operation. If f_s is a multiple of M, the diagram in Fig. 2.2 reduces to the one in Fig. 2.3.

Example 2.2 Interpret the block diagram given in Fig. 2.4.

Solution 2.2 At the input of the block, we receive 300 samples per-second which are obtained from an analog signal via sampling operation. At the output of the downsampler only 1 of every 3 samples is kept and the other 2 samples are omitted.

$$f_s \rightarrow \boxed{\downarrow M} \rightarrow \left\lceil \frac{f_s}{M} \right\rceil$$

Fig. 2.2 Sampling frequency at the downsampler output

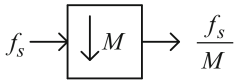


Fig. 2.3 Sampling frequency at the downsampler output when f_s is a multiple of M

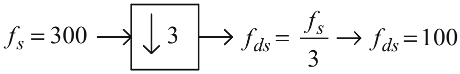


Fig. 2.4 Downsampler for Example 2.2

That means at the output of the downsampler, 100 samples every per-second are released.

Example 2.3 Find the Fourier series representation of

$$p[n] = \sum_{r=-\infty}^{\infty} \delta[n - rM]. \tag{2.1}$$

Solution 2.3 The given signal is a periodic signal with period M. Its Fourier series coefficients are computed as

$$P[k] = \frac{1}{M} \sum_{n=-\frac{M-1}{2}}^{\frac{M+1}{2}} p[n]e^{-j\frac{2\pi}{M}kn} \rightarrow P[k] = \frac{1}{M} \sum_{n=-\frac{M-1}{2}}^{\frac{M+1}{2}} \delta[n]e^{-j\frac{2\pi}{M}kn} \rightarrow P[k] = \frac{1}{M}. \tag{2.2}$$

Using the Fourier series coefficients in (2.2), the Fourier series representation of (2.1) can be written as

$$p[n] = \sum_{k,M} P[k]e^{j\frac{2\pi}{M}kn} \rightarrow p[n] = \frac{1}{M} \sum_{k,M} e^{j\frac{2\pi}{M}kn}. \tag{2.3}$$

The mathematical expression $p[n] = \sum_{r=-\infty}^{\infty} \delta[n - rM]$ can also be written as

$$p[n] = \begin{cases} 1 & \textit{if} \quad n = 0, \pm M, \pm 2M, \ldots \\ 0 & \textit{otherwise}. \end{cases} \tag{2.4}$$

And equating the right hand sides of (2.3) and (2.4) to each other, we get the equality

$$\frac{1}{M} \sum_{k=0}^{M-1} e^{j\frac{2\pi}{M}kn} = \begin{cases} 1 & n = 0, \pm M, \pm 2M, \ldots \\ 0 & \textit{otherwise}. \end{cases} \tag{2.5}$$

For the expression in (2.5), if we change the sign of n appearing on both sides of the equation, we obtain an alternative expression for (2.5) as

$$\frac{1}{M}\sum_{k=0}^{M-1} e^{-j\frac{2\pi}{M}kn} = \begin{cases} 1 & n = 0, \pm M, \pm 2M, \ldots \\ 0 & otherwise. \end{cases} \tag{2.6}$$

2.1.1 Fourier Transform of the Downsampled Signal

Let's find the Fourier transform of the compressed signal $y[n] = x[Mn]$. The Fourier transform of $y[n]$ can be calculated using

$$Y_n(w) = \sum_{n=-\infty}^{\infty} x[Mn]e^{-jwn} \tag{2.7}$$

where defining $r \triangleq Mn$, we obtain

$$Y_n(w) = \sum_{r=0,\pm M,\pm 2M} x[r]e^{-jw\frac{r}{M}} \tag{2.8}$$

which can be written after parameter changes as

$$Y_n(w) = \sum_{n=0,\pm M,\pm 2M} x[n]e^{-jw\frac{n}{M}} \tag{2.9}$$

The frontiers of the sum symbol in (2.9) can be changed to $-\infty$ and ∞ if (2.1) is used in (2.9) as

$$Y_n(w) = \sum_{n=-\infty}^{\infty} x[n] \sum_{r=-\infty}^{\infty} \delta[n - rM]e^{-jw\frac{n}{M}}$$

where replacing $\sum_{r=-\infty}^{\infty} \delta[n - rM]$ by its Fourier series representation, we get

$$Y_n(w) = \sum_{n=-\infty}^{\infty} x[n]\frac{1}{M}\sum_{k,M} e^{-j\frac{2\pi}{M}kn}e^{-jw\frac{n}{M}} \tag{2.10}$$

which can be rearranged as

$$Y_n(w) = \frac{1}{M}\sum_{k,M} \underbrace{\sum_{n=-\infty}^{\infty} x[n]e^{-j\frac{w+k2\pi}{M}n}}_{=X_n\left(\frac{w+k2\pi}{M}\right)} \tag{2.11}$$

The expression in (2.11) can be reduced to

$$Y_n(w) = \frac{1}{M}\sum_{k,M} X_n\left(\frac{w + k2\pi}{M}\right). \tag{2.12}$$

In (2.10), if (2.5) was used, then we would obtain

$$Y_n(w) = \frac{1}{M}\sum_{k,M} X_n\left(\frac{w - k2\pi}{M}\right). \tag{2.13}$$

Hence, considering (2.12) and (2.13), we can write the Fourier transform of $y[n] = x[Mn]$ as

$$Y_n(w) = \frac{1}{M}\sum_{k=0}^{M-1} X_n\left(\frac{w \pm k2\pi}{M}\right). \tag{2.14}$$

Example 2.4 If $y[n] = x[Mn]$ the relation between Fourier transforms of $x[n]$ and $y[n]$ is given as

$$Y_n(w) = \frac{1}{M}\sum_{k=0}^{M-1} X_n\left(\frac{w \pm k2\pi}{M}\right).$$

Using the inverse Fourier transform expression for $y[n]$, i.e.,

$$y[n] = \frac{1}{2\pi}\int_{w=0}^{2\pi} Y_n(w)e^{jwn}dw \tag{2.15}$$

show that $y[n] = x[Mn]$.

Solution 2.4 The inverse Fourier transform is given as

$$y[n] = \frac{1}{2\pi}\int_{0}^{2\pi} Y_n(w)e^{jwn}dw$$

where inserting

$$Y_n(w) = \frac{1}{M}\sum_{k=0}^{M-1} X_n\left(\frac{w \pm k2\pi}{M}\right)$$

we get

$$y[n] = \frac{1}{2\pi M}\sum_{k=0}^{M-1}\int_{0}^{2\pi} X_n\left(\frac{w + k2\pi}{M}\right)e^{jwn}dw \tag{2.16}$$

In (2.16), if we let $\lambda = \frac{w+k2\pi}{M}$, then $dw = Md\lambda$, and changing the frontiers of the integral (2.16) reduces to

$$y[n] = \frac{1}{2\pi}\sum_{k=0}^{M-1}\int_{\frac{k2\pi}{M}}^{\frac{(k+1)2\pi}{M}} X_n(\lambda)e^{jM\lambda n}d\lambda. \tag{2.17}$$

If (2.17) is expanded for all k values, we obtain

$$\begin{aligned} y[n] = {} & \frac{1}{2\pi}\int_{0}^{\frac{2\pi}{M}} X_n(\lambda)e^{jM\lambda n}d\lambda + \frac{1}{2\pi}\int_{\frac{2\pi}{M}}^{\frac{4\pi}{M}} X_n(\lambda)e^{jM\lambda n}d\lambda + \cdots \\ & + \frac{1}{2\pi}\int_{\frac{M-1}{M}2\pi}^{2\pi} X_n(\lambda)e^{jM\lambda n}d\lambda \end{aligned} \tag{2.18}$$

where using the property $\int_a^b(\cdot) + \int_b^c(\cdot) = \int_a^c(\cdot)$ and changing λ with w, we get the expression

$$y[n] = \frac{1}{2\pi}\int_{0}^{2\pi} X_n(w)e^{jMwn}dw. \tag{2.19}$$

When (2.19) is compared to

$$x[n] = \frac{1}{2\pi}\int_{0}^{2\pi} X_n(w)e^{jwn}dw$$

it is seen that $y[n] = x[Mn]$.

2.1.2 How to Draw the Frequency Response of Downsampled Signal

To draw the graph of

$$Y_n(w) = \frac{1}{M}\sum_{k=0}^{M-1} X_n\left(\frac{w - k2\pi}{M}\right)$$

students usually expand the summation as

$$Y_n(w) = \frac{1}{M}X_n\left(\frac{w}{M}\right) + \frac{1}{M}X_n\left(\frac{w-2\pi}{M}\right) + \frac{1}{M}X_n\left(\frac{w-4\pi}{M}\right) + \cdots \quad (2.20)$$

and try to draw each shifted graph and sum the shifted graphs. However, this approach is too time consuming and error-prone. Instead of this approach, we will suggest a simpler method to draw the graph of $Y_n(w)$ as explained in the following lines.

Since $Y_n(w)$ is the Fourier transform of the digital signal $y[n]$, then $Y_n(w)$ is a periodic signal and its period equals to 2π.

To draw the graph of $Y_n(w)$, we can follow the following steps.

Step 1: First one period of $X_n(w)$ around origin is drawn. For this purpose, the frequency interval is chosen as $-\pi < w \leq \pi$.
Step 2: Considering one period of $X_n(w)$ around origin, we draw one period of $\frac{1}{M}X_n\left(\frac{w}{M}\right)$. To draw (in one period) the graph of $\frac{1}{M}X_n\left(\frac{w}{M}\right)$, we multiply the horizontal axis of $X_n(w)$ by M, and multiply the vertical axis of $X_n(w)$ by $\frac{1}{M}$.
Step 3: In Step 3, we shift the resulting graph in Step 2 to the left and right by multiples of 2π and sum the shifted replicas.

Let's now give an example to illustrate the topic.

Example 2.5 One period of the Fourier transform of $x[n]$ is depicted in Fig. 2.5. Draw the Fourier transform of $y[n] = x[2n]$, i.e., draw $Y_n(w)$.

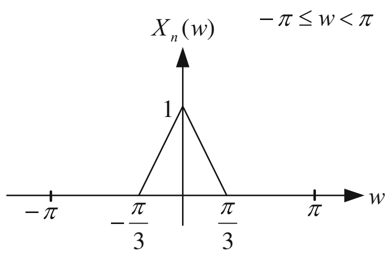


Fig. 2.5 One period of the Fourier transform of $x[n]$

Solution 2.5 First let's draw the graph of $Y_{1n}(w) = \frac{1}{2}X_n(\frac{w}{2})$. For this purpose, we multiply the frequency axis of $X_n(w)$ by 2 and vertical axis of $X_n(w)$ by $\frac{1}{2}$. The resulting graph is shown in Fig. 2.6.

In the second step, we shift the graph of $Y_{1n}(w)$ to the left and right by multiples of 2π and sum the shifted graphs. In other words, we draw the graph of $Y_n(w) = \sum_{k=-\infty}^{\infty} Y_{1n}(w - k2\pi)$. The shifted graphs and their summation result are depicted in Figs. 2.7, 2.8, and 2.9.

Right Shifted Functions:

Left Shifted Functions:

Sum of the Shifted Functions:

Exercise: One period of the Fourier transform of $x[n]$ is depicted in Fig. 2.10. Draw the Fourier transform of $y[n] = x[3n]$, i.e., draw $Y_n(w)$.

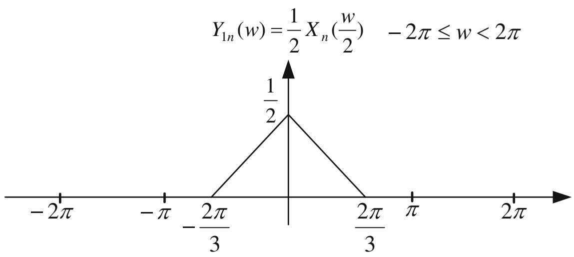


Fig. 2.6 The graph of $\frac{1}{2}X_n(\frac{w}{2})$

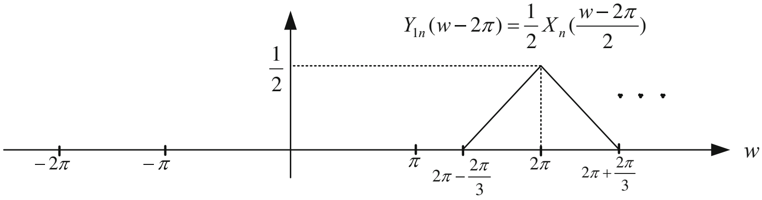


Fig. 2.7 Right shifted functions

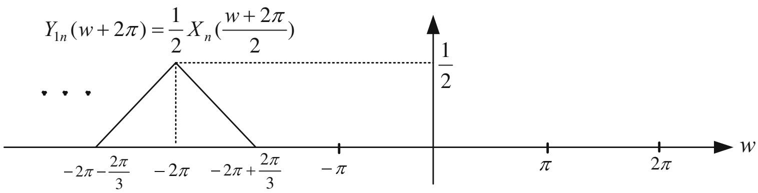


Fig. 2.8 Left shifted functions

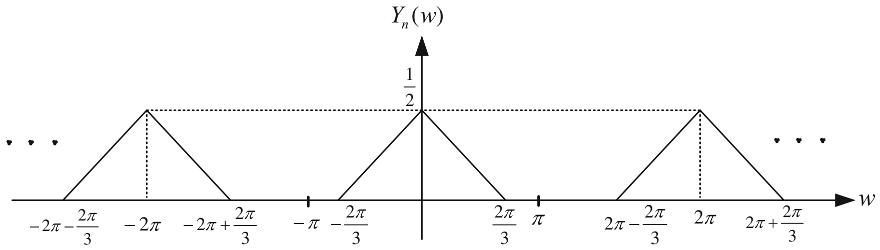
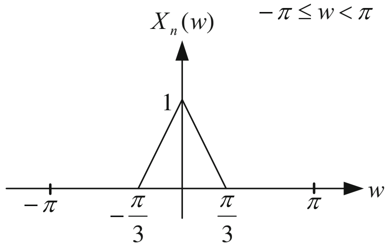


Fig. 2.9 Sum of the shifted functions

Fig. 2.10 One period of the Fourier transform of $x[n]$



2.1.3 Aliasing in Downsampling

A digital signal is nothing but a mathematical sequence obtained via sampling of a continuous time signal. If we have sufficient number of samples, we can reconstruct the continuous time signal from its samples.

If we have too many samples, generated during the sampling operation we can eliminate some of these excessive samples via the downsampling operation. However, while performing the downsampling operation, we should be careful to keep sufficient number of samples in the digital signal such that the reconstruction of the continuous time signal is still possible after downsampling operation.

If we eliminate a number of samples more than a threshold value, the rest of the samples may not be sufficient to reconstruct the continuous time signal and this effect is seen as the aliasing in the spectrum graph of the downsampled signal.

Example 2.6 Assume that we have a low pass continuous time signal with bandwidth $f_N = 40$ Hz. We choose the sampling frequency according to the criteria $f_s > 2f_N \rightarrow f_s > 80$ as $f_s = 120$. This means that we take 120 samples per-second from the continuous time signal. However, our chosen sampling frequency is not very cost efficient.

The lower limit for the sampling frequency is $f_s > 80$ which means that the minimum sampling frequency can be chosen as $f_s = 81$. However we use $f_s = 120$ which means that every per-second we transmit 120 − 81 = 39 excessive samples which are not necessary to reconstruct the continuous time signal. We can reconstruct the continuous time signal using only 81 samples. We can omit the excessive 39 samples via downsampling operation.

Let's now determine the criteria for no aliasing in downsampling operation. After downsampling operation, we have $\frac{f_s}{M}$ remaining samples per-second. If this number of remaining samples is greater than $2f_N$, then no aliasing occurs. That is if

$$\frac{f_s}{M} > 2f_N \rightarrow M < \frac{f_s}{2f_N} \tag{2.21}$$

is satisfied, then aliasing is not seen in the spectrum of the downsampled signal. Let's simplify (2.21) more as

$$M < \frac{f_s}{2f_N} \rightarrow M < \frac{1}{2\underbrace{T_s f_N}_{f_D}} \tag{2.22}$$

where f_D is the digital frequency, and manipulating more, we have

$$M < \frac{1}{2f_D} \rightarrow M < \frac{\pi}{2\pi f_D} \rightarrow M < \frac{\pi}{w_D} \rightarrow Mw_D < \pi \tag{2.23}$$

where w_D is the angular digital frequency.

Let's now graphically illustrate the no aliasing criteria after downsampling operation. Assume that one period of the Fourier transform of the digital signal $x[n]$ to be downsampled is given as in Fig. 2.11. Let $y[n] = x[Mn]$ be the downsampled signal.

Depending on the value of M, we can draw the two possible graphs of $\frac{1}{M}X_n\left(\frac{w}{M}\right)$ as shown in Figs. 2.12 and 2.13.

When the graph in Fig. 2.12 is shifted to the left and right by multiples of 2π, no overlapping occurs among shifted graphs. However, this case does not hold for the graph shown in Fig. 2.13. If the graph shown in Fig. 2.13 is shifted to the left and right by multiples of 2π, overlapping is observed between shifted replicas, and this situation is depicted in Fig. 2.14.

Example 2.7 The continuous time signal $x_c(t) = \cos(6000\pi t)$ is sampled with sampling period $T_s = \frac{1}{8000}$ and the digital sequence $x[n]$ is obtained. Next the digital

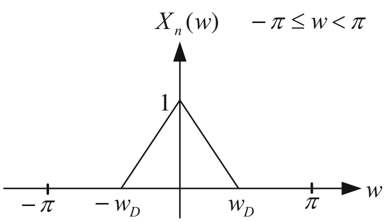


Fig. 2.11 One period of the Fourier transform of the digital signal $x[n]$ to be downsampled

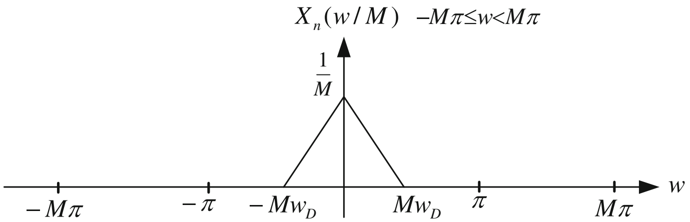


Fig. 2.12 Case-1: Graph of $\frac{1}{M}X_n\left(\frac{w}{M}\right)$

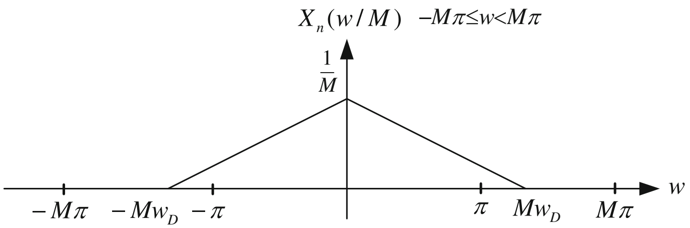


Fig. 2.13 Case-2: Graph of $\frac{1}{M}X_n\left(\frac{w}{M}\right)$

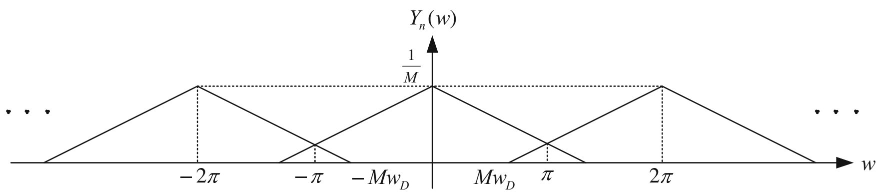


Fig. 2.14 Aliasing in downsampled signal spectrum graph

signal $x[n]$ is downsampled and $y[n] = x[4n]$ is obtained. Decide whether aliasing occurs in spectrum of $y[n]$ or not.

Solution 2.7 If the given continuous time signal is compared to $\cos(2\pi ft)$, the frequency of the continuous time signal is found as $f = 3000$ Hz. And the sampling frequency is $f_s = 8000\,\text{Hz}$. After downsampling operation sampling frequency reduces to $f_s = \frac{8000}{4} = 2000$ Hz and this value is less than $2f = 6000$ Hz. This means that aliasing is seen in the spectrum of $y[n]$.

Exercise: For the system in Fig. 2.15, $x_c(t) = \cos(5000\pi t)$, $T_s = \frac{1}{10{,}000}$, and $M = 2$. According to given information, draw the Fourier transforms of the signals $x_c(t), x[n], y[n]$, and $y_r(t)$, and also write the time domain expression for $y_r(t)$.

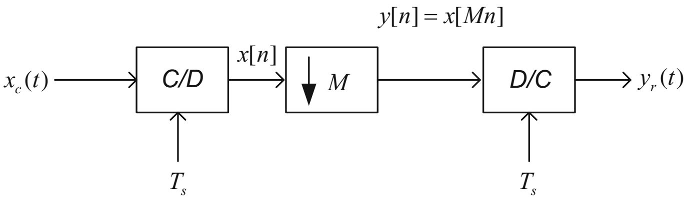


Fig. 2.15 Signal processing system for exercise

2.1.4 Interpretation of the Downsampling in Terms of the Sampling Period

If $x[n] = x_c(nT_s)$, then for the downsampled signal $y[n] = x[Mn] \rightarrow y[n] = x_c(n\underbrace{MT_s}_{T_s'})$ new sampling period is $T_s' = MT_s$ which is an integer multiple of T_s. The digital signal obtained from $x_c(t)$ using sampling period T_s is shown in Fig. 2.16.

The digital signal $x[n]$ in Fig. 2.16 is written as a mathematical sequence as

$$x[n] = [\cdots a \quad b \quad c \quad d \quad e \quad f \quad \underbrace{g}_{n=0} \quad h \quad i \quad j \quad k \quad l \quad m \cdots].$$

Now consider $y[n] = x[2n] \rightarrow y[n] = x_c(n2T_s)$, in this case the samples are taken from $x_c(t)$ at every $T_s' = 2T_s$. This operation is illustrated in Fig. 2.17.

The digital signal $y[n]$ in Fig. 2.17 can be written as a mathematical sequence as

$$y[n] = [\cdots \; a \quad c \quad e \quad \underbrace{g}_{n=0} \quad i \quad k \quad m \; \cdots].$$

Similarly, if $g[n] = x[4n] \rightarrow g[n] = x_c(n4T_s)$, the samples are taken from $x_c(t)$ at every $T_s' = 4T_s$. This operation is illustrated in Fig. 2.18.

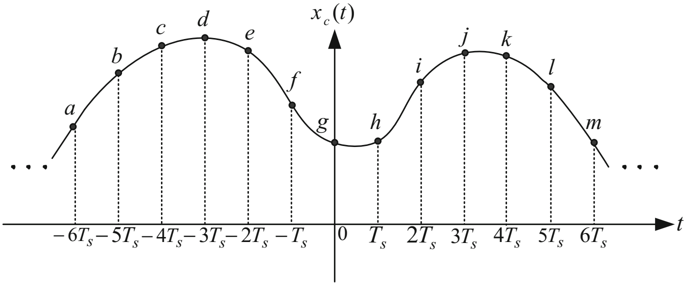


Fig. 2.16 Sampling of the continuous time signal with sampling period T_s

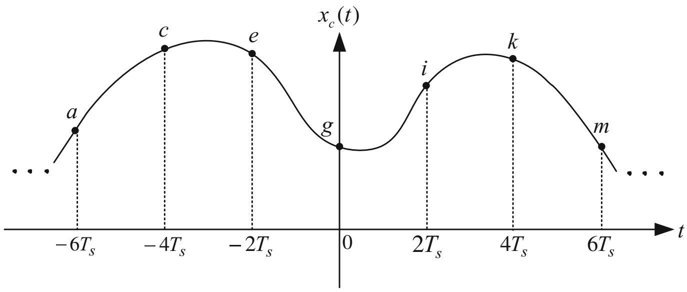


Fig. 2.17 Sampling of the continuous time signal with sampling period $2T_s$

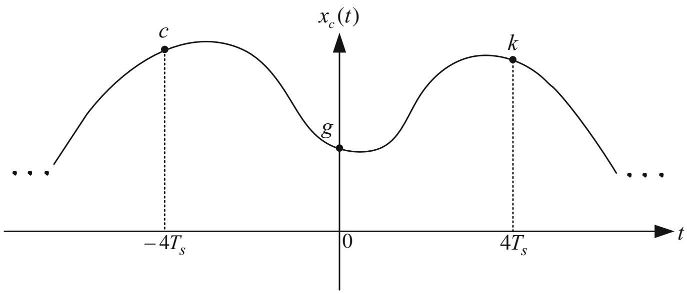


Fig. 2.18 Sampling of the continuous time signal with sampling period $4T_s$

The digital signal $g[n]$ in Fig. 2.18 can be written as a mathematical sequence as

$$y[n] = [\cdots\ c \quad \underbrace{g}_{n=0} \quad k\ \cdots].$$

Example 2.8 For the signal processing system given in Fig. 2.19, $x_c(t) = \cos(5000\pi t)$, $T_s = \frac{1}{8000}$, and $M = 3$. Using the given information, calculate and draw the Fourier transforms of the signals $x_c(t), x[n], y[n]$, and $y_r(t)$. Besides, write the time domain expression for $y_r(t)$.

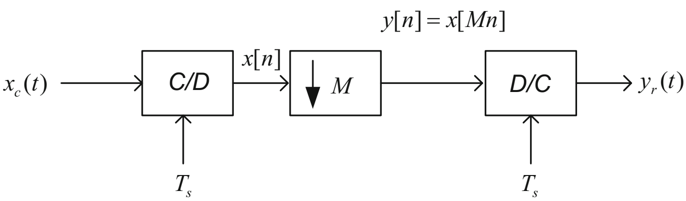


Fig. 2.19 Signal processing system for Example 2.8

Solution 2.8 Before starting to the solution, let's provide some background information as

$$\mathrm{Cos}(\theta) = \frac{1}{2}\left(e^{j\theta} + e^{-j\theta}\right) \quad FT\left\{e^{jw_0 t}\right\} = 2\pi\delta(w - w_0) \tag{2.24}$$

$$FT\{\cos(w_N t)\} = \pi(\delta(w - w_N) + \delta(w + w_N)). \tag{2.25}$$

Accordingly, the Fourier transform of $x_c(t)$ is found as

$$X_c(w) = \pi(\delta(w - 5000\pi) + \delta(w + 5000\pi)).$$

and graphically it is shown in Fig. 2.20.

For the given example, since $f_s > 2f_N \rightarrow 8000 > 2 \times 2500$ criteria is satisfied, no aliasing is observed in the Fourier transform of $x[n]$, and for this reason, one period of the Fourier transform of $x[n]$ for the interval $-\pi \le w < \pi$ equals $X_n(w) = \frac{1}{T_s} X_c\left(\frac{w}{T_s}\right)$ which is depicted in Fig. 2.21.

For the downsampled signal, we have $y[n] = x[3n]$, let's draw one period of $Y_n(w) = \frac{1}{3} X_n\left(\frac{w}{3}\right)$ using one period of $X_n(w)$ around origin as in Fig. 2.22 where impulses are labeled with letters so that we can distinguish them while forming the Fourier transform of $y[n]$.

If the graph in Fig. 2.22 is carefully inspected, we see that after downsampling operation one period of the Fourier transform of the downsampled signal extends beyond the interval $(-\pi, \pi)$ in frequency axis. This means that the number of samples omitted is greater than the allowed threshold and for this reason perfect reconstruction of the continuous time signal is not possible anymore. It may be reconstructed with some distortion or the reconstructed signal may be a totally

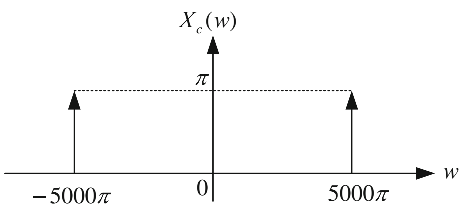


Fig. 2.20 Fourier transform of $x_c(t)$ in Example 2.8

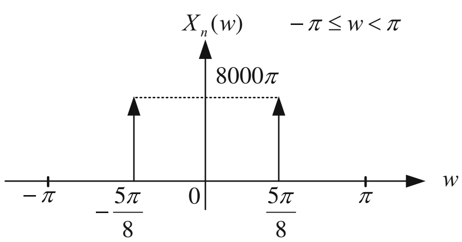


Fig. 2.21 One period of the Fourier transform of $x[n]$ for Example 2.8

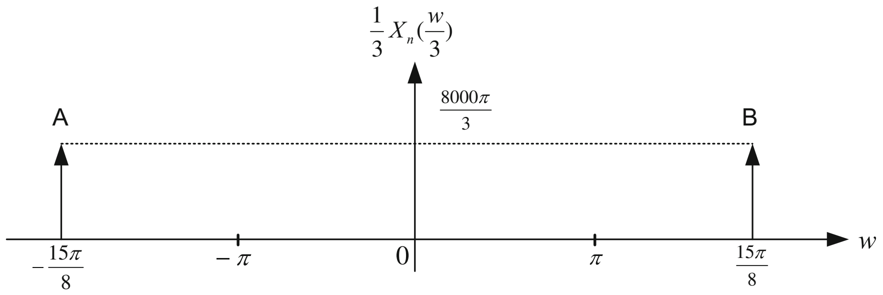


Fig. 2.22 The graph of $\frac{1}{3}X_n\left(\frac{w}{3}\right)$ for Example 2.8

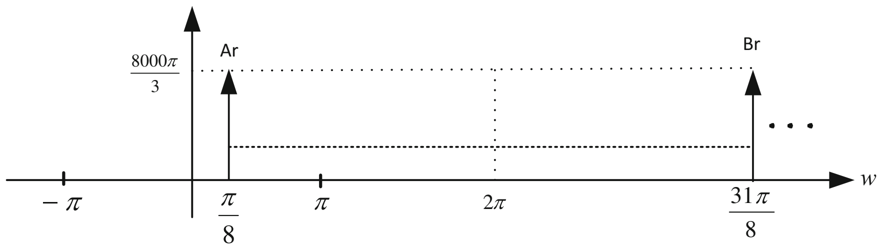


Fig. 2.23 One period of $Y_n(w)$ shifted to the right by 2π

different one. The amount of distortion in the reconstructed continuous time signal depends on the rate of the omitted samples, i.e., rate of the compression or rate of the downsampling. As the number of omitted samples increases, the amount of distortion in the reconstructed signal increases, as well.

To get the graph of $Y_n(w)$, we shift its one period depicted in Fig. 2.22 to the left and to the right by multiples of 2π and sum the shifted replicas. The right shifted graph by 2π is given in Fig. 2.23.

And the left shifted graph by 2π is shown in Fig. 2.24a.

Summing the centered, right shifted, and left shifted graphs, we get the graph of $Y_n(w)$ as shown in Fig. 2.24b.

Now let's find the expression for the reconstructed signal $y_r(t)$. For this purpose, we consider the graph of $Y_n(w)$ for the interval $-\pi \le w < \pi$ and draw $Y_r(w) = T_s X_n(T_s w)$. To achieve this, we divide the frequency axis by T_s and multiply the amplitudes by T_s. These operations generate the graph depicted in Fig. 2.25.

If the inverse Fourier transform of $Y_r(w)$ depicted in Fig. 2.25 is calculated, we obtain the time domain expression of the reconstructed signal as

$$y_r(t) = \frac{1}{3}\cos(1000\pi t)$$

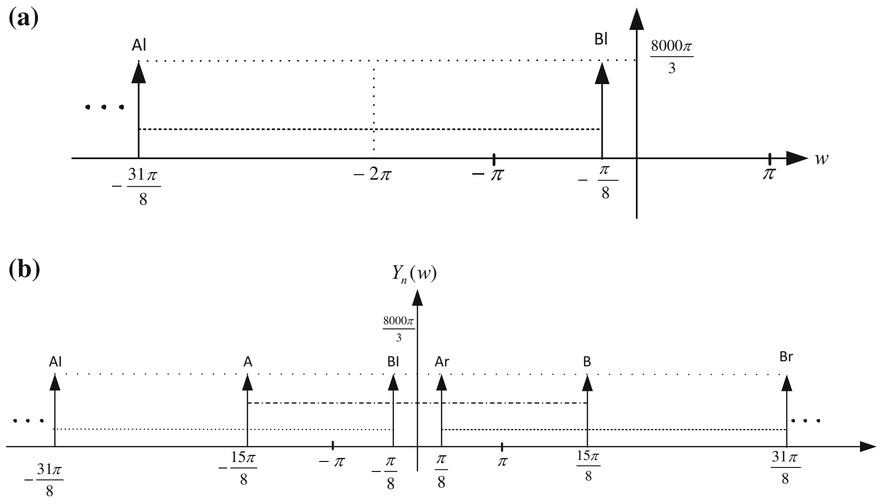
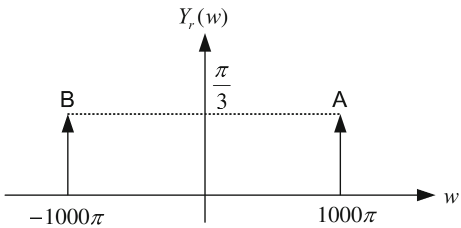


Fig. 2.24 **a** One period of $Y_n(w)$ shifted to the left by 2π. **b** The graph of $Y_n(w)$ for Example 2.8

Fig. 2.25 Fourier transform of the reconstructed signal for Example 2.8



which is quite different from the sampled signal $x_c(t) = \cos(5000\pi t)$. The reason for this is that during the downsampling operation too many samples, beyond the allowable threshold, are omitted and this resulted in aliasing in frequency domain and perfect reconstruction of the original signal is not possible anymore.

Question: During the downsampling operation we have to omit more samples than the number of allowable one. However, we want to decrease the effect of aliasing at the spectrum of the digital signal. What can we do for this?

Answer: If $y[n] = x[Mn]$ alising occurs in $Y_n(w)$, if the largest frequency of $X_n(w)$ in the interval $-\pi \le w < \pi$ is greater than $\frac{\pi}{M}$. This situation is depicted in Fig. 2.26.

For the conversion of $y[n]$ to continuous time signal $y_r(t)$, the portion of $Y_n(w)$ for the interval $-\pi \le w < \pi$ in Fig. 2.26 is used. This portion is depicted alone in Fig. 2.27.

As it is seen from Fig. 2.27, the overlapping shaded parts cause distortion in the reconstructed signal. Then how can we decrease the distortion amount? If we can

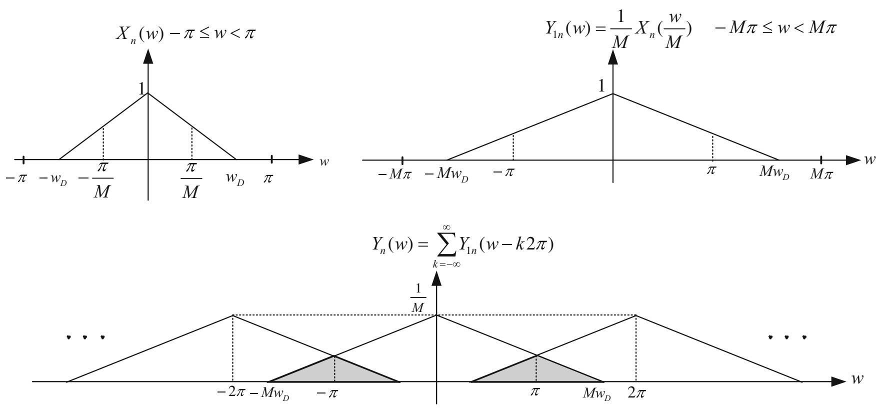
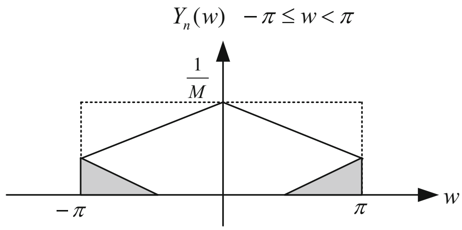


Fig. 2.26 Aliasing case in downsampled signal

Fig. 2.27 $Y_n(w),\ -\pi \le w < \pi$

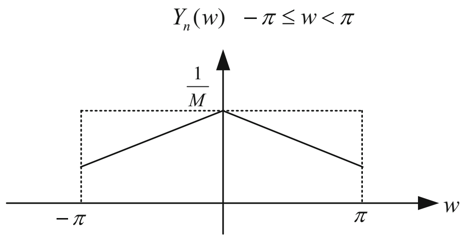


eliminate the shaded regions in the spectrum of the downsampled signal, the reconstructed signal will have less distortion.

However due to the clipping of the parts extending beyond the interval $(-\pi, \pi)$, some distortion will always be available in the reconstructed signal. This distortion is due to the information loss owing to the clipping of the spectrum regions in Fig. 2.26 for the intervals $\pi \le w < Mw_d$ and $-M\pi \le w < \pi$. What we do here is that we want try to decrease the amount of distortion, not complete elimination of it.

Then if we can get a spectrum graph for $Y_n(w),\ -\pi \le w < \pi$ as shown in Fig. 2.28 the reconstructed signal will have less distortion.

Fig. 2.28 After elimination of the overlapping shaded parts in Fig. 2.27



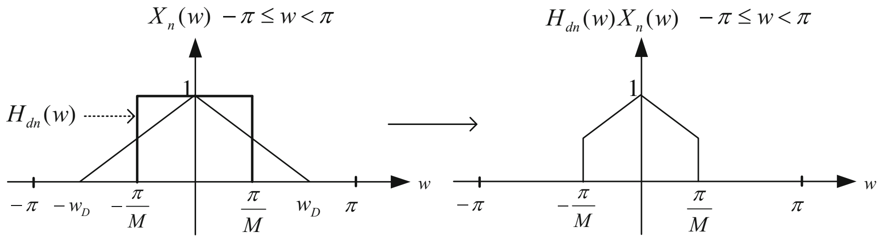


Fig. 2.29 Elimination of the high frequency parts by a decimator filter

We can omit the overlapping shaded parts if we can filter high frequency portions of $X_n(w)$ before downsampling operation, i.e., the portions of $X_n(w)$ for the intervals $\frac{\pi}{M} \le w < \pi$ and $-\pi \le w < -\frac{\pi}{M}$ should be filtered out. This can be achieved using a low pass filter as shown in bold lines Fig. 2.29. The lowpass filter clips the wigs of the signal that extends beyond the interval $(-\pi, \pi)$. And this clipping prevents the overlapping problem in downsampled signal spectrum.

The lowpass filter used in Fig. 2.29 is called decimator filter whose frequency domain expression for its one period around origin is written as

$$H_{dn}(w) = \begin{cases} 1 & if \quad |w| < \frac{\pi}{M} \\ 0 & if \quad \frac{\pi}{M} < |w| < \pi. \end{cases} \tag{2.26}$$

The time domain expression of the decimator filter can be computed using the inverse Fourier transform as

$$h_{dn}[n] = \frac{1}{2\pi} \int_{w,2\pi} H_{dn}(w) e^{jwn} dw \rightarrow h_{dn}[n] = \frac{1}{2\pi} \int_{-\frac{\pi}{M}}^{\frac{\pi}{M}} 1 \times e^{jwn} dw \tag{2.27}$$

yielding the expression

$$h_{dn}[n] = \frac{\sin\left(\frac{\pi n}{M}\right)}{\pi n} \rightarrow h_{dn}[n] = \frac{1}{M} \sin c\left(\frac{n}{M}\right). \tag{2.28}$$

The filtering process before downsampling operation is illustrated in Fig. 2.30. The system in Fig. 2.30 is called **decimator system**, and the overall operation in Fig. 2.30 is named as **decimation**.

For the system in Fig. 2.30, we have $Y_{1n}(w) = H_{dn}(w)X_n(w)$ and $y[n] = y_1[Mn]$. One period of $Y_n(w)$ is written as $Y_n(w) = \frac{1}{M} Y_{1n}(\frac{w}{M})$, $-\pi \le w < \pi$. One period of $Y_n(w)$ is shown in Fig. 2.31.

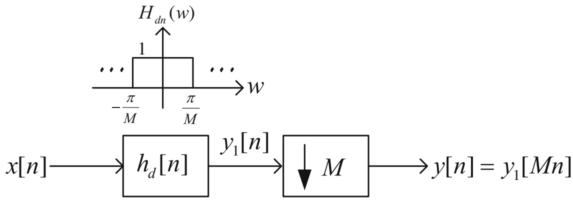


Fig. 2.30 Decimator system

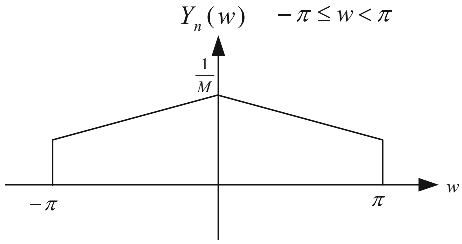


Fig. 2.31 One period of $Y_n(w)$

One period of $Y_n(w)$ can be expressed as

$$Y_{nop}(w) = \begin{cases} Y_n(w) & -\pi \leq w < \pi \\ 0 & otherwise \end{cases} \tag{2.29}$$

which can be used for the calculation of the Fourier transform of $y[n]$ as

$$Y_n(w) = \sum_{k=-\infty}^{\infty} Y_{nop}(w - k2\pi). \tag{2.30}$$

Considering Fig. 2.31 the graph of (2.30) can be drawn as in Fig. 2.32.

Exercise: If $y[n] = x[3n]$ and the Fourier transform of $x[n]$ for $-\pi \leq w < \pi$ is as given in Fig. 2.33, draw the Fourier transform of $y[n]$, i.e., draw $Y_n(w)$.

Downsampling can also be used for de-multiplexing operations, i.e., separating digital data to its components. We below give some examples to illustrate the use of downsampling for de-multiplexing operations.

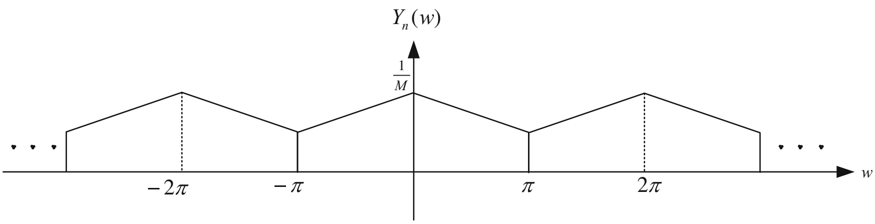


Fig. 2.32 Fourier transform of filtered and dowsampled signal

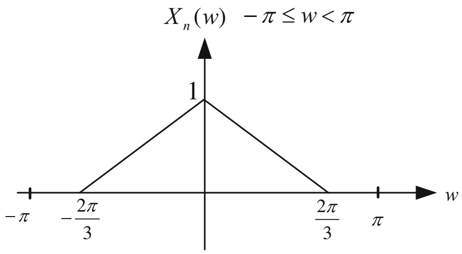


Fig. 2.33 One period of the Fourier transform of a digital signal

Note: The simplest de-multiplexer is the serial to parallel converter.

Example 2.9 The delay system is described in Fig. 2.34.

If

$$x[n] = [1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad \underbrace{9}_{n=0} \quad 10 \quad 11 \quad 12 \quad 13 \quad 14 \quad 15]$$

find the output of each unit given in Fig. 2.35.

Solution 2.9 To get $y[n] = x[n - n_0], n_0 > 0$, it is sufficient to shift $n = 0$ pointer to the left by n_0 units in $x[n]$ sequence. For negative n_0, we shift the $n = 0$ pointer to the right by n_0 units. According to this information, $x[n - 1]$ can be calculated as

$$x[n-1] = [1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad \underbrace{8}_{n=0} \quad 9 \quad 10 \quad 11 \quad 12 \quad 13 \quad 14 \quad 15].$$

If we divide the time axis by 2 and take only the integer division results, we get the signals

$$y_1[n] = [1 \quad 3 \quad 5 \quad 7 \quad \underbrace{9}_{n=0} \quad 11 \quad 13 \quad 15] \quad y_2[n] = [2 \quad 4 \quad 6 \quad \underbrace{8}_{n=0} \quad 10 \quad 12 \quad 14]$$

at the outputs of the downsamplers.

As it is seen from the obtained sequences, the system separates the odd and even indexed samples.

Fig. 2.34 Delay system

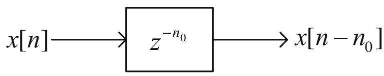
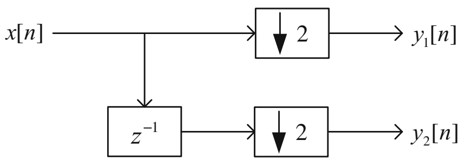


Fig. 2.35 Signal processing system for Example 2.9



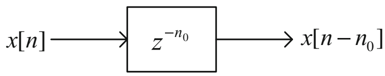


Fig. 2.36 Delay system

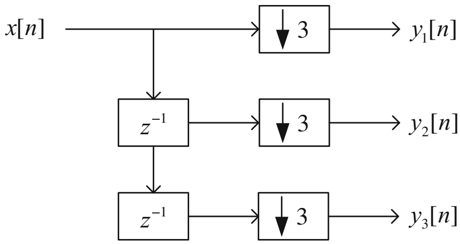


Fig. 2.37 Signal processing system for Example 2.10

Example 2.10 The delay system is shown in Fig. 2.36.

If

$$x[n] = [1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad \underbrace{9}_{n=0} \quad 10 \quad 11 \quad 12 \quad 13 \quad 14 \quad 15]$$

find the output of each unit given in Fig. 2.37.

Solution 2.10 Following similar steps as in the previous example, we find the digital signals at the outputs of the downsamplers as

$$y_1[n] = [3 \quad 6 \quad 9 \quad 12 \quad 15] \quad y_2[n] = [2 \quad 5 \quad 8 \quad 11 \quad 14]$$
$$y_3[n] = [1 \quad 4 \quad 7 \quad 10 \quad 13]$$

which are nothing but sub-sequences obtained by dividing data signal $x[n]$ into non-overlapping sequences.

2.1.5 Drawing the Fourier Transform of Downsampled Signal in Case of Aliasing (Practical Method)

Let $y[n] = x[Mn]$ be the downsampled digital signal. To draw the Fourier transform of $y[n]$ in case of aliasing, we follow the subsequent steps.

Step 1: First we draw the graph of $\frac{1}{M}X_n\left(\frac{w}{M}\right)$. For this purpose, we divide the horizontal axis of the graph of $X_n(w)$ by $\frac{1}{M}$, i.e., we multiply the horizontal axis by M, and multiply the amplitude values by $1/M$.
Step 2: In case of aliasing, the graph of $\frac{1}{M}X_n\left(\frac{w}{M}\right)$ extends beyond the interval $(-\pi, \pi)$. The portion of the graph extending to the left of $-\pi$ is denoted by 'A', and the potion extending to the right of π is denoted by 'B'.

Step 3: The portion of the graph denoted by 'A' in Step 2 is shifted to the right by 2π, and the portion denoted by 'B' is shifted to the left by 2π. The overlapping lines are summed and one period of $Y_n(w)$ around origin is obtained. Let's denote this one period by $Y_{n1}(w)$.
Step 4: In the last step, one period of $Y_n(w)$ around origin denoted by $Y_{n1}(w)$ is shifted to the left and right by multiples of 2π and all the shifted replicas are summed to get $Y_n(w)$, this is mathematically stated as

$$Y_n(w) = \sum_{k=-\infty}^{\infty} Y_{n1}(w - k2\pi).$$

Now let's explain these steps using graphics.

Let the Fourier transform of $x[n]$ be as shown in Fig. 2.38.

In case of aliasing, one period of $\frac{1}{M}X_n\left(\frac{w}{M}\right)$ around origin will be as shown in Fig. 2.39.

If Fig. 2.39 is inspected carefully, it is seen that the function $\frac{1}{M}X_n\left(\frac{w}{M}\right)$ takes values outside the interval $(-\pi, \pi)$ on horizontal axis. In Fig. 2.40, the shadowed triangles denoted by 'A' and 'B' show the portion of $\frac{1}{M}X_n\left(\frac{w}{M}\right)$ extending outside of $(-\pi, \pi)$.

Fig. 2.38 Fourier transform of $x[n]$

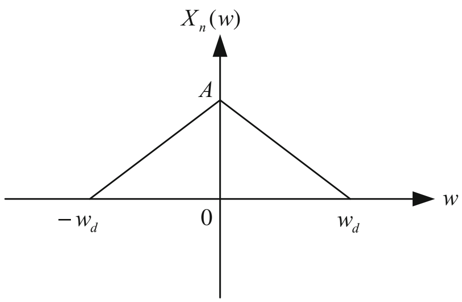
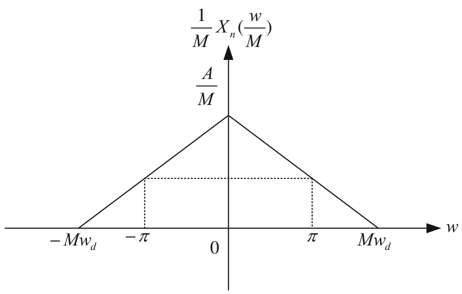


Fig. 2.39 One period of $\frac{1}{M}X_n\left(\frac{w}{M}\right)$ around origin in case of aliasing



If the shadowed triangles 'A' and 'B' in Fig. 2.40 are shifted to the right and left by 2π, we obtain the graphic in Fig. 2.41. If the overlapping lines in Fig. 2.41 are summed, we obtain the graphic shown in bold lines in Fig. 2.42. As it is clear from Fig. 2.41, overlapping regions distorts the original signal. The amount of distortion depends on the widths of the shadowed triangles. In other words, as the function $\frac{1}{M}X_n\left(\frac{w}{M}\right)$ extends outside the interval $(-\pi, \pi)$ more, the amount of distortion on the original signal due to overlapping increases.

The graph obtained after summing the overlapping lines is depicted alone in Fig. 2.43.

Fig. 2.40 One period of $\frac{1}{M}X_n\left(\frac{w}{M}\right)$ around origin in case of aliasing

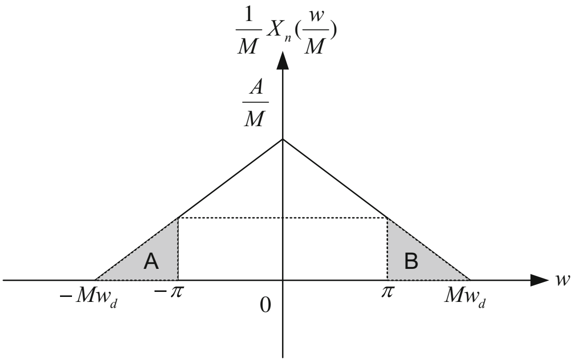


Fig. 2.41 Shaded parts shifted

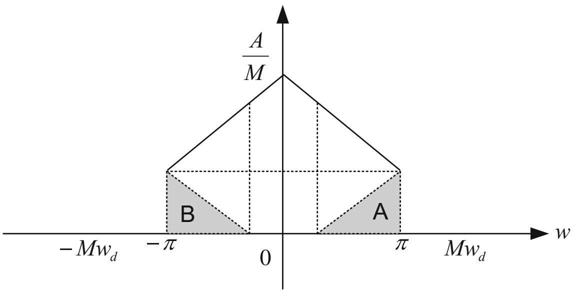


Fig. 2.42 Sum of the overlapping lines

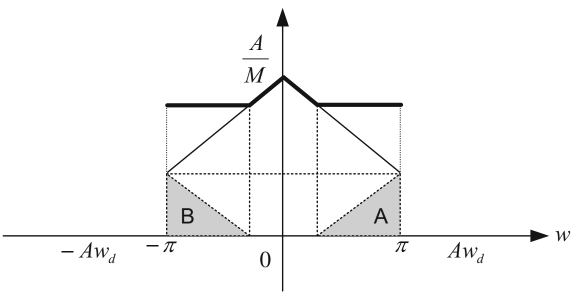


Fig. 2.43 The resulting graph after summing the overlapping lines

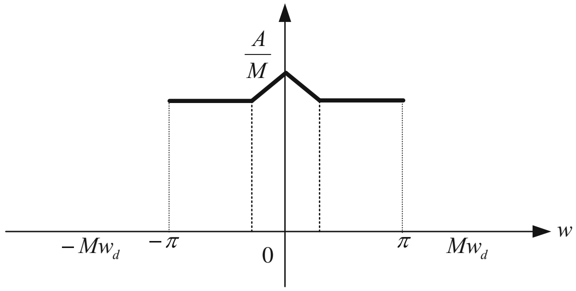
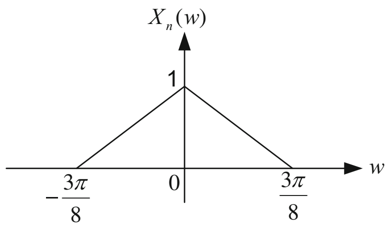


Fig. 2.44 One period of $X_n(w)$



Exercise 2.11 The Fourier transform of $x[n]$, i.e., $X_n(w)$, is shown in Fig. 2.44. Draw the Fourier transform of the downsampled signal $y[n] = x[Mn],\ M = 4$.

Solution 2.11

Step 1: First we draw the graph of $\frac{1}{M}X_n\left(\frac{w}{M}\right)$ as in Fig. 2.45.

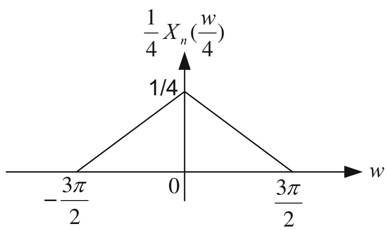
For the graph of Fig. 2.45, the parts that fall outside of the interval $(-\pi, \pi)$ are denoted by the shaded triangles 'A' and 'B' in Fig. 2.46.

If the shaded parts 'A' and 'B' in Fig. 2.46 are shifted to the right and to the left by 2π, we obtain the graph in Fig. 2.47.

The equations of the overlapping line on the interval $(-\pi, -\pi/2)$ in Fig. 2.47 can be written as $\frac{1}{12\pi}w + \frac{1}{4}$ and $-\frac{1}{12\pi}w - \frac{1}{24}$, and when these equations are summed, we obtain $\frac{5}{24}$. In a similar manner, the sum of the equations of the overlapping line on the interval $(\pi/2, \pi)$ can be found as $\frac{5}{24}$. Hence one period of $Y_n(w)$ around origin can be drawn as shown in Fig. 2.48.

In the last step, shifting one period of $Y_n(w)$ to the left and right by multiples of 2π and summing the shifted replicas we obtain the graph of $Y_n(w)$.

Fig. 2.45 One period of $\frac{1}{M}X_n\left(\frac{w}{M}\right)$



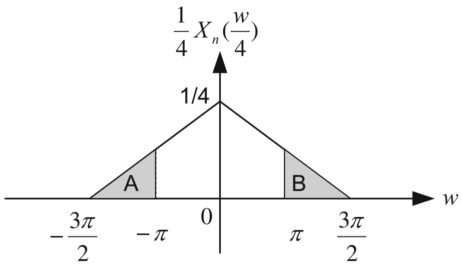


Fig. 2.46 One period of $\frac{1}{M}X_n\left(\frac{w}{M}\right)$

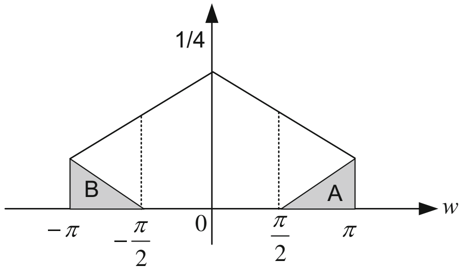


Fig. 2.47 *Shaded parts* shifted to the right and to the left by 2π

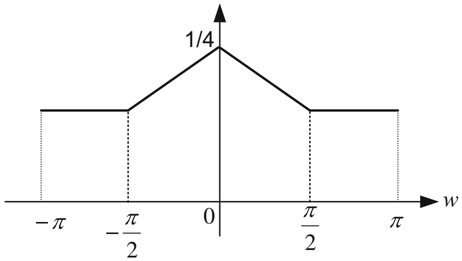


Fig. 2.48 One period of $Y_n(w)$ around origin

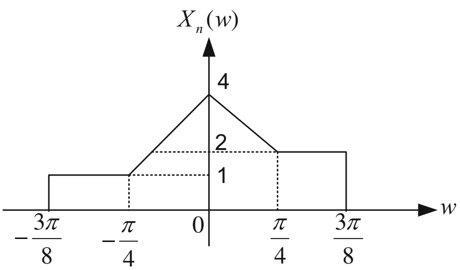
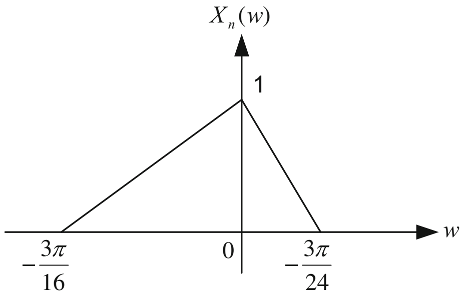


Fig. 2.49 One period of the Fourier transform of $x[n]$

Exercise: One period of the Fourier transform of $x[n]$ is shown in Fig. 2.49. Draw the Fourier transform of the downsampled signal $y[n] = x[4n]$.
Exercise: One period of the Fourier transform of $x[n]$ is shown in Fig. 2.50. Draw the Fourier transform of the downsampled signal $y[n] = x[8n]$.

Fig. 2.50 $X_n(w)$ bir periyodu



2.2 Upsampling: Increasing the Sampling Rate by an Integer Factor

Assume that we want to transmit an analog signal. For this purpose, we first take some samples from the continuous time signal and form a mathematical sequence, and this process is called sampling. To decrease the transmission overhead, we omit some of the digital samples and this process is called downsampling. After downsampling operation, we transmit the remaining samples. At the receiver side, for better reconstruction of the analog signal, we try to find a method to increase the number of digital samples. For this purpose, we try to find the samples omitted during the downsampling operation. After finding the omitted samples, we can reconstruct the analog signal in a better manner.

This means that first we reconstruct the original digital signal from downsampled digital signal then by using the reconstructed digital signal, we reconstruct the continuous time signal.

Reconstruction of the original digital signal from the downsampled signal includes a two-step process. The first step is called up sampling also named as signal-expansion. In this step, the compressed signal, i.e., downsampled signal, is expanded in time axis, and for the new time instants, 0 values are assigned for the new amplitudes. The second step is called interpolation which is the reconstruction part for the omitted digital samples. In this part, the 0 values assigned to new time amplitudes for the expanded signal are replaced by the estimated values.

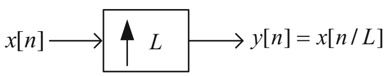
Now let's explain the upsampling operation.

2.2.1 Upsampling (Expansion)

The block diagram of the upsampler (expander) is shown in Fig. 2.51.

The mathematical expression of the upsampling operation is

Fig. 2.51 Upsampling operation



$$y[n] = \begin{cases} x\left[\frac{n}{L}\right] & n = 0, \pm L, \pm 2L, \ldots \\ 0 & otherwise. \end{cases} \tag{2.31}$$

For simplicity of the expression we will assume that for the new time indices in the expanded signal, the amplitude values are 0, so we will not always explicitly write the second condition in (2.31), i.e., we will only use $y[n] = x\left[\frac{n}{L}\right]$ to describe the signal expansion.

To draw the graph of $y[n] = x\left[\frac{n}{L}\right]$, or to obtain the expanded signal, $y[n] = x\left[\frac{n}{L}\right]$ we divide the time axis of $x[n]$ by $1/L$, i.e., we multiply the time axis of $x[n]$ by L. This operation is illustrated with an example now.

Example 2.12 If $x[n] = [1 \quad 3 \quad 5 \quad 7 \quad 9 \quad \underbrace{11}_{n=0} \quad 13 \quad 15 \quad 17]$ find $y[n] = x\left[\frac{n}{3}\right]$.

Solution 2.12 The indices for amplitude values of $x[n]$ are explicitly written in

$$x[n] = [\underbrace{1}_{n=-5} \quad \underbrace{3}_{n=-4} \quad \underbrace{5}_{n=-3} \quad \underbrace{7}_{n=-2} \quad \underbrace{9}_{n=-1} \quad \underbrace{11}_{n=0} \quad \underbrace{13}_{n=1} \quad \underbrace{15}_{n=-2} \quad \underbrace{17}_{n=3}].$$

Dividing the indices of $x[n]$ by $1/3$, i.e., multiplying the indices by 3, we get the sequence

$$[\underbrace{1}_{n=-15} \quad \underbrace{3}_{n=-12} \quad \underbrace{5}_{n=-9} \quad \underbrace{7}_{n=-6} \quad \underbrace{9}_{n=-3} \quad \underbrace{11}_{n=0} \quad \underbrace{13}_{n=3} \quad \underbrace{15}_{n=6} \quad \underbrace{17}_{n=9}].$$

Inserting missing indices and inserting 0 for amplitudes of the missing indices, we obtain the signal $y[n]$ as

$$y[n] = [1 \quad 0 \quad 0 \quad 3 \quad 0 \quad 0 \quad 5 \quad 0 \quad 0 \quad 7 \quad 0 \quad 0 \quad 9 \quad 0 \quad 0 \quad \underbrace{11}_{n=0} \quad 0 \quad 0 \quad 13$$
$$0 \quad 0 \quad 15 \quad 0 \quad 0 \quad 17].$$

2.2.2 Mathematical Formulization of Upsampling

The upsampling, expansion, of $x[n]$ by L is defined as

$$y[n] = \begin{cases} x\left[\frac{n}{L}\right] & n = 0, \pm L, \pm 2L, \ldots \\ 0 & otherwise \end{cases} \tag{2.32}$$

which can be written in terms of impulse function as

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]. \tag{2.33}$$

When the summation in (2.33) is expanded, we obtain

$$y[n] = \cdots + x[-1]\delta[n+L] + x[0]\delta[n] + x[1]\delta[n-L] + \cdots$$

Note that to find $x\left[\frac{n}{L}\right]$, we simply insert $L-1$ zeros between two samples of $x[n]$, that is, if

$$x[n] = [a \quad b \quad c \quad d \quad e],$$

then to get $x\left[\frac{n}{4}\right]$ simply insert 3 zeros between every two samples of $x[n]$, and this operation yields

$$x\left[\frac{n}{4}\right] = [a \quad 0 \quad 0 \quad 0 \quad b \quad 0 \quad 0 \quad 0 \quad c \quad 0 \quad 0 \quad 0 \quad d \quad 0 \quad 0 \quad 0 \quad e].$$

2.2.3 Frequency Domain Analysis of Upsampling

Let's try to find the Fourier transform of

$$y[n] = \begin{cases} x\left[\frac{n}{L}\right] & n = 0, \pm L, \pm 2L, \ldots \\ 0 & otherwise. \end{cases} \tag{2.34}$$

For this purpose, let's start with the definition of the Fourier transform of $y[n]$

$$Y_n(w) = \sum_{n=-\infty}^{\infty} y[n]e^{-jwn} \tag{2.35}$$

where substituting $\sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]$ for $y[n]$, we get

$$Y_n(w) = \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]e^{-jwn} \tag{2.36}$$

in which changing the order of summation terms, we obtain

$$Y_n(w) = \sum_{k=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x[k]\delta[n-kL]e^{-jwn} \tag{2.37}$$

which can be rearranged as

$$Y_n(w) = \sum_{k=-\infty}^{\infty} x[k] \underbrace{\sum_{n=-\infty}^{\infty} \delta[n-kL]e^{-jwn}}_{e^{-jwkL}} \tag{2.38}$$

yielding the expression

$$Y_n(w) = \sum_{k=-\infty}^{\infty} x[k]e^{-jwkL}. \tag{2.39}$$

If (2.39) is compared to the Fourier transform of $x[n]$

$$X_n(w) = \sum_{n=-\infty}^{\infty} x[n]e^{-jwn} \tag{2.40}$$

it is seen that

$$Y_n(w) = X_n(Lw) \tag{2.41}$$

Referring to (2.41), it is understood that the graph of $Y_n(w)$ can be obtained by dividing the frequency axis of $X_n(w)$ by L. As it is clear from (2.41) that the spectrum of the upsampled signal gets compressed in frequency domain. In fact, if a signal is expanded in time domain, it is compressed in frequency domain, similarly, if a signal is compressed in time domain, its spectrum expands in frequency domain.

Example 2.13 One period of the Fourier transform of $x[n]$ around origin is given in Fig. 2.52. Draw one period of the Fourier transform of $y[n] = x\left[\frac{n}{L}\right]$.

Solution 2.13 Dividing the frequency axis of $X_n(w)$ by L, we obtain the Fourier transform of $y[n]$ which is depicted in Fig. 2.53.

Note: Don't forget that the Fourier transforms $X_n(w)$ and $Y_n(w)$ are periodic functions with common period 2π. In fact, the Fourier transform of any digital signal is a periodic function with period 2π regardless whether the digital signal is periodic or not in time domain. If the digital signal is periodic in time domain then

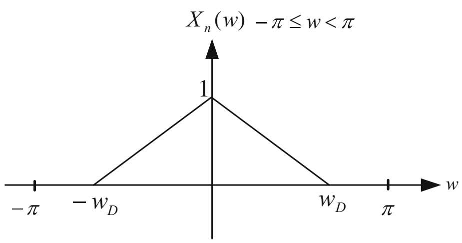


Fig. 2.52 One period of the Fourier transform of a digital signal

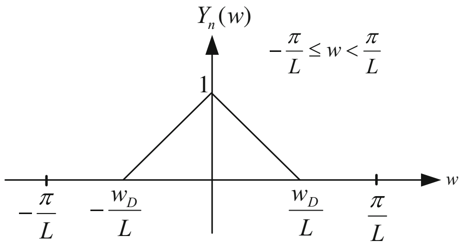


Fig. 2.53 One period of the Fourier transform of upsampled signal for Example 2.12

its Fourier transform is an impulse train with period 2π, i.e., its Fourier transform is a discrete signal.

Example 2.14 One period of the Fourier transform of $x[n]$ around origin is given in Fig. 2.54. Draw one period of the Fourier transform of $y[n] = x\left[\frac{n}{2}\right]$.

Solution 2.14 Dividing the frequency axis of $X_n(w)$ by 2, we get the graph in Fig. 2.55 for the Fourier transform of $y[n]$.

To get the graph in Fig. 2.55, we divided the horizontal axis of $X_n(w)$ by 2. Since $Y_n(w)$ is a periodic function with period 2π, the graph in Fig. 2.55 can also be drawn for the interval $-\pi \le w < \pi$ as shown in Fig. 2.56.

Example 2.15 For the system given in Fig. 2.44 $M = L = 2$, and

$$x[n] = [\underbrace{1}_{n=0} \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10].$$

Find the signals $x_d[n]$ and $y[n]$ in Fig. 2.57.

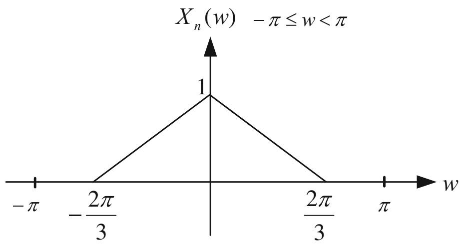


Fig. 2.54 One period of the Fourier transform of a digital signal

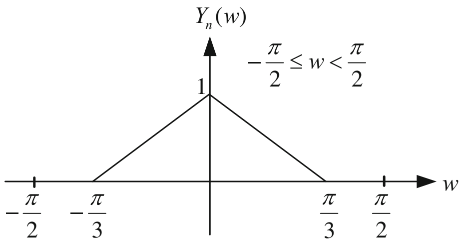


Fig. 2.55 One period of the Fourier transform of upsampled signal for Example 2.13

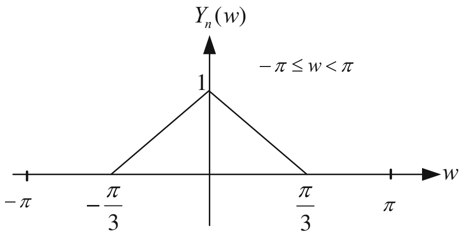


Fig. 2.56 One period of the Fourier transform of upsampled signal for Example 2.13

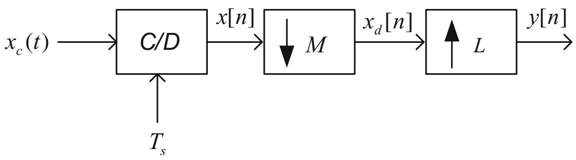


Fig. 2.57 Signal processing system for Example 2.14

Solution 2.15 To find $x_d[n]$, we divide the time indices of $x[n]$ by 2 and keep only integer division results. This operation yields

$$x_d[n] = [\underbrace{1}_{n=0} \quad 3 \quad 5 \quad 7 \quad 9].$$

To find $y[n]$, we divide the time indices of $x_d[n]$ by $\frac{1}{2}$, i.e., multiply the time indices of $x_d[n]$ by 2. For new indices, amplitude values are equated to 0. The result of this operation is the signal

$$y[n] = [\underbrace{1 \quad 0}_{n=0} \quad 3 \quad 0 \quad 5 \quad 0 \quad 7 \quad 0 \quad 9].$$

The overall procedure is illustrated in Fig. 2.58.

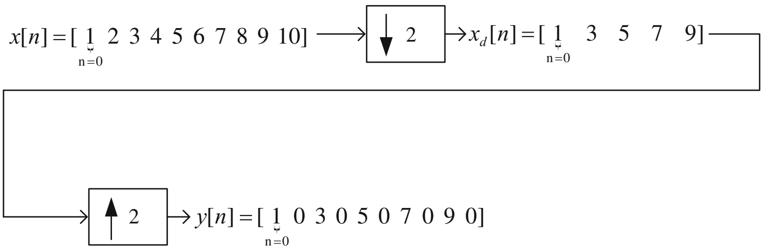


Fig. 2.58 Downsampling and upsampling

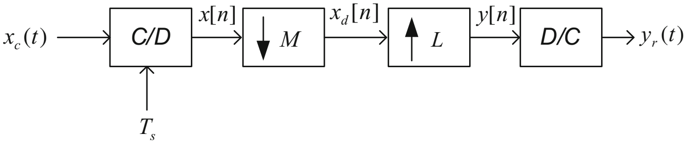


Fig. 2.59 Signal processing system

2.2.4 Interpolation

Let's consider the signal processing system shown in Fig. 2.59. The system includes one downsampler, one upsampler and one D/C converter. Let's study the reconstructed signal $y_r(t)$.

Assume that $y[n]$ is a causal signal. The signal $y_r(t)$ is calculated from the digital signal $y[n]$ using

$$y_r(t) = \sum_{n=-\infty}^{\infty} y[n]h_r(t - nT_s) \tag{2.42}$$

where $h_r(t)$ can either be ideal reconstuction filter, i.e., $h_r(t) = sinc(t/T_s)$ or triangular approximated reconstruction filter, or any other approximated filter. When we expand the summation in (2.42), we see that some of the shifted filters are multiplied by 0, since some of the samples of $y[n]$ are 0. The expansion of (2.42) happens to be as

$$y_r(t) = y[0]h_r(t) + y[1]h_r(t - T_s) + y[2]h_r(t - 2T_s) + y[3]h_r(t - 3T_s) + \cdots \tag{2.43}$$

yielding

$$y_r(t) = 1 \times h_r(t) + 0 \times h_r(t - T_s) + 3 \times h_r(t - 2T_s) + 0 \times h_r(t - 3T_s) + \cdots \tag{2.44}$$

Multiplication of some of the shifted filters by 0 results in information loss in the reconstructed signal.

Question: So how can we increase the quality of the reconstructed signal?

Answer: If we can replace 0 values in the expanded signal $y[n]$ by their estimated values, $y_r(t)$ expression in (2.44) will not include 0 multiplication terms and reconstructed signal becomes better. That is,

$$x[n] = [\underbrace{1}_{n=0}\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10]$$

$$y[n] = [\underbrace{1}_{n=0}\ 0\ 3\ 0\ 5\ 0\ 7\ 0\ 9\ 0]$$

Replace 0's by the estimated values of the omitted samples

Omitted samples are 2, 4, 6, 8, 10

So how can we find a method to find approximate values for the omitted samples of original signal $x[n]$? If we can approximate omitted samples, we can replace 0's in the expanded signal by the approximated values, then reconstruct the continuous time signal. The quality of the reconstructed signal will be better.

We know that the amplitude values of a continuous time signal at time instants t_i and t_{i+1} does not change sharply. Otherwise, it violates the definition of continuous time signal. For instance, the amplitude values of a continuous time signal for three time instants are given in Fig. 2.60.

Hence for the omitted samples, we can make a linear estimation. Assume that $L = M = 2$, in this case, during the downsampling operation; we omit one sample from every other 2 samples. After upsampling operation, we have 0 in the place of omitted sample. We can estimate the omitted sample using the neighbor samples of the omitted sample.

In Fig. 2.60, assume that after sampling operation, we obtain the digital signal [a b c], and in this case, downsampled signal can be calculated as $[a \quad c]$. The expanded signal or upsampled signal becomes as $[\mathrm{a} \quad 0 \quad \mathrm{c}]$ where 0 can be replaced by the estimated value $\frac{a+c}{2}$. In general if there are $L-1$ zeros between two samples of the expanded signal, we can estimate the omitted samples drawing a line between the amplitudes of these two samples as illustrated in Fig. 2.61.

The missing samples in Fig. 2.61. can be calculated using

$$y[n_i] = b + \frac{a-b}{L}(n_{k+L-1} - n_i), \quad i = k : k + L - 1. \tag{2.45}$$

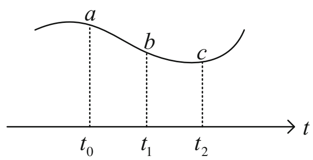


Fig. 2.60 Amplitude values of a continuous time signal for three distinct time instants

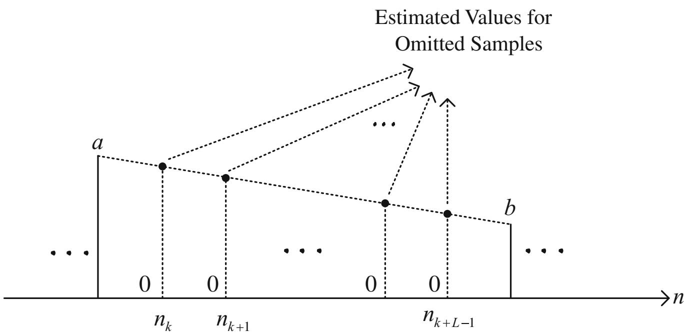


Fig. 2.61 Linear estimation of the missing samples

Let $\Delta = \frac{a-b}{L}$, when (2.45) is expanded for $i = k : k + L - 1$, we get the amplitude vector

$$[b + (\mathrm{L}-1)\Delta \quad b + (\mathrm{L}-2)\Delta \cdots b + 2\Delta \quad b + \Delta]. \tag{2.46}$$

Example 2.16 Let $x[n] = [\underbrace{1}_{n=0} \quad 2 \quad 5 \quad 7 \quad 9 \quad 10 \quad 10]$ find the signals $x_d[n] = x[3n]$ $y[n] = x_d\left[\frac{n}{3}\right]$ and using linear estimation method, estimate the missing samples in $y[n]$.

Solution 2.16 To calculate the downsampled signal, we divide the time axis of $x[n]$ by 3 and keep only integer division results, and in a similar manner, to calculate the upsampled signal, we multiply the time axis of the downsampled signal by 3, and for the new time instants 0's are assigned for amplitude values. The downsampled and upsampled signals can be calculated as

$$x_d[n] = [\underbrace{1}_{n=0} \quad 7 \quad 10] \quad y[n] = [\underbrace{1}_{n=0} \quad 0 \quad 0 \quad 7 \quad 0 \quad 0 \quad 10].$$

and these signals are graphically shown in Fig. 2.62.

The missing samples in upsampled signal can be calculated using

$$\Delta = \frac{a-b}{L}, \quad \text{and} \quad [b + (\mathrm{L}-1)\Delta \qquad b + (\mathrm{L}-2)\Delta \quad \cdots \quad b + 2\Delta \quad b + \Delta]$$

For the first 2 missing samples

$$\Delta = \frac{1-7}{3} \rightarrow \Delta = -2$$

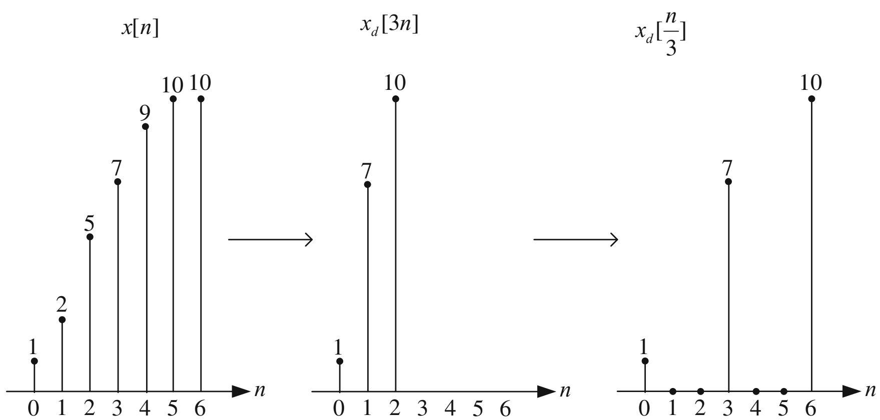


Fig. 2.62 Original signal, downsampled signal, upsampled signal

and the missing samples are

$$[7+2(-2) \quad 7+1(-2)] \rightarrow [3 \quad 4].$$

For the next 2 missing samples

$$\Delta = \frac{7-10}{3} \rightarrow \Delta = -1$$

and the missing samples are

$$[10\ +\ 2(-1) \quad 10+1(-1)] \rightarrow [8 \quad 9].$$

The calculation of the missing samples is graphically illustrated in Fig. 2.63. Hence with the estimated values, the upsampled signal becomes as

$$y[n] = [\underbrace{1}_{n=0} \quad 3 \quad 4 \quad 7 \quad 8 \quad 9 \quad 10]. \tag{2.47}$$

The original sequence before downsampling operation was

$$x[n] = [\underbrace{1}_{n=0} \quad 2 \quad 5 \quad 7 \quad 9 \quad 10 \quad 10]. \tag{2.48}$$

When (2.47) is compared to (2.48), we see that the calculated samples are close to the original omitted samples.

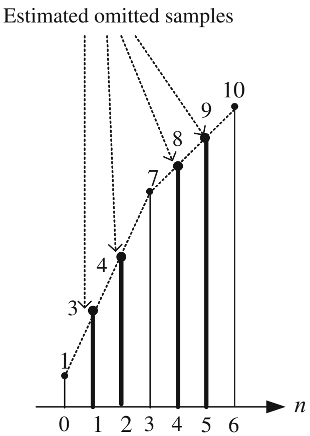


Fig. 2.63 Calculation of the missing samples

2.2.5 Mathematical Analysis of Interpolation

We explained an estimation method for the calculation of missing samples in expanded signal. However, we did not follow a mathematical analysis. How can we find the missing samples in upsampled (expanded) signal using a mathematical approach?

In time domain, it is difficult to find a mathematical approach for the estimation of missing samples. Let's approach to the problem in frequency domain. Let's consider the system involving downsampling and upsampling operations given in Fig. 2.64 where we assume that $L = M$.

Let's assume that the Fourier transform of $x[n]$ is as in Fig. 2.65. We will inspect the Fourier transforms of $y[n]$ and $x[n]$ in Fig. 2.64 and find a relation between them.

Considering Fig. 2.65 the Fourier transform of $x_d[Mn]$ can be drawn as in Fig. 2.66.

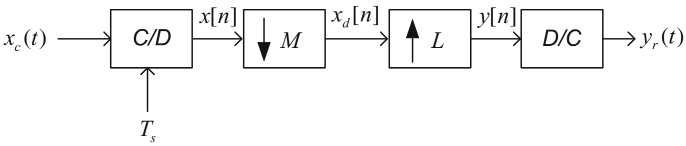


Fig. 2.64 Signal processing system including upsampling and downsampling operations

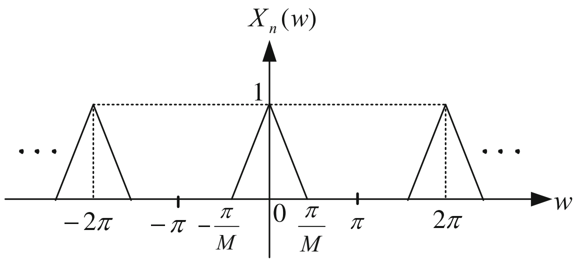


Fig. 2.65 Fourier transform of a digital signal

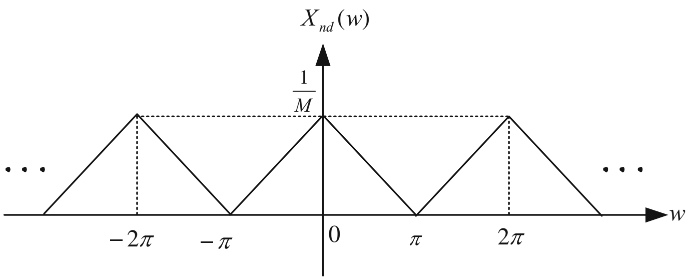


Fig. 2.66 Fourier transform of $x_d[Mn]$

Dividing the horizontal axis of the graph in Fig. 2.66 by L, we obtain the graph of $Y_n(w)$ as Fig. 2.67.

If we compare the graph of $X_n(w)$ in Fig. 2.65 to the graph of $Y_n(w)$ in Fig. 2.67, it is seen that for $\frac{\pi}{L} \leq |w| < 2\pi - \frac{\pi}{L}$ $X_n(w) = 0$ but $Y_n(w) \neq 0$, and for other frequency intervals, $Y_n(w) = \frac{1}{M} X_n(w)$. This is illustrated in Fig. 2.68.

How can we make $Y_n(w)$ to be equal to $X_n(w)$ for all frequency values? This is possible if we multiply $Y_n(w)$ by a lowpass digital filter with the transfer function as in Fig. 2.69.

Since $L = M$ and $Y_i(w) = H_i(w) Y_n(w)$, we can show the multiplication of $H_i(w) Y_n(w)$ as in Fig. 2.70.

The result of the above multiplication is depicted in Fig. 2.71.

For $L = M$, we have $Y_i(w) = X_n(w)$ which means that $y_i[n] = x[n]$, that is omitted samples are reconstructed perfectly.

Let's now do the time domain analysis of this reconstruction process. If $Y_i(w) = H_i(w) Y_n(w)$, then $y_i[n] = h_i[n] * y[n]$. The time domain expression $h_i[n]$ can be obtained via inverse Fourier transform

$$h_i[n] = \frac{1}{2\pi} \int_{2\pi} H_i(w) e^{jwn} dw \tag{2.49}$$

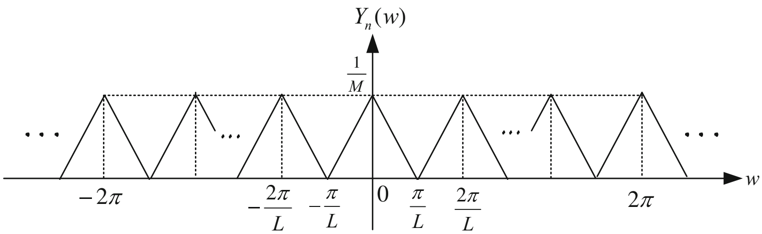


Fig. 2.67 Fourier transform of the signal $y[n]$ in Fig. 2.64

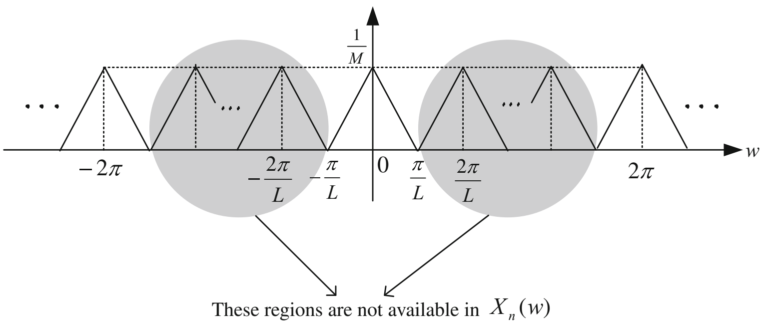


Fig. 2.68 Comparison of $X_n(w)$ and $Y_n(w)$

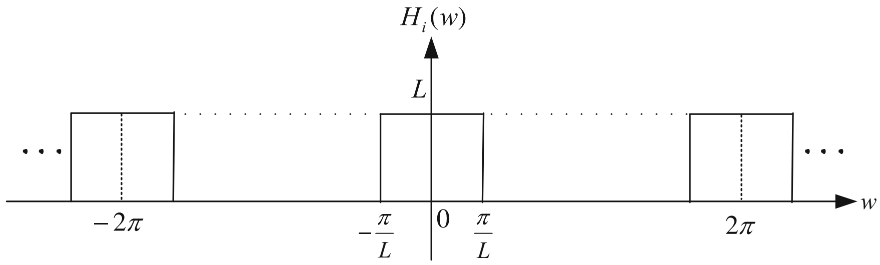


Fig. 2.69 Lowpass digital filter

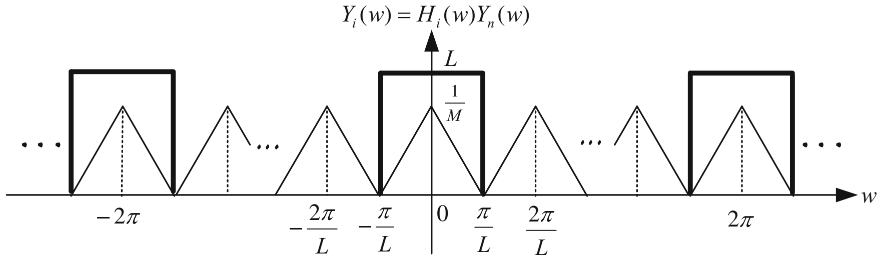


Fig. 2.70 The multiplication of $H_i(w)Y_n(w)$

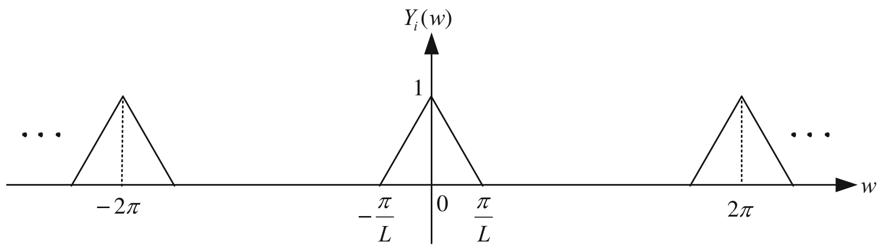


Fig. 2.71 The graph of $Y_i(w) = H_i(w)Y_n(w)$

where using the frontiers $-\frac{\pi}{L}, \frac{\pi}{L}$, we get

$$h_i[n] = \frac{1}{2\pi} \int_{-\frac{\pi}{L}}^{\frac{\pi}{L}} Le^{jwn} dw \rightarrow h_i[n] = \frac{\sin\left(\frac{\pi n}{L}\right)}{\frac{\pi n}{L}} \tag{2.50}$$

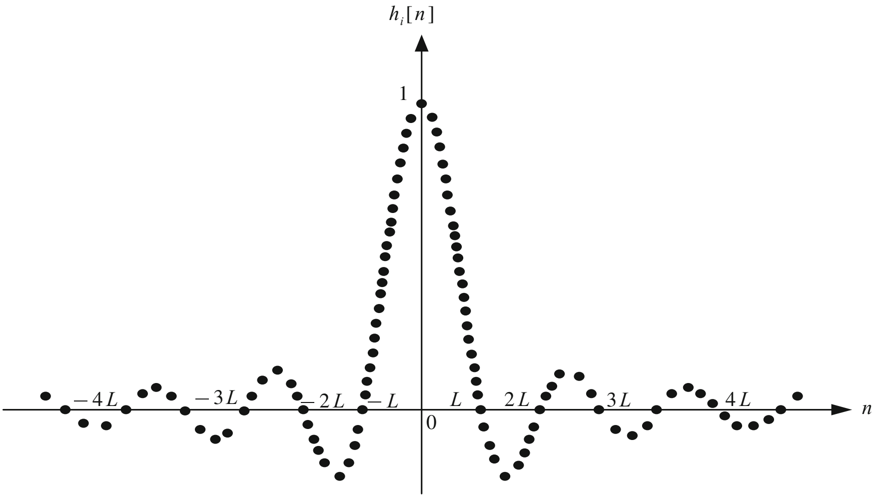


Fig. 2.72 The graph of $\sin c(n/L)$

which can be expressed in terms of $\sin c(\cdot)$ function as

$$h_i[n] = \sin c\left(\frac{n}{L}\right). \tag{2.51}$$

The graph of $\sin c(n/L)$ is depicted in Fig. 2.72.

As it is seen from Fig. 2.72 that $h_i[n] = \sin c\left(\frac{n}{L}\right)$ equals to 0 when n is a multiple of L. The digital filter with impulse response $h_i[n] = \sin c\left(\frac{n}{L}\right)$ is called interpolating filter which is used to reconstruct those digital samples omitted during downsampling operation, i.e., used to reconstruct missing samples in the expanded, or upsampled signal.

Exercise: The continuous time signal $x_c(t) = \cos(2\pi t)$ is sampled with sampling period $T_s = 1/8$ s.

(a) For a mathematical sequence $x[n]$ from the samples taken from continuous time signal in the interval 0–1 s.
(b) $x[n]$ is downsampled by $M = 2$, and $x_d[n]$ is the downsampled signal, find $x_d[n]$.
(c) The downsampled signal $x_d[n]$ is upsampled and let $y[n]$ be the upsampled signal, find $y[n]$.
(d) Calculate the missing samples in $y[n]$ using the ideal interpolation filter.

2.2.6 Approximation of the Ideal Interpolation Filter

Since digital $\sin c(\cdot)$ filter is an ideal filter, it is difficult to implement such filters, instead we can use an approximation of this digital filter. As it is clear from Fig. 2.72, the digital $\sin c(\cdot)$ filter includes a large main lobe centered upon origin, and many other side lobes. To approximate the digital $\sin c(\cdot)$ filter, we can use triangles for the lobes in Fig. 2.72. The simplest approximation is to use an isosceles triangle for the main lobe and omit the other side lobes.

The simplest approximated digital can filter can be obtained as shown in Fig. 2.73.

Referring to Fig. 2.73 the approximated interpolation filter can mathematically be expressed as

$$h_{ai}[n] = \begin{cases} \frac{n}{L}+1, & \mathit{if} \quad -L \le n < 0 \\ -\frac{n}{L}+1, & \mathit{if} \quad 0 \le n < L \\ 0, & \mathit{otherwise} \end{cases} \tag{2.52}$$

which can be expressed in more compact form as

$$h_{ai}[n] = \begin{cases} -\frac{|n|}{L}+1, & \mathit{if} \quad -L \le n < L \\ 0, & \mathit{otherwise}. \end{cases} \tag{2.53}$$

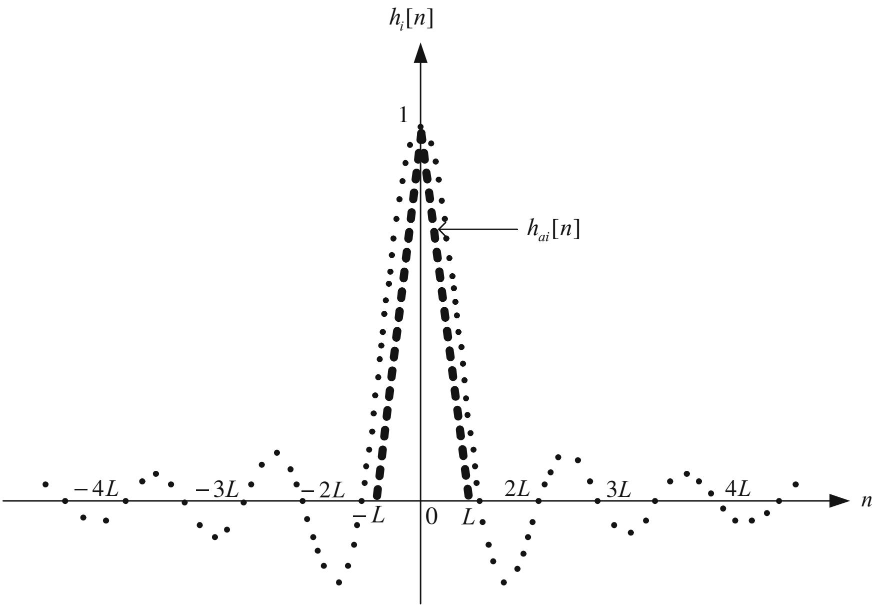


Fig. 2.73 Approximation of the ideal interpolation filter

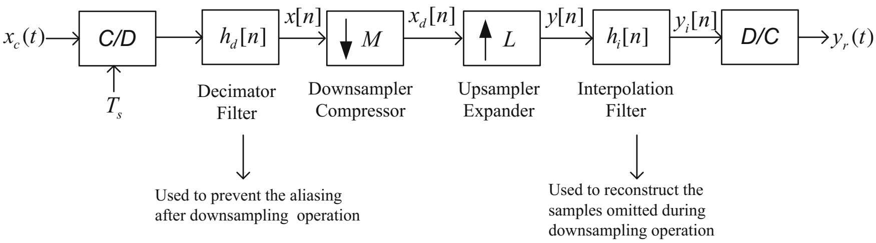


Fig. 2.74 Signal processing system with interpolation filter

With the interpolation filter our complete signal processing system becomes as in Fig. 2.74.

For the reconstruction of the samples omitted during downsampling operation, if approximated interpolating filter is used, the reconstructed digital signal can be written as

$$y_i[n] = h_{ai}[n] * y[n] \rightarrow y_i[n] = \sum_{k=-\infty}^{\infty} y[k]h_{ai}[n-k] \tag{2.54}$$

where $h_{ai}[n]$ denotes the approximated reconstruction filter, or interpolation filter. Now let's try to write a relation between $x_d[n]$ and $y_i[n]$ given in Fig. 2.74. We know that

$$y[n] = \sum_{k=-\infty}^{\infty} x_d[k]\delta[n-kL]. \tag{2.55}$$

When (2.53) is replaced into

$$y_i[n] = h_i[n] * y[n] \tag{2.56}$$

we get

$$y_i[n] = h_i[n] * \sum_{k=-\infty}^{\infty} x_d[k]\delta[n-kL] \tag{2.57}$$

which is simplified as

$$y_i[n] = \sum_{k=-\infty}^{\infty} x_d[k]h_i[n-kL]. \tag{2.58}$$

When (2.58) is expanded, we get the explicit form of $y_i[n]$ as

$$y_i[n] = \cdots + x_d[-1]h_i[n+L] + x_d[0]h_i[n] + x_d[1]h_i[n-L] + \cdots \tag{2.59}$$

Using the ideal interpolation filter, i.e., ideal reconstruction filter,

$$h_i[n] = \frac{\sin\left(\frac{\pi n}{L}\right)}{\frac{\pi n}{L}}$$

in (2.58), we can write the reconstructed digital signal as

$$y_i[n] = \sum_{k=-\infty}^{\infty} x_d[k] \frac{\sin\left(\frac{\pi(n-kL)}{L}\right)}{\frac{\pi(n-kL)}{L}} \tag{2.60}$$

or in terms of $\sin c(\cdot)$ function, we can write (2.60) as

$$y_i[n] = \sum_{k=-\infty}^{\infty} x_d[k] \sin c\left(\frac{n-kL}{L}\right). \tag{2.61}$$

Note: Digital reconstructed signal expression $y_i[n] = \sum_{k=-\infty}^{\infty} x_d[k]h_i[n-kL]$ is quite similar to the analog reconstructed signal expression $x_r(t) = \sum_{k=-\infty}^{\infty} x[k] h_r(t-kT_s)$.

Example 2.17 For the system given in Fig. 2.75 $L = M = 3$ and $x[n] = [1 \quad 2 \quad 3 \quad 4]$. Find $x_d[n], y[n]$, and $y_i[n]$. Use approximated linear digital filter for $h_i[n]$.

Solution 2.17 For $L = M = 3$, if $x[n] = [1 \quad 2 \quad 3 \quad 4]$, then $x_d[n] = [1 \quad 4]$ and $y[n] = [1 \quad 0 \quad 0 \quad 4]$.

To find $y_i[n]$ we can use either

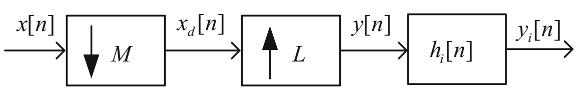
$$y_i[n] = \sum_{k=-\infty}^{\infty} y[k]h_{ai}[n-k] \tag{2.62}$$

or

$$y_i[n] = \sum_{k=-\infty}^{\infty} x_d[k]h_i[n-kL] \tag{2.63}$$

Let's use both of them separately. First using (2.53), let's calculate and draw the linear approximated digital interpolation filter as in Fig. 2.76.

Fig. 2.75 Signal processing system for Example 2.16



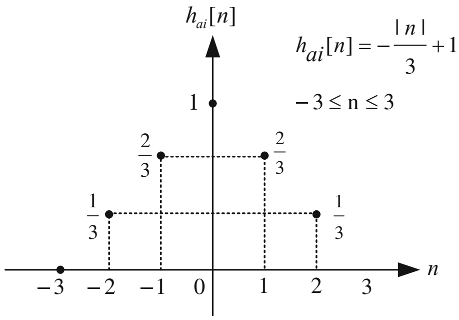


Fig. 2.76 Approximated interpolation filter

Expanding (2.62), we get

$$y_i[n] = y[0]h_{ai}[n] + y[1]h_{ai}[n-1] + y[2]h_{ai}[n-2] + y[3]h_{ai}[n-3]. \tag{2.64}$$

If $y[n] = [1 \quad 0 \quad 0 \quad 4]$ is considered, we see that the amplitude values at indices $n = 1$, and $n = 2$, are missing. When $n = 1$ is placed into (2.64), we get

$$y_i[1] = \underbrace{y[0]}_{1}\,\underbrace{h_{ai}[1]}_{2/3} + \underbrace{y[1]}_{0}\,\underbrace{h_{ai}[0]}_{1} + \underbrace{y[2]}_{0}\,\underbrace{h_{ai}[-1]}_{2/3} + \underbrace{y[3]}_{4}\,\underbrace{h_{ai}[-2]}_{1/3} \tag{2.65}$$

which yields

$$y_i[1] = \frac{2}{3} + \frac{4}{3} \to y_i[1] = 2 \tag{2.66}$$

and when $n = 2$ is placed into (2.64), we obtain

$$y_i[2] = \underbrace{y[0]}_{1}\,\underbrace{h_{ai}[2]}_{1/3} + \underbrace{y[1]}_{0}\,\underbrace{h_{ai}[1]}_{1} + \underbrace{y[2]}_{0}\,\underbrace{h_{ai}[0]}_{2/3} + \underbrace{y[3]}_{4}\,\underbrace{h_{ai}[-1]}_{2/3} \tag{2.67}$$

which yields

$$y_i[2] = \frac{1}{3} + \frac{8}{3} \to y_i[2] = 3 \tag{2.68}$$

So missing samples are found as $y_i[1] = 2$ and $y_i[2] = 3$, and when these samples are replaced by 0's in $y[n]$, we get

$$y_i[n] = [1 \quad 2 \quad 3 \quad 4]$$

Now let's use the formula

$$y_i[n] = \sum_{k=-\infty}^{\infty} x_d[k] h_i[n - kL]. \tag{2.69}$$

When (2.69) is expanded, noting that $x_d[n] = [1 \quad 4]$ and $L = 3$, we get

$$y_i[n] = x_d[0] h_{ai}[n] + x_d[1] h_{ai}[n-3]. \tag{2.70}$$

When (2.70) is evaluated for $n = 1$, we obtain

$$y_i[1] = \underbrace{x_d[0]}_{1} \underbrace{h_{ai}[1]}_{2/3} + \underbrace{x_d[1]}_{4} \underbrace{h_{ai}[-2]}_{1/3}$$

which yields

$$y_i[1] = \frac{2}{3} + \frac{4}{3} \to y_i[1] = 2 \tag{2.71}$$

and when (2.69) is evaluated for $n = 2$, we get

$$y_i[2] = \underbrace{x_d[0]}_{1} \underbrace{h_{ai}[2]}_{1/3} + \underbrace{x_d[1]}_{4} \underbrace{h_{ai}[-1]}_{2/3} \tag{2.72}$$

which yields

$$y_i[2] = \frac{1}{3} + \frac{8}{3} \to y_i[2] = 3. \tag{2.73}$$

Hence, both formulas give the same results. In addition, we had already introduced the linear estimation method using the continuity property of analog signals. It is now very clear that the linear estimation method is nothing but the use of triangle approximated digital reconstruction filter.

Example 2.18 Show that the systems given in Fig. 2.77 have the same outputs for the same inputs.

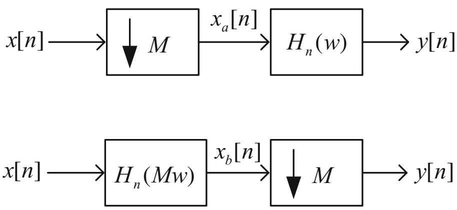


Fig. 2.77 Signal processing systems for Example 2.17

Solution 2.18 For the first system we have

$$X_{an}(w) = \frac{1}{M}\sum_{k=0}^{M-1} X_n\left(\frac{w-k2\pi}{M}\right) \tag{2.74}$$

and

$$Y_n(w) = H_n(w)X_{an}(w) \to Y_n(w) = \frac{H_n(w)}{M}\sum_{k=0}^{M-1} X_n\left(\frac{w-k2\pi}{M}\right) \tag{2.75}$$

For the second system we have

$$X_{bn}(w) = H_n(Mw)X_n(w) \tag{2.76}$$

and

$$Y_n(w) = \frac{1}{M}\sum_{k=0}^{M-1} X_{bn}\left(\frac{w-k2\pi}{M}\right). \tag{2.77}$$

When (2.76) is inserted into (2.77), we obtain

$$Y_n(w) = \frac{1}{M}\sum_{k=0}^{M-1} H_n\left(M\frac{w-k2\pi}{M}\right)X_n\left(\frac{w-k2\pi}{M}\right). \tag{2.78}$$

Since $H_n(w)$ is a periodic function with period 2π, (2.78) can be written as

$$Y_n(w) = \frac{1}{M}\sum_{k=0}^{M-1} H_n(w)X_n\left(\frac{w-k2\pi}{M}\right) \tag{2.79}$$

which is equal to

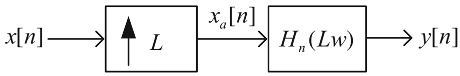
$$Y_n(w) = H_n(w)\frac{1}{M}\sum_{k=0}^{M-1} X_n\left(\frac{w-k2\pi}{M}\right) \to Y_n(w) = H_n(w)X_{an}(w). \tag{2.80}$$

When (2.75) is compared to (2.80), we see that both systems have the same outputs for the same inputs.

Exercise: Show that the systems given below have the same outputs for the same inputs (Fig. 2.78).

Example 2.19 For the system given in Fig. 2.79, find a relation in time domain between system input $x[n]$ and system output $y[n]$.

Fig. 2.78 Signal processing system for exercise



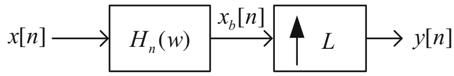
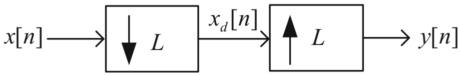


Fig. 2.79 Signal processing system for Example 2.18



Solution 2.19 We have $x_d[n] = x[Ln]$ and $y[n] = x_d\left[\frac{n}{L}\right]$. Putting $x_d[n]$ expression into $y[n]$ expression, we get $y[n] = x[\frac{Ln}{L}] \rightarrow y[n] = x[n]$. However, this is not always correct. Since we know that for $L = 2$ if $x[n] = [1 \quad 2 \quad 3]$, then $x_d[n] = [1 \quad 3]$ and $y[n] = [1 \quad 0 \quad 3]$, it is obvious that $x[n] \neq y[n]$.

But using $x_d[n] = x[Ln]$ and $y[n] = x_d\left[\frac{n}{L}\right]$, we found $y[n] = x[n]$. So, what is wrong with our approach to the problem?

Because, we did not pay attention to the criteria in upsampling operation. That is, $y[n] = x_d\left[\frac{n}{L}\right]$ if $n = kL, k \in Z$; otherwise, $y[n] = 0$. Then $y[n] = x[n]$ is valid only for some values of n and these n values are multiples of L. That is for $L = 2$ if $x[n] = [1 \quad 2 \quad 3]$, then $x_d[n] = [1 \quad 3]$ and $y[n] = [1 \quad 0 \quad 3]$, and $y[n] = x[n]$ for $n = 0, 2$ only.

However, for some signals, no information loss occurs after compression operation. This is possible if the omitted samples are also zeros. In this case, expanded signal equals to the original signal. For example, if

$$x[n] = [\underbrace{a}_{n=0} \quad 0 \quad b \quad 0 \quad c \quad 0 \quad d]$$

then after downsampling by $L = 2$, we get

$$x_d[n] = [a \quad b \quad c \quad d]$$

and after expansion by $L = 2$, we obtain

$$y[n] = [\underbrace{a}_{n=0} \quad 0 \quad b \quad 0 \quad c \quad 0 \quad d]$$

Thus, we see that $y[n] = x[n]$ for every n values.

To write a mathematical expression between $x[n]$ and $y[n]$, let's express $x_d[n]$ in terms of $x[n]$ as

$$x_d[n] = \sum_{n=-\infty}^{\infty} x[n] \sum_{r=-\infty}^{\infty} \delta[n - rM] \tag{2.81}$$

and express $y[n]$ in terms of $x_d[n]$ as

$$y[n] = \sum_{k=-\infty}^{\infty} x_d[k]\delta[n - kL]. \tag{2.82}$$

Inserting (2.81) into (2.82), we obtain

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] \sum_{r=-\infty}^{\infty} \delta[k - rM] \sum_{n=-\infty}^{\infty} \delta[n - kL] \tag{2.83}$$

which is the final expression showing the relation between $x[n]$ and $y[n]$.

Example 2.20 Find a method to check whether information loss occurs or not after downsampling by M.

Solution 2.20 If $x[n]$ is downsampled by M, we omit $M - 1$ samples from every M samples. If we denote the information bit indices by the numbers $0, 1, 2, \ldots, M \ldots$, then the first omitted samples have indices $1, 2, \ldots, M - 1$ and the second set of omitted indices have indices $M + 1, M + 2, \ldots, 2M - 1$, and so on.

Hence, by summing the absolute values of the omitted samples and checking whether it equals to zero or not, we can conclude whether information loss occurs or not after downsampling operation. That is, we calculate

$$Loss = \sum_{k=-\infty}^{\infty} \sum_{n=1}^{M-1} |x[n + kM]| \tag{2.84}$$

and if $Loss \neq 0$, then information loss occurs after downsampling of $x[n]$, otherwise not.

Example 2.21 If

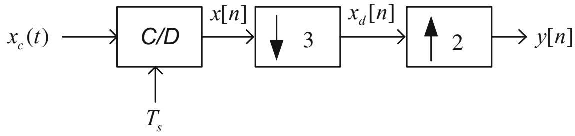
$$y[n] = \begin{cases} x[n] & \textit{if } n \textit{ is even} \\ 0 & \textit{otherwise} \end{cases} \tag{2.85}$$

then write a mathematical expression between $x[n]$ and $y[n]$.

Solution 2.21 Using (2.85), we can express $y[n]$ in terms of $x[n]$ as

$$y[n] = \frac{1 + (-1)^n}{2} x[n]. \tag{2.86}$$

Fig. 2.80 Signal processing system for Example 2.21



Since $\cos(\pi n) = (-1)^n$, then (2.86) can also be written as

$$y[n] = \frac{1 + \cos(\pi n)}{2} x[n].$$

Example 2.22 For the system given in Fig. 2.80, $x_c(t) = \cos(2000\pi t)$, $T_s = \frac{1}{4000}$ sec find $x[n], x_d[n]$ and $y[n]$.

Solution 2.22 When continuous time signal is sampled, we get

$$x[n] = x_c(t)|_{t=nT_s} \rightarrow x[n] = \cos\left(2000\pi n \frac{1}{4000}\right) \rightarrow x[n] = \cos\left(\frac{\pi}{2} n\right). \tag{2.87}$$

After downsampling operation, we have

$$x_d[n] = x[3n] \rightarrow x_d[n] = \cos\left(\frac{3\pi}{2} n\right) \tag{2.88}$$

After upsampling operation, we have

$$y[n] = \begin{cases} x_d\left[\frac{n}{2}\right] & n \text{ is even} \\ 0 & \text{otherwise} \end{cases} \tag{2.89}$$

which yields

$$y[n] = \begin{cases} \cos\left(\frac{\pi}{4} n\right) & n \text{ is even} \\ 0 & \text{otherwise} \end{cases} \tag{2.90}$$

The mathematical expression in (2.90) can be written in a more compact manner as

$$y[n] = \frac{1 + \cos(\pi n)}{2} \cos\left(\frac{\pi}{4} n\right). \tag{2.91}$$

Using the property

$$\cos(a)\cos(b) = \frac{1}{2}(\cos(a+b) + \cos(a-b)) \tag{2.92}$$

Equation (2.91) can be written as

$$y[n] = \frac{1}{2}\cos\left(\frac{\pi}{4}n\right) + \frac{1}{4}\cos\left(\frac{5\pi}{4}n\right) + \frac{1}{4}\cos\left(\frac{3\pi}{4}n\right) \tag{2.93}$$

where using $\cos(\theta) = \cos(2\pi - \theta)$ Eq. (2.93) can be written as

$$y[n] = \frac{1}{2}\cos\left(\frac{\pi}{4}n\right) + \frac{1}{2}\cos\left(\frac{3\pi}{4}n\right). \tag{2.94}$$

Note: $\cos\left(\frac{5\pi}{4}n\right) = \cos\left(2\pi n - \frac{5\pi}{4}n\right) \rightarrow \cos\left(\frac{5\pi}{4}n\right) = \cos\left(\frac{3\pi}{4}n\right)$

Example 2.23 $x_c(t) = e^{jw_N t}$ and $x[n] = x_c(t)|_{t=nT_s}$, $T_s = 1$ find the Fourier transforms of $x_c(t)$ and $x[n]$.

Solution 2.23 The Fourier transform of the continuous time exponential signal is

$$X_c(w) = 2\pi\delta(w - w_N) \tag{2.95}$$

which is depicted in Fig. 2.81.

If $x[n] = x_c(t)|_{t=nT_s}$, then one period of the Fourier transform of $x[n]$ is

$$X_n(w) = \frac{1}{T_s}X_c\left(\frac{w}{T_s}\right), \qquad |w| < \pi \tag{2.96}$$

which is shown in Fig. 2.82.

Figure 2.82 can mathematically be expressed as $X_n(w) = 2\pi\delta(w - w_D)$, $|w| < 2\pi$. Since $X_n(w)$ is the Fourier transform of a digital signal, it is a periodic function and its period equals to 2π and it can be written as

$$X_n(w) = 2\pi\sum_{k=-\infty}^{\infty}\delta(w - w_D - k2\pi). \tag{2.97}$$

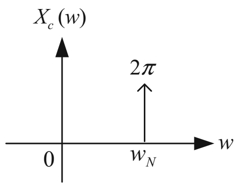


Fig. 2.81 Fourier transform of continuous time exponential signal

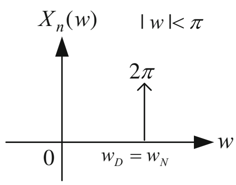


Fig. 2.82 One period of the Fourier transform of digital exponential signal

After sampling of the continuous time exponential signal, we obtain

$$x[n] = e^{j\underbrace{w_N T_s}_{w_D} n} \rightarrow x[n] = e^{jw_D n}.$$

Hence we can write the following transform pair in general

$$e^{jw_0 n} \overset{FT}{\leftrightarrow} 2\pi \sum_{k=-\infty}^{\infty} \delta(w - w_0 - k2\pi). \tag{2.98}$$

Example 2.24 Given $x[n] = e^{j\frac{\pi}{3}n}$, find Fourier transform of $x[n]$, i.e., $X_n(w)$.

Solution 2.24 $X_n(w) = 2\pi\delta(w - \frac{\pi}{3})$, $|w| < \pi$ and $X_n(w)$ is periodic with period 2π, so in more compact form, we can write it as

$$X_n(w) = 2\pi \sum_{k=-\infty}^{\infty} \delta\left(w - \frac{\pi}{3} - k2\pi\right) \tag{2.99}$$

Example 2.25 $x[n] = \cos(w_0 n)$, $y[n] = \cos(\frac{\pi}{3}n)$, $w[n] = \cos(\frac{2\pi}{3}n)$, find the Fourier transforms of $x[n], y[n]$, and $w[n]$.

Solution 2.25 We know that $\cos(\theta) = \frac{1}{2}\left(e^{j\theta} + e^{-j\theta}\right)$ and $\sin(\theta) = \frac{1}{2j}\left(e^{j\theta} - e^{-j\theta}\right)$, and using the Fourier transform of digital exponential function, we obtain the results

$$X_n(w) = \pi(\delta(w - w_0) + \delta(w + w_0)), \quad |w| < \pi$$
$$Y_n(w) = \pi\left(\delta\left(w - \frac{\pi}{3}\right) + \delta\left(w + \frac{\pi}{3}\right)\right), \quad |w| < \pi$$
$$W_n(w) = \pi\left(\delta\left(w - \frac{2\pi}{3}\right) + \delta\left(w + \frac{2\pi}{3}\right)\right), \quad |w| < \pi.$$

$X_n(w)$, $Y_n(w)$, and $W_n(w)$ are periodic functions with period 2π.

Example 2.26 The transfer function of a lowpass digital filter is depicted in Fig. 2.84. Accordingly, find the output of the block diagram shown in Fig. 2.83 for the input signal

$$x[n] = \cos\left(\frac{\pi}{3}n\right) + \cos\left(\frac{2\pi}{3}n\right).$$

The Fourier transform of the filter impulse is given as in Fig. 2.84.

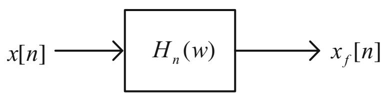


Fig. 2.83 Lowpass filtering of digital signals

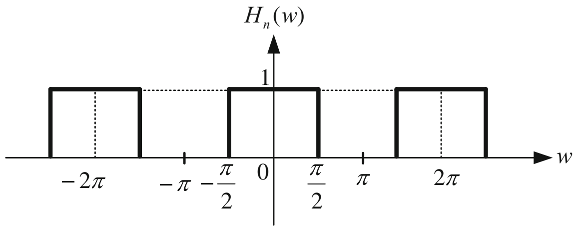


Fig. 2.84 Digital lowpass filter transfer function

Solution 2.26 If digital frequency w is between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$, that is if $|w| < \frac{\pi}{2}$, the digital frequency is accepted as low frequency. On the other hand, if $\frac{\pi}{2} < |w| < \pi$, the digital frequency is accepted as high frequency.

One period of Fourier transform of $x[n]$ can be calculated as

$$X_n(w) = \pi\left(\delta\left(w - \frac{\pi}{3}\right) + \delta\left(w + \frac{\pi}{3}\right)\right) + \pi\left(\delta\left(w - \frac{2\pi}{3}\right) + \delta\left(w + \frac{2\pi}{3}\right)\right), \ |w| < \pi \tag{2.100}$$

which is graphically illustrated in Fig. 2.85.

At the output of the block diagram, we have $X_{fn}(w) = H_n(w)X_n(w)$ and this multiplication is graphically illustrated in Fig. 2.86.

As it is obvious from Fig. 2.86, the signal $X_{fn}(w) = H_n(w)X_n(w)$ equals to

$$X_{fn}(w) = \pi\left(\delta\left(w - \frac{\pi}{3}\right) + \delta\left(w + \frac{\pi}{3}\right)\right). \tag{2.101}$$

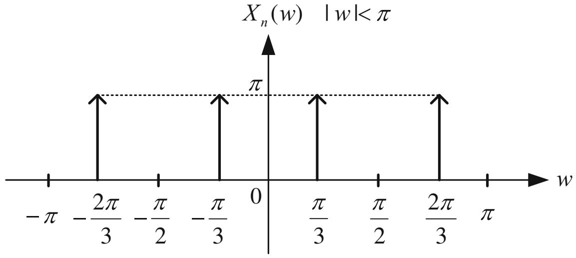


Fig. 2.85 Fourier transform of the input signal in Example 2.25

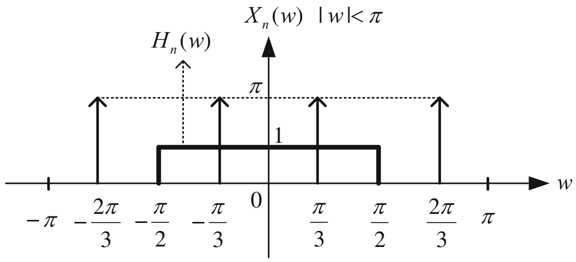


Fig. 2.86 Multiplication of $X_n(w)$ and $H_n(w)$

That is, high frequency part of the signal is filtered by the low pass filter, and at the output of the filter, only low frequency components exist. In time domain, the filter output equals to

$$x_f[n] = \cos\left(\frac{\pi}{3}n\right). \quad (2.102)$$

Example 2.27 In the system of Fig. 2.87, $x_c(t) = \cos(2000\pi t) + \cos(5000\pi t)$, $T_s = \frac{1}{3000}$ and transfer function of the digital filter is depicted in Fig. 2.88.

Find $x[n], x_f[n]$, and $x_d[n]$.

Solution 2.27 $x[n] = x_c(t)|_{t=nT_s}$ leads to

$$x[n] = \cos\left(\frac{2\pi}{3}n\right) + \cos\left(\frac{5\pi}{3}n\right). \quad (2.103)$$

Since $\cos\left(\frac{5\pi}{3}n\right) = \cos\left(2\pi n - \frac{5\pi}{3}n\right) \rightarrow \cos\left(\frac{5\pi}{3}n\right) = \cos\left(\frac{\pi}{3}n\right)$, then (2.103) becomes as

$$x[n] = \cos\left(\frac{2\pi}{3}n\right) + \cos\left(\frac{\pi}{3}n\right). \quad (2.104)$$

The digital filter eliminates high frequency component of $x[n]$, hence at the output of the filter we have

$$x_f[n] = \cos\left(\frac{\pi}{3}n\right). \quad (2.105)$$

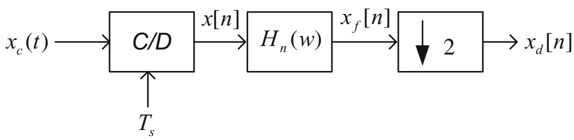


Fig. 2.87 Signal processing system for Example 2.26

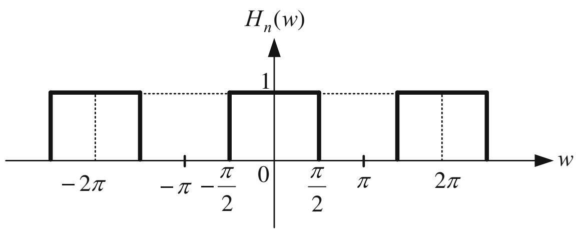


Fig. 2.88 Digital lowpass filter transfer function

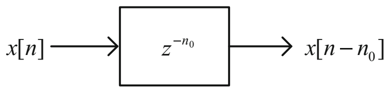


Fig. 2.89 Delay system

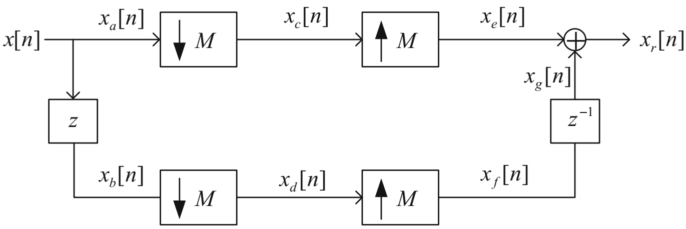


Fig. 2.90 Signal processing system for Example 2.27

After downsampling operation, we get

$$x_d[n] = x_f[2n] \to x_d[n] = \cos\left(\frac{2\pi}{3}n\right). \tag{2.106}$$

Example 2.28 The delay system is shown in Fig. 2.89.

In the system shown in Fig. 2.90, $M = 2$ and $x[n] = [1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6]$. Find $x_a[n]$, $x_b[n]$, $x_c[n]$, $x_d[n], x_e[n], x_f[n]$ and $x_r[n]$.

Solution 2.28 If $x[n] = [1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6]$, then $x_a[n] = [\underbrace{1}_{n=0} \quad 2 \quad 3 \quad 4 \quad 5 \quad 6]$ and since $x_b[n] = x[n+1]$ moving $n = 0$ pointer to the right by one unit, we get

$$x_b[n] = [1 \quad \underbrace{2}_{n=0} \quad 3 \quad 4 \quad 5 \quad 6]$$

After downsampling, we have

$$x_c[n] = [\underbrace{1}_{n=0} \quad 3 \quad 5] \quad x_d[n] = [\underbrace{2}_{n=0} \quad 4 \quad 6].$$

After upsampling, we have

$$x_e[n] = [\underbrace{1}_{n=0} \quad 0 \quad 3 \quad 0 \quad 5] \quad x_f[n] = [\underbrace{2}_{n=0} \quad 0 \quad 4 \quad 0 \quad 6].$$

After delay operator z^{-1}, we have

$$x_g[n] = [\underbrace{0}_{n=0} \quad 2 \quad 0 \quad 4 \quad 0 \quad 6].$$

And at the system output, we have

$$x_r[n] = x_e[n] + x_g[n]$$

where

$$x_e[n] = [\underbrace{1}_{n=0} \quad 0 \quad 3 \quad 0 \quad 5] \quad x_g[n] = [\underbrace{0}_{n=0} \quad 2 \quad 0 \quad 4 \quad 0 \quad 6].$$

Hence,

$$x_r[n] = [1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6].$$

The signal flow of the system in Fig. 2.90 is shown in Fig. 2.91.
Exercise: For the system given in Fig. 2.92, $M = 3$ and

$$x[n] = [1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \quad 11 \quad 12 \quad 13 \quad 14 \quad 15].$$

Find the output of every block and finally find $x_r[n]$.

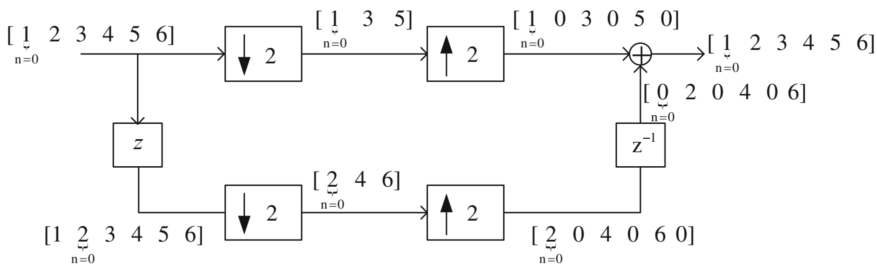


Fig. 2.91 Signal flow for the system in Fig. 2.90

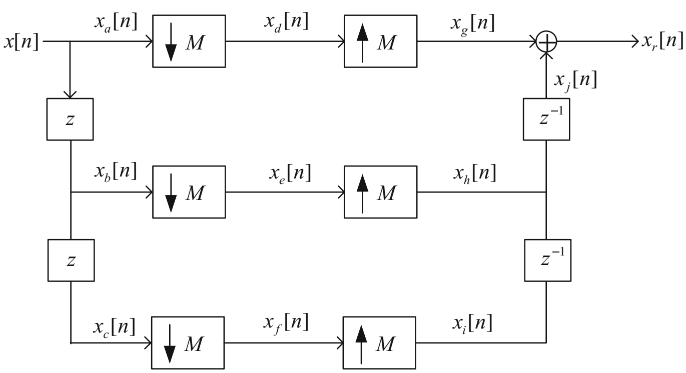


Fig. 2.92 Signal processing system for exercise

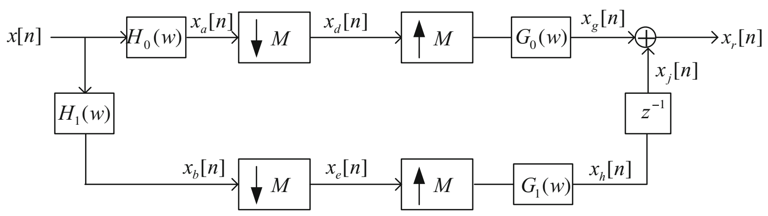


Fig. 2.93 Signal processing system for Example 2.28

Example 2.29 For the system shown in Fig. 2.93, $x[n] = \cos\left(\frac{2\pi}{3}n\right) + \cos\left(\frac{\pi}{3}n\right)$, $M = 2$.

Find $H_0(w)$, $H_1(w)$, $G_0(w)$, and $G_1(w)$ such that $x_r[n] = x[n]$.

Solution 2.29 $H_0(w)$ can be chosen as a low pass digital filter. $H_1(w)$ can be chosen as a high pass digital filter. $G_0(w)$ and $G_1(w)$ are interpolating $\sin c(\cdot)$ filters.

2.2.7 *Anti-aliasing Filter*

Consider the continuous to digital conversion system shown in Fig. 2.94.

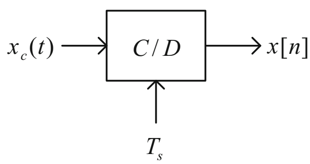
We know that to obtain one period the Fourier transform of $x[n]$, we multiply the frequency axis of the Fourier transform of $x_c(t)$ by T_s and multiply the amplitude axis of the Fourier transform of $x_c(t)$ by $1/T_s$, i.e., we calculate $\frac{1}{T_s}X_c\left(\frac{w}{T_s}\right)$. If the Fourier transform of $x_c(t)$ has a bandwidth greater than π/T_s, then $\frac{1}{T_s}X_c\left(\frac{w}{T_s}\right)$ extends beyond $(-\pi, \pi)$ and aliasing observed in the Fourier transform of $x[n]$. This situation is described in Fig. 2.95.

Since $X_n(w)$ is periodic with period 2π when $\frac{1}{T_s}X_c\left(\frac{w}{T_s}\right)$ extends beyond $(-\pi, \pi)$, overlapping will be observed in $X_n(w)$ as shown in Fig. 2.96.

The portion of $X_n(w)$ in Fig. 2.96 for $|w| < \pi$ is shown in Fig. 2.96.

To decrease the effect of aliasing (overlapping) in the digital signal, we can filter the spectral components for $|w| > \pi/T_s$ in $X_c(w)$ before sampling operation. In this way, we can eliminate the overlapping shaded parts in Fig. 2.97. We name this filter as anti-aliasing filter and it is mathematically defined as

Fig. 2.94 Continuous to digital conversion



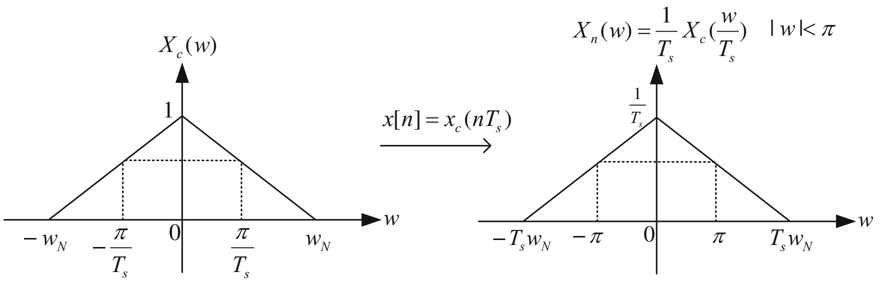


Fig. 2.95 Aliasing case in the Fourier transform of $x[n]$

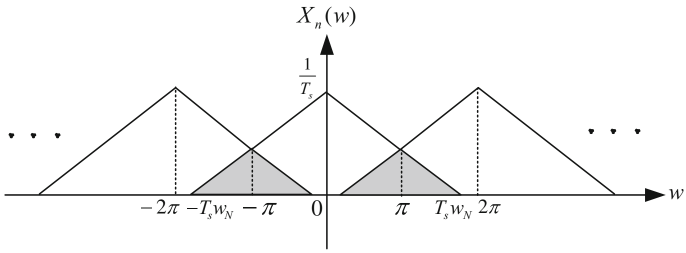
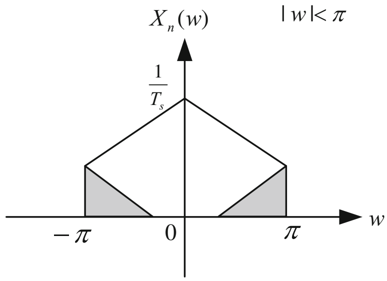


Fig. 2.96 Aliasing in $X_n(w)$

Fig. 2.97 $X_n(w)$ in Fig. 2.96 for $|w| < \pi$



$$H_{aa}(w) = \begin{cases} 1 & if \quad |w| < \frac{\pi}{T_s} \\ 0 & otherwise \end{cases} \tag{2.107}$$

whose time domain expression can be computed using inverse Fourier transform

$$h_{aa}(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H_{aa}(w) e^{jwt} dw$$

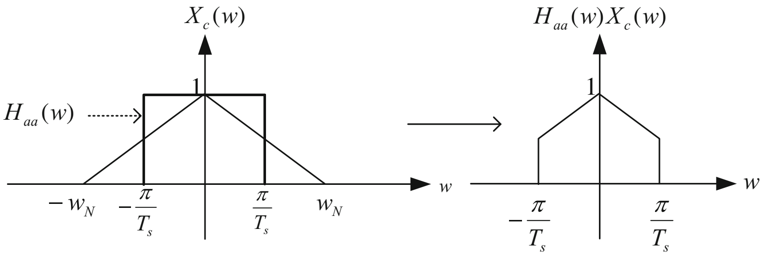


Fig. 2.98 Anti-aliasing filtering

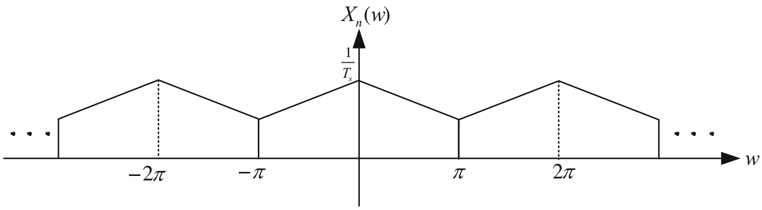


Fig. 2.99 The Fourier transform of a digital signal obtained by sampling of a continuous time signal filtered by an anti-aliasing filter

as

$$h_{aa}(t) = \frac{\sin\left(\frac{\pi t}{T_s}\right)}{\pi t}. \tag{2.108}$$

Anti-aliasing filtering is shown in Fig. 2.98.

The digital signal obtained after sampling of the filtered analog signal shown in Fig. 2.98 has the Fourier transform depicted in Fig. 2.99.

2.3 Practical Implementations of C/D and D/C Converters

Up to now we have studied theoretical C/D and D/C converter systems. However, the practical implementation of these units in real life shows some differences. The practical implementation of the C/D converter is shown in the first part of Fig. 2.100, and in a similar manner, the practical implementation of the D/C converter is shown in the second part of Fig. 2.100.

C/D and D/C conversion systems include analog-to-digital and digital-to-analog converter units and the contents of these units are shown in Fig. 2.100. Now we will inspect every component of the complete system shown in Fig. 2.100.

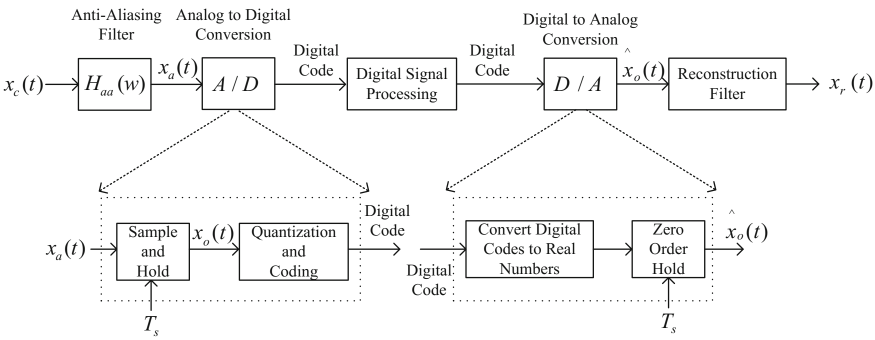


Fig. 2.100 Practical implementations of C/D and D/C converter systems

2.3.1 C/D Conversion

A practical C/D converter includes the units shown in Fig. 2.101.

Where antialiasing filter is used to decrease of amount of distortion in digital signal in case of aliasing. Antialiasing filter is defined as

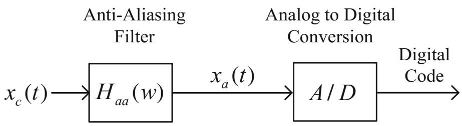
$$H_{aa} = \begin{cases} 1 & |w| < \frac{\pi}{T_s} \\ 0 & otherwise \end{cases} \tag{2.109}$$

Inside A/D converter, we have Sample-and-Hold and Quantizer-Coder units which are shown in Fig. 2.102.

For the coding of quantization levels, two's complement, one's complement or unsigned binary representations can be used.

Once the analog signal is represented by bit sequences, i.e., codes, these bit sequences are processed depending on the application. For instance, in digital communication, these bit sequences are encoded by channel codes and obtained bit sequences are converted to complex symbols, i.e., digitally modulated, and transmitted. In data storage, these bit sequences are again coded using forward error corrections codes, such as Reed Solomon codes as in compact disc storage, and stored. Alternatively, these bit sequences can be passed through data compression algorithms and then stored.

Fig. 2.101 Practical C/D converter.



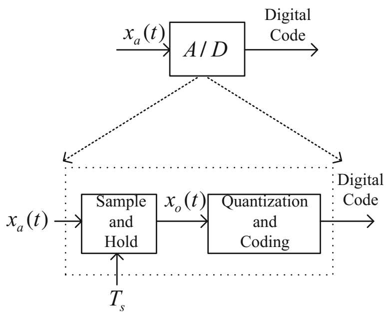


Fig. 2.102 Components of A/D converter

2.3.2 *Sample and Hold*

The aim of the sample and hold circuit is to produce a rectangular signal and the amplitudes of the rectangles are determined at the sampling time instants. The simplest sample and hold circuit as shown in Fig. 2.103 which is constructed using a capacitor.

Since usually sampling frequency f_s is a large number, such as 10 kHz etc., it is logical to use a digital switch for the place of a mechanical switch as shown in Fig. 2.104.

In the literature, much better sample and hold circuits are available. To give an idea about design improvement, the circuit in Fig. 2.104 can be improved by appending a buffer to the output preventing back current flows etc., and this improved circuit is shown in Fig. 2.105.

The sample and hold operation for the input sine signal is illustrated in Fig. 2.106.

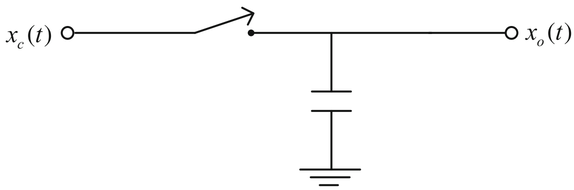


Fig. 2.103 A simple sample and hold circuit

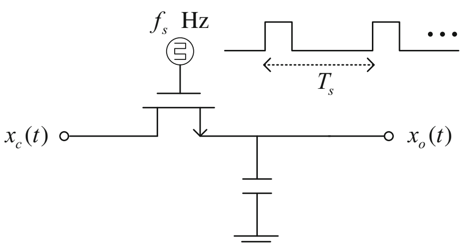


Fig. 2.104 Mechanical switch is replaced by an electronic switch

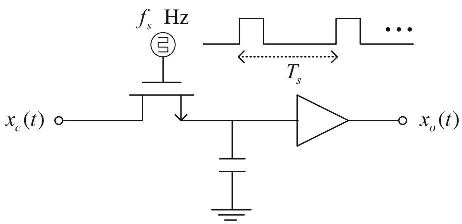


Fig. 2.105 Sample and hold circuit with a buffer at its output

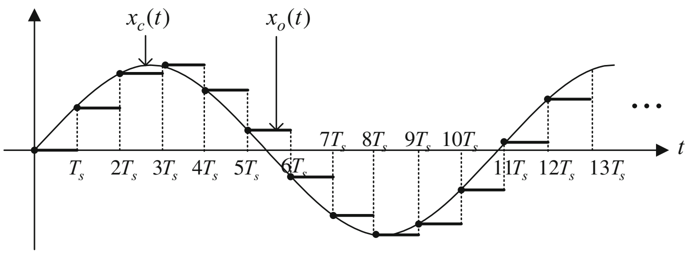


Fig. 2.106 Calculation of the output of the sample and hold circuit for sine input signal

For sine input signal after sample and hold operation, we obtain the signal $x_o(t)$ which is depicted alone in Fig. 2.107.

Question: Can we write a mathematical expression for the signal $x_o(t)$ shown in Fig. 2.107.

Yes, we can write. For this purpose, let's first define $h_o(t)$ function as shown in Fig. 2.108.

If the graph of $x_o(t)$ in Fig. 2.107 is inspected, it is seen that $x_o(t)$ signal is nothing but sum of the shifted and scaled $h_o(t)$ functions. Using $h_o(t)$ functions, we can write $x_o(t)$ as

$$x_o(t) = \sum_{k=-\infty}^{\infty} x_c(nT_s)h_o(t - nT_s) \rightarrow x_o(t) = \sum_{k=-\infty}^{\infty} x[n]h_o(t - nT_s). \qquad (2.110)$$

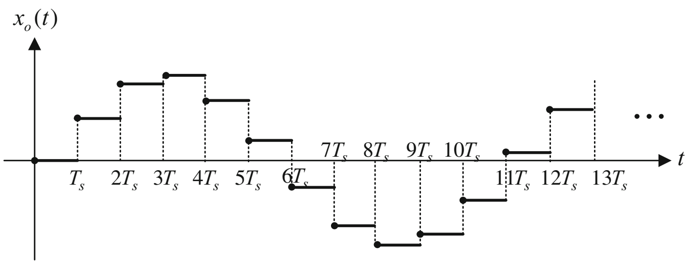


Fig. 2.107 Output of the sample and hold circuit for sine input signal

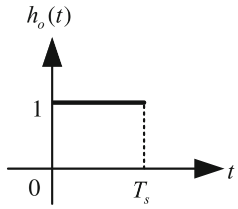


Fig. 2.108 Rectangle pulse signal

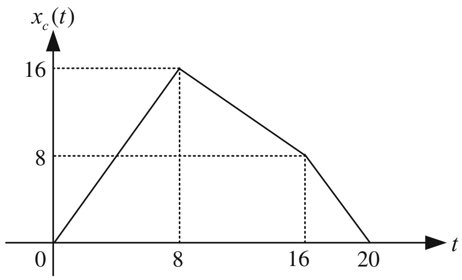


Fig. 2.109 Continuous time signal for sample and hold circuit

Example 2.30 The signal shown in Fig. 2.109 is passed through a sample and hold circuit. Find the signal at the output of the sample and hold circuit. Take sampling period as $T_s = 2$.

Solution 2.30 First we determine the amplitude values for the time instants t such that $t = nT_s$ where $T_s = 2$ and n is integer. This operation result is shown in Fig. 2.110. In addition, we also write the line equations for the computation of the amplitude values for the given time instants.

The amplitude values of the continuous time signal at time instants $t = nT_s$ are shown clearly in Fig. 2.111.

In the next step, we draw horizontal lines for the determined amplitudes, and for the first two samples, the drawn horizontal lines are shown in Fig. 2.112.

And for the first 4 samples, the horizontal drawn lines are shown in Fig. 2.113.

Repeating this procedure for all the other samples, we obtain the graph shown in Fig. 2.114.

The drawn horizontal lines for all the samples are depicted alone in Fig. 2.115.

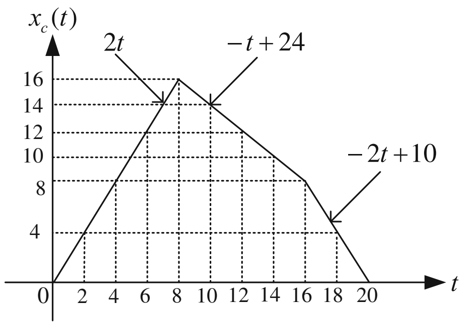


Fig. 2.110 The continuous time signal in details

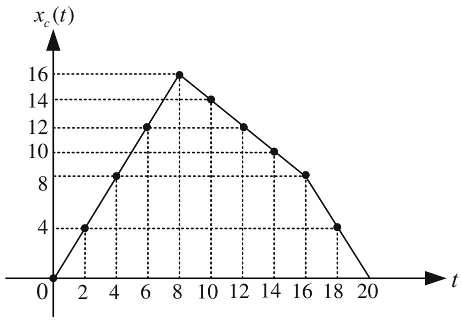


Fig. 2.111 Amplitudes shown explicitly for the time instants $t = nT_s$ where $T_s = 2$

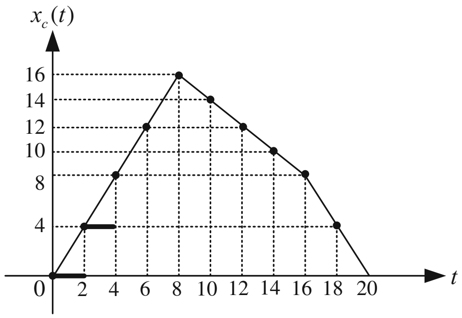


Fig. 2.112 Horizontal lines are drawn for the first two samples

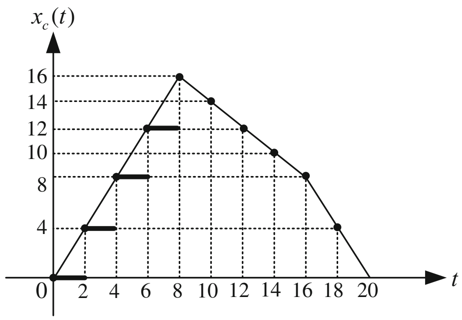


Fig. 2.113 Horizontal lines are drawn for the first four samples

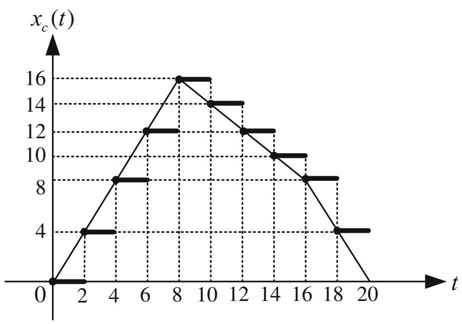
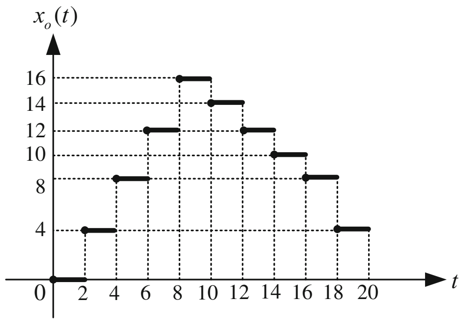


Fig. 2.114 Horizontal lines are drawn for all the samples

Fig. 2.115 Output of the sample and hold system



2.3.3 *Quantization and Coding*

During data storage or data transmission, we use bit sequences to represent real number. Since there are an infinite number of real numbers, it is not possible to represent this vast amount of real numbers by limited length bit streams. For this reason, we choose a number of real numbers to represent by bit streams and try to round other real numbers to the chosen ones when it is necessary to represent them by bit streams.

Mid-Level Quantizer

A typical quantizer includes the real number intervals used to map real numbers falling into these intervals to the quantization levels as shown in Fig. 2.116.

The quantizer in Fig. 2.116 is called mid-level quantizer. The quantizer maps the real numbers in the range $\left[-\frac{\Delta}{2}, \frac{\Delta}{2}\right)$ to Q_0, maps the real numbers in the range $\left[\frac{\Delta}{2}, \frac{3\Delta}{2}\right)$ to Q_1 etc. In this quantizer, Δ is called the step size of the quantizer. Smaller Δ means more sensitive quantizer. The mapping between real numbers and quantization levels is defined as $Q_i = Q(x)$ where Q_i may be chosen as the center of interleaves.

If Fig. 2.116 is inspected, it is seen that if we have equal number of intervals on the negative and positive regions, it means that the total number of intervals is an odd number, which is not a desired situation. Since using N bits, it is possible to represent 2^N levels. For this reason, we design these quantizers such that if one side has even number of intervals, then the other side has odd number of intervals.

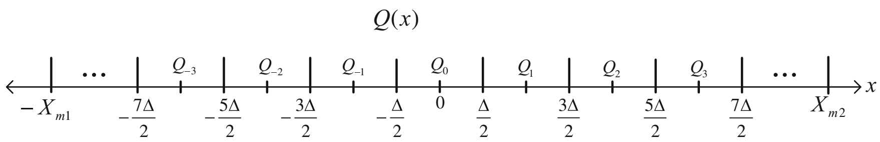


Fig. 2.116 A typical mid-level quantizer

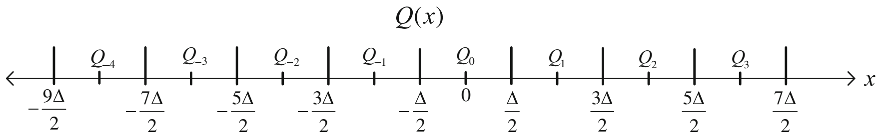


Fig. 2.117 Mid-level quantizer for Example 2.31

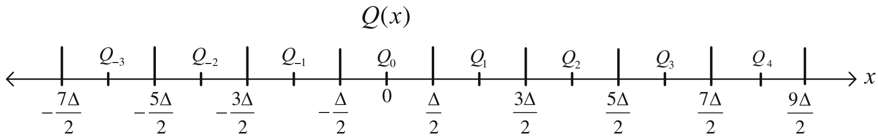


Fig. 2.118 An alternative mid-level quantizer for Example 2.31

Example 2.31 A 3-bit quantizer includes $2^3 = 8$ quantization intervals. A mid-level type quantizer consisting of 8 levels can be shown as in Fig. 2.117.

Or alternatively as in Fig. 2.118.

We will use mid-level quantizers as in Fig. 2.117.

As it is clear from the Example 2.30, for an N-bit mid-level quantizer, the minimum number that can be quantized is $-(2^N + 1)/2$ and the maximum number that can be quantized is $(2^N - 1)/2$.

The quantization levels are represented by binary sequences, such as two's complement, one's complements, unsigned representation, or private bit sequences can be assigned for quantization levels.

Example 2.32 Design a 3-bit quantizer for the real numbers in the range $[-14 \cdots 14]$.

Solution 2.32 For a 3-bit quantizer $X_{m1} = -9\Delta/2$ and $X_{m2} = 7\Delta/2$. Equating X_{m2} to -14, we obtain

$$\frac{7\Delta}{2} = 14 \rightarrow \Delta = 4.$$

So our quantizer can quantize the real numbers in the range

$$\left[-\frac{9\Delta}{2} \cdots \frac{7\Delta}{2}\right] = [-18 \cdots 14].$$

The bit sequences for our quantizer can be assigned to the intervals as in Fig. 2.119 and centers of the interleavers can be calculated as in Fig. 2.120.

Mid-Rise Quantizer

The mid-rise quantizer is shown in Fig. 2.121. As it is clear from Fig. 2.121, there is no interval centered at the origin.

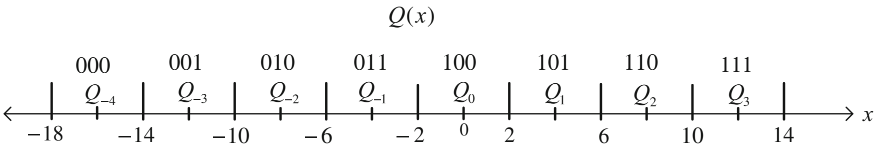


Fig. 2.119 Bit sequences assigned to the quantization intervals

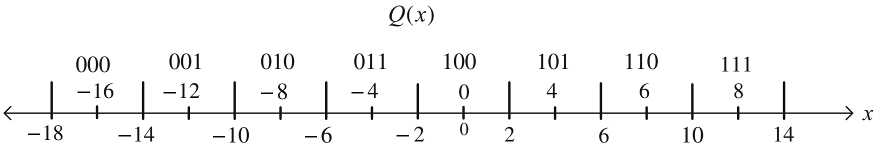


Fig. 2.120 Mid-level quantizer

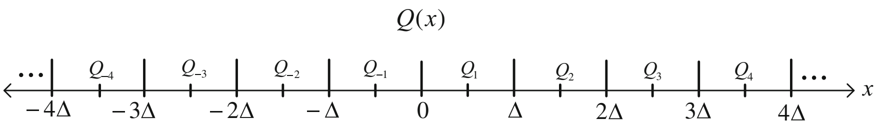


Fig. 2.121 Mid-rise quantizer

Assume that we want to quantize a sequence of digital samples represented by $x[n]$. Let $\hat{x}[n]$ be the sequence obtained after quantization. Since quantization distorts the original signal, the quantized samples mathematically can be written as

$$\hat{x}[n] = Q(x[n]) \rightarrow \hat{x}[n] = x[n] + e[n] \qquad (2.111)$$

where $e[n]$ is called quantization noise.

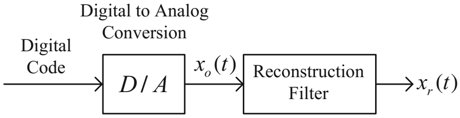
2.3.4 D/C Converter

The practical implementation of D/C converter is shown in Fig. 2.122.

The content of the D/A converter is detailed in Fig. 2.123.

The digital codes are converted to real numbers according to the used coding scheme. At the output of the code-to-digital converter, we have digital samples which can be written as

Fig. 2.122 D/C conversion



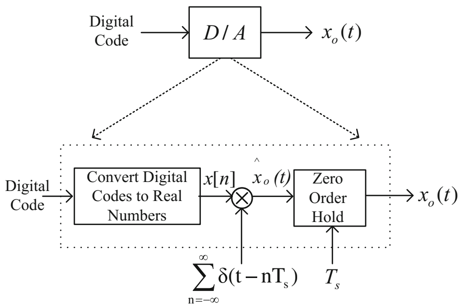


Fig. 2.123 D/A conversion

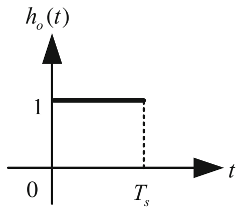


Fig. 2.124 Impulse response of zero order hold

$$\hat{x}[n] = x[n] + e[n] \tag{2.112}$$

where $e[n]$ is the quantization error. The zero order hold filter impulse response is shown Fig. 2.124.

The output of the code-to-digital converter in Fig. 2.123 is

$$\hat{x}_o(t) = \sum_{n=-\infty}^{\infty} \hat{x}[n]\delta(\mathrm{t} - \mathrm{nT_s}). \tag{2.113}$$

When $\hat{x}_o(t)$ is passed through zero order hold filter, we obtain

$$x_o(t) = \hat{x}_o(t) * h_o(t) \to x_o(t) = \sum_{n=-\infty}^{\infty} \hat{x}[n]\mathrm{h_o}(\mathrm{t} - \mathrm{nT_s}). \tag{2.114}$$

Substituting $\hat{x}[n] = x[n] + e[n]$ in (2.114), we get

$$x_o(t) = \sum_{n=-\infty}^{\infty} x[n]\mathrm{h_o}(\mathrm{t} - \mathrm{nT_s}) + \sum_{n=-\infty}^{\infty} e[n]\mathrm{h_o}(\mathrm{t} - \mathrm{nT_s}). \tag{2.115}$$

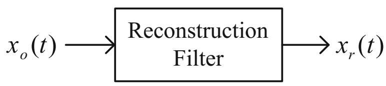


Fig. 2.125 Reconstruction filter block diagram

Now let's consider the last unit of the D/C converter the reconstruction filter as shown in Fig. 2.125.

The Fourier transform of $x_o(t)$ in (2.115) can be calculated using

$$X_o(w) = \sum_{n=-\infty}^{\infty} x[n]\mathrm{H_o}(w)e^{-jwnT_s} + \sum_{n=-\infty}^{\infty} e[n]\mathrm{E_o}(w)e^{-jwnT_s} \tag{2.116}$$

where taking the common term $\mathrm{H_o}(w)$ outside the parenthesis, we obtain

$$X_o(w) = \left(\underbrace{\sum_{n=-\infty}^{\infty} x[n]e^{-jwnT_s}}_{X_n(T_s w)} + \underbrace{\sum_{n=-\infty}^{\infty} e[n]e^{-jwnT_s}}_{E_n(T_s w)} \right) \mathrm{H_o}(w) \tag{2.117}$$

which can be written as

$$X_o(w) = (X_n(T_s w) + E_n(T_s w))H_o(w). \tag{2.118}$$

From Fig. 2.125, we can write

$$X_r(w) = H_r(w)X_o(w) \tag{2.119}$$

where $H_r(w)$ is the frequency response of the reconstruction filter. If we choose $H_r(w)$ as

$$H_r(w) = \begin{cases} \frac{T_s}{H_o(w)} & |w| < \frac{\pi}{T_s} \\ 0 & otherwise \end{cases} \tag{2.120}$$

and substituting it into (2.119) and using (2.118) in (2.119), we obtain

$$X_r(w) = T_s X_n(T_s w) + T_s E_n(T_s w) \quad |w| < \frac{\pi}{T_s} \tag{2.121}$$

which is the Fourier transform of

$$x_r(t) = x_a(t) + e(t). \tag{2.122}$$

Since $x[n] = x_a(nT_s)$, $e[n] = e(nT_s)$, the continuous time signals $x_a(t)$ and $e(t)$ can be obtained from their samples using

$$x_a(t) = \sum_{n=-\infty}^{\infty} x[n] \sin c\left(\frac{t - nT_s}{T_s}\right) \tag{2.123}$$

and

$$e(t) = \sum_{n=-\infty}^{\infty} e[n] \sin c\left(\frac{t - nT_s}{T_s}\right). \tag{2.124}$$

Then $x_r(t)$ in (2.122) using (2.123) and (2.124) can be written as

$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n] \sin c\left(\frac{t - nT_s}{T_s}\right) + \sum_{n=-\infty}^{\infty} e[n] \sin c\left(\frac{t - nT_s}{T_s}\right).$$

2.4 Problems

(1) $x[n] = [1 \quad 2 \quad 0 \quad -3 \quad -1 \quad 1 \quad 4 \quad -1 \quad 0 \quad 1 \quad -2 \quad 5 \quad 1 \quad 3]$ is given. Find the signals $x[2n]$, $x[3n]$, $x[4n]$, $x[n/2]$, $x[n/3]$, and $x[n/4]$.

(2) One period of the Fourier transform of $x[n]$ around origin is shown in Fig. 2.126. Draw the Fourier transform of the downsampled signal $y[n] = x[2n]$.

(3) The delay system is described in Fig. 2.127.

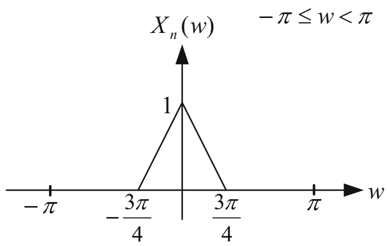


Fig. 2.126 One period of $X_n(w)$ around origin

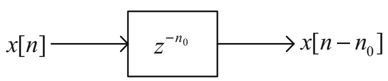


Fig. 2.127 Delay system

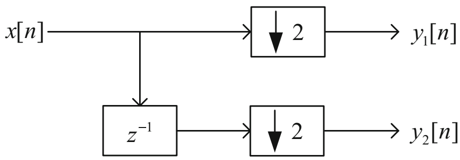


Fig. 2.128 Signal processing system

If

$$x[n] = [a \quad b \quad c \quad d \quad e \quad f \quad \underset{n=0}{g} \quad h \quad \imath \quad j \quad k \quad l \quad m \quad n \quad o \quad p \quad r]$$

find the output of each unit in Fig. 2.128.

(4) Calculate the inverse Fourier transform of the digital filter

$$H_{dn}(w) = \begin{cases} 1 & if \quad w| < \frac{\pi}{M} \\ 0 & if \quad \frac{\pi}{M} < |w| < \pi. \end{cases} \tag{2.125}$$

(5) Draw the graph of

$$h_{dn}[n] = \frac{\sin\left(\frac{\pi n}{M}\right)}{\pi n} \tag{2.126}$$

roughly, and find the triangle approximation of (2.126). Calculate the approximated model for $n = -5, \ldots, 5$.

(6) The graph of $X(t)$ is shown in Fig. 2.129. Considering Fig. 2.129 draw the graph of

$$Y(t) = \sum_{k=-\infty}^{\infty} X(t - kT), \quad T = 3. \tag{2.127}$$

Fig. 2.129 The graph of $X(t)$

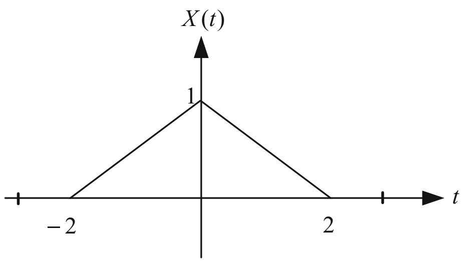


Fig. 2.130 Downsampler

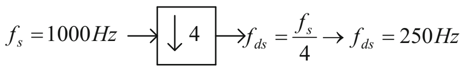
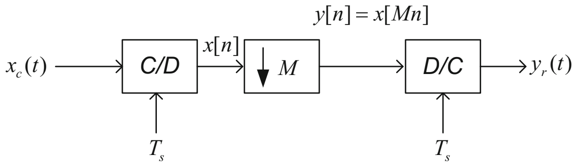


Fig. 2.131 System for Question 9



(7) Repeat Question-6 for $T = 1$, $T = 4$ and $T = 5$.

(8) Comment on the system shown in Fig. 2.130.

(9) For the system of Fig. 2.131, $x_c(t)$ is a lowpass signal with bandwidth 3000 Hz, $T_s = \frac{1}{8000}$ s and $M = 2$. Is system output $y_r(t)$ equal to system input $x_c(t)$? If they are equal to each other, justify the reasoning behind it. If they are not equal to each other, again explain the reasoning behind it.

(10) If $x[n] = [1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7]$ and $L = 4$, draw the graph of

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n - kL].$$

(11) For the system of Fig. 2.132, $M = L = 2$ and

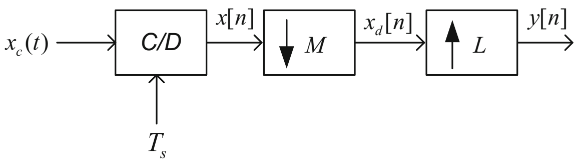
$$x[n] = [a \quad b \quad c \quad d \quad e \quad f \quad g \quad h \quad \underbrace{l}_{n=0} \quad j \quad k \quad l \quad m \quad n \quad o \quad p \quad r \quad s].$$

Find $x_d[n]$ and $y[n]$.

(12) Draw the graph of $h_{ai}[n] = -\frac{|n|}{L} + 1$, $-L \leq n \leq L$ for $L = 3$ and $L = 8$.

(13) $x_d[k] = [1 \quad 4 \quad 7 \quad 10 \quad 13]$, $h_{ai}[n] = -\frac{|n|}{L} + 1$, $-L \leq n \leq L$, $L = 3$, calculate and draw

Fig. 2.132 Signal processing system



$$y_i[n] = \sum_{k=-\infty}^{\infty} x_d[k] h_{ai}[n - kL].$$

(14) For the system of Fig. 2.133, $x[n] = \cos\left(\frac{\pi}{4} n\right) \ 0 \le n \le 10$, $h_{ai}[n]$ is the triangle approximated reconstruction filter. Find $x_d[n], y[n]$ and $y_i[n]$ for $M = L = 2$.

(15) For the system of Fig. 2.134,

$$H_{aa}(w) = \begin{cases} 1 & if \ |w| < \frac{\pi}{T_s} \\ 0 & otherwise \end{cases}$$

Express the Fourier transform of $x[n]$ in terms of the Fourier transform of $x_c(t)$.

(16) For the system of Fig. 2.135, $M = 3$, $X_n(w)$ is the one period of the Fourier transform of $x[n]$. Draw the Fourier transform of $x_d[n]$.

(17) For the system of Fig. 2.136, $M = L = 2$ and $X_n(w)$ is the one period of the Fourier transform of $x[n]$.

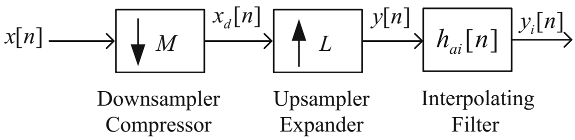


Fig. 2.133 Signal processing system for Question 14

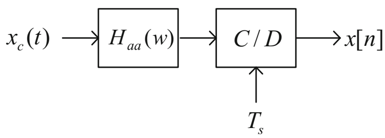


Fig. 2.134 Signal processing system for Question 15

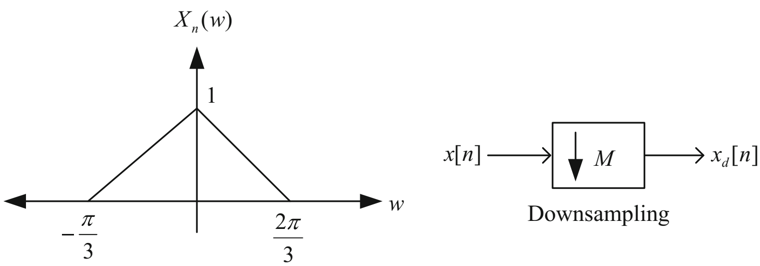


Fig. 2.135 Downsampling of digital signal

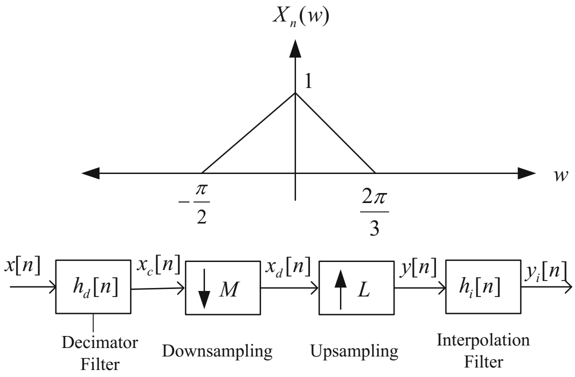


Fig. 2.136 Signal processing system for Question 17

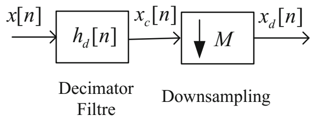


Fig. 2.137 Decimation system

(a) Draw the Fourier transforms of $x_c[n], x_d[n]$, and $y[n]$.
(b) Draw the triangle approximation model of the interpolation filter for $L = 2$.
(c) Draw the Fourier transform of $y_i[n]$ for $\sin c(\cdot)$ interpolation filter.
(d) If $x_c[n] = [1.0 \quad 1.7 \quad 2.4 \quad 3.2 \quad 4]$, calculate $x_d[n], y[n]$, using triangle approximated interpolation filter.

(18) For the system of Fig. 2.137, $M = 2$, and $H_d(w)$ is defined as

$$H_d(w) = \begin{cases} 1 & if \quad |w| \le \frac{\pi}{M} \\ 0 & otherwise \end{cases} \tag{2.128}$$

(a) Calculate the inverse Fourier transform of $H_d(w)$, i.e., calculate $h_d[n]$. Next, find the triangle approximated model of $h_d[n]$.
(b) For $x[n] = [\,1 \quad 2 \quad 3 \quad 4\,]$ calculate $x_o[n]$ and $x_d[n]$.

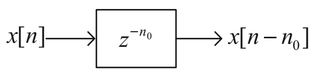


Fig. 2.138 Delay system

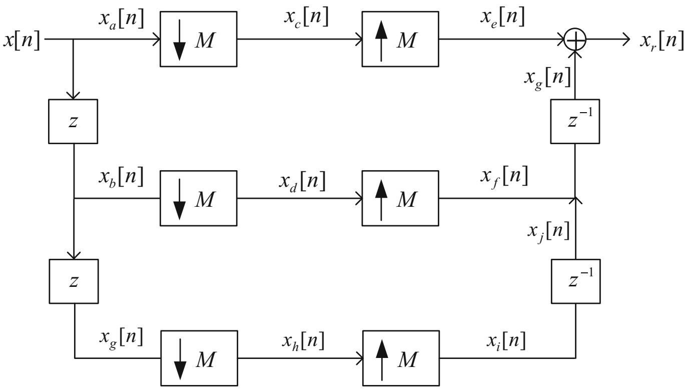


Fig. 2.139 Signal processing system for Question 19

(19) The delay system is shown in Fig. 2.138.

For the system of Fig. 2.139, $M = 3$, $x[n] = [a \ b \ c \ d \ e \ f \ g \ h \ i \ j \ k \ l \ m \ n \ o \ p \ r \ s \ t \ u \ v \ w \ x \ y]$. Find the signal at the output of each unit, and find the system output.

http://www.springer.com/978-981-10-4961-3

Understanding Digital Signal Processing
Gazi, O.
2018, IX, 303 p. 334 illus., Hardcover
ISBN: 978-981-10-4961-3