Chapter 11

Multirate Signal Processing

Multirate signal processing refers to systems which allow sequences which arise from *different* sampling rates to be processed together.

There are two early applications that motivate its use in digital audio. Suppose we have an audio signal $x_a(t)$ which has a significant energy only up to $f_M = 22,000$Hz. One way is to implement a bandlimiting filter of width 22kHz with a sharp transition from pass-band to stop-band as illustrated in the previous chapter. This requires the design

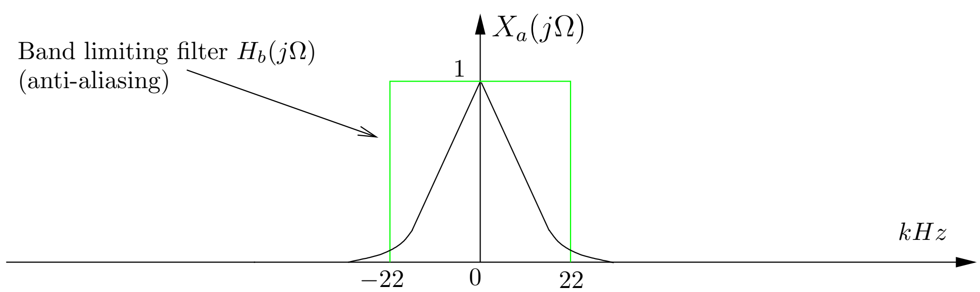


Figure 11.1: Spectrum of the continuous signal.

of a very good analog filter. This is illustrated in Figures 11.1 and 11.2.

Another method is to have a much less stringent front-end anti-aliasing filter and then *oversampling* it, *i.e.,* sample at a rate larger (*e.g.* 44kHz) than required and then change sampling rate in discrete domain.

Then we pass the over-sampled discrete-time signal through a digital filter and then *down-sample* by a proper factor (a factor of 2 in this example), *i.e.,* drop every second discrete sample to get back to original sampling rate. Figure 11.3 shows the steps of this method.

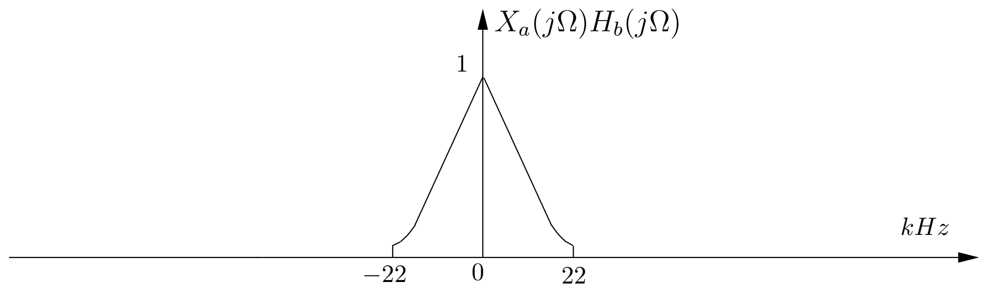


Figure 11.2: Filtering with an ideal analog low-pas filter.

The second application is in altering sampling rate of a system. For example CD is sampled at 44kHz and DAT (for digital audio players) is sampled at 48kHz. In order to convert the sampling rate, instead of going back to continuous-time and re-sample the signal again, we can do this alteration purely in discrete domain.

We have seen that periodic sampling of a continuous-time signal $x_c(t)$ at a sampling rate of $\frac{1}{T_s}$ is given by

$$x[n] = x_c(nT_s).$$

As seen in the digital audio example, it is sometimes necessary to change the sampling rate of a discrete-time signal to obtain a new discrete-time representation of the underlying continuous-time signal $x_c(t)$ as,

$$y[n] = x_c(nT_s')$$

for a different sampling period $T_s' \neq T_s$. A trivial approach to obtain such a sequence $y[n]$ from $x[n]$ would be to reconstruct $x_c(t)$ from $x[n]$ using the optimal interpolator, and then resample the reconstructed signal with period T_s'. Often this is not desirable since we would have non-ideality in the reconstruction filter (interpolation). Therefore it is of interest to consider methods that change the sampling rate by only discrete-time operations. For example if $T_s' = MT_s$, then we see that since

$$x[n] = x_c(nT_s), \qquad y[n] = x_c(nT_s') = x_c(nMT_s),$$

we can write directly the new discrete-time sequence in terms of $x[n]$ as

$$y[n] = x[Mn],$$

and therefore obtain it completely by discrete-time operations.

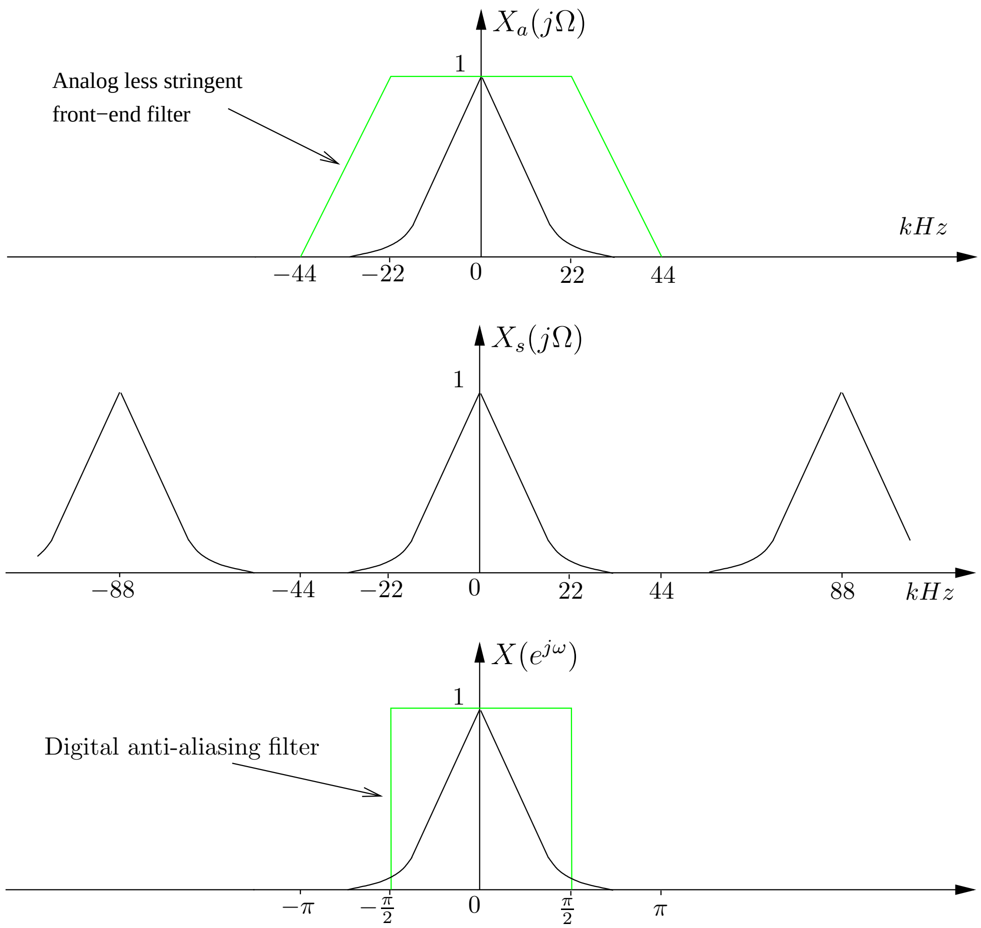


Figure 11.3: Filtering with a much less stringent front-end anti-aliasing filter.

11.1 Downsampling: Sampling Rate Reduction by an Integer Factor

This need for change in sampling rate was seen in the motivating digital audio example where one might *oversample* a signal (*i.e.*, sample at a rate higher than necessary) and then after discrete-time processing reduce the sampling rate by *sub-sampling* or *decimation* or *down-sampling*. This means keeping only every M-th sample of the discrete-time process. This is usually represented as shown in Fig. 11.4.

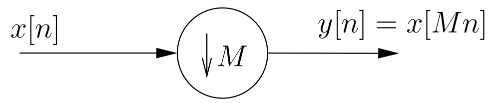


Figure 11.4: Down-sampling by factor M.

Example 11.1 *The MP3 audio supports a range of sampling frequencies, 8kHz, 11.025 kHz, 12 kHz, 16kHz, 22.05kHz, 24kHz, 32kHz, 44.1kHz, and 48kHz. For example for high end audio music since the human ear can at most listen up to 20kHz, if we band-limit audio at 20kHz, a sampling rate of 44.1kHz (standard for CDs) or 48kHz (standard for DAT) is suitable. However, if one wants to either save storage space or the content is a speech then one can have a smaller sampling rate. Suppose we have an MP3 file already at sampling rate 44.1kHz, and one wants to convert it into a 22.05kHz format, what is the most efficient way to do it?*

An obvious or naive method is to play-back the audio and resample the continuous-time signal at the lower rate. However, we notice that since the information needed for lower sampling rate is already contained in the 44.1kHz sampled audio, therefore by retaining only every other samples from the 44.1kHz sampled audio one can produce an MP3 file in the 22.05 kHz sampling rate format.

For example the down-sampling is shown pictorially for a discrete-time sequence in Fig. 11.6.

Example 11.2 $x[n] = \cos\left(\frac{2\pi}{3}n\right)$

$$y[n] = x[3n] = \cos(2\pi n) = 1$$

This sampling process is shown in Fig. 11.7.

Note that the origin of time is very important, since if we down-sample $x'[n] = x[n+1]$ instead of $x[n]$, the output is

$$y'[n] = x'[2n] = x[2n+1] \neq x[2n] = y[n].$$

Example 11.3 *If we down-sample from*

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

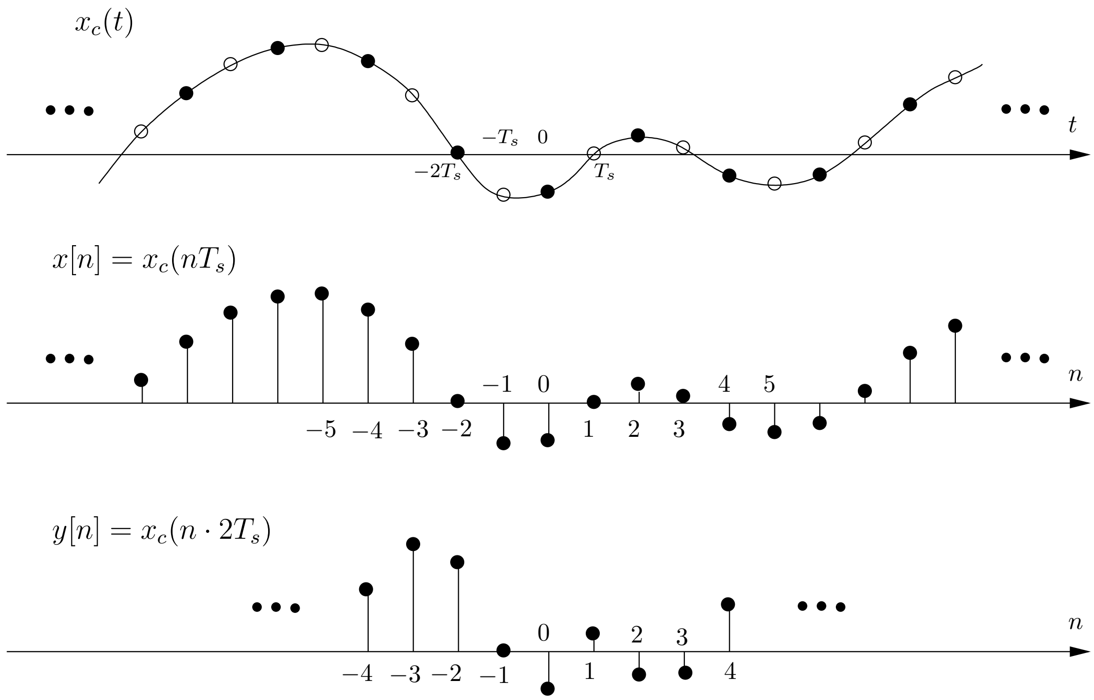


Figure 11.5: Sampling from a continuous-time signal with different sampling periods.

and

$$x'[n] = x[n+1] = \cos\left(\frac{2\pi}{3}(n+1)\right) = \cos\left(\frac{2\pi}{3}n + \frac{2\pi}{3}\right)$$

with down-sampling factor 3*, we have*

$$y'[n] = x'[3n] = \cos\left(\frac{2\pi}{3} \cdot 3n + \frac{2\pi}{3}\right) = \cos\frac{2\pi}{3} = -\frac{1}{2},$$

and clearly $y'[n] \neq y[n] = x[3n] = 1$.

Therefore, down-sampling is *not* a time-invariant operation.

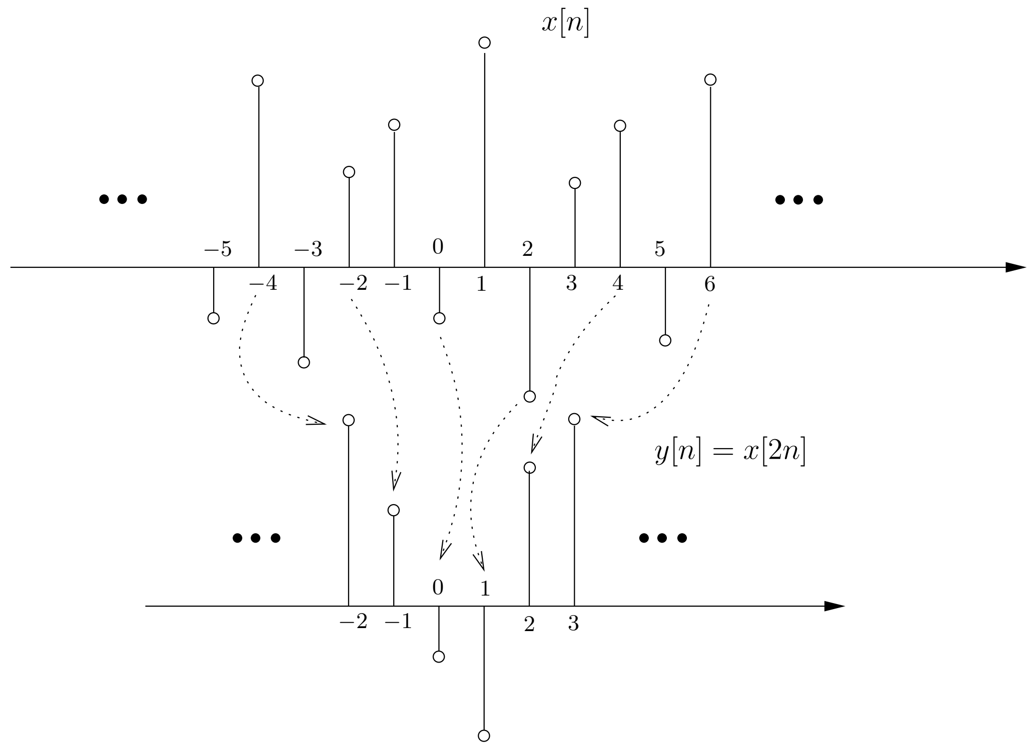


Figure 11.6: Down-sampling by factor 2.

Example 11.4 *Let*

$$
\begin{aligned}
x[n] &= \sin\left(\frac{2\pi}{3}n\right), \\
x_1[n] &= x[n-1] = \sin\left(\frac{2\pi}{3}n - \frac{2\pi}{3}\right), \\
x_2[n] &= x[n-2] = \sin\left(\frac{2\pi}{3}n - \frac{4\pi}{3}\right), \\
x_3[n] &= x[n-3] = \sin\left(\frac{2\pi}{3}n - 2\pi)\right) = \sin\left(\frac{2\pi}{3}n\right).
\end{aligned}
$$

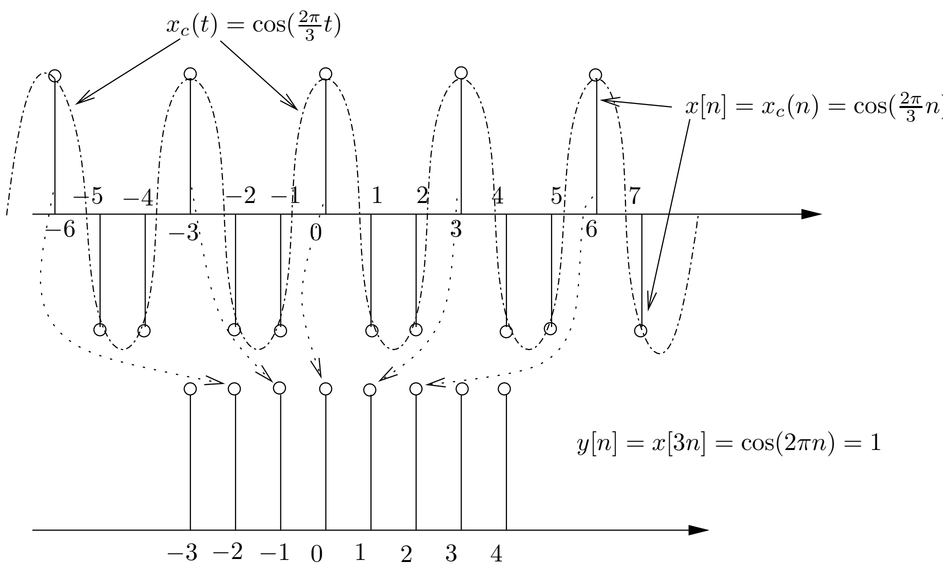


Figure 11.7: Down-sampling from $x_c(t) = \cos(\frac{2\pi}{3}t)$ and down-sampling factor 3.

Hence

$$\begin{aligned} y[n] &= x[3n] = \sin(2\pi n) = 0, \\ y_1[n] &= x_1[3n] = x[3n-1] = \sin\left(-\frac{2\pi}{3}n\right) = -\sin\left(\frac{2\pi}{3}n\right) = -\frac{\sqrt{3}}{2}, \\ y_2[n] &= x_2[3n] = x[3n-2] = \sin\left(-\frac{4\pi}{3}n\right) = \frac{\sqrt{3}}{2}, \\ y_3[n] &= x_3[3n] = x[3n-3] = \sin(2\pi n) = 0. \end{aligned}$$

Here we see a periodicity in the shifting property.

Example 11.5 *Let we sample from the sequences*

$$\begin{aligned} x[n] &= \left(\frac{1}{2}\right)^n u[n], \\ x_1[n] &= x[n-1] = \left(\frac{1}{2}\right)^{n-1} u[n-1], \\ x_2[n] &= x[n-2] = \left(\frac{1}{2}\right)^{n-2} u[n-2] \end{aligned}$$

with down-sampling factor 2*. Then we have*

$$\begin{aligned} y[n] &= x[2n] = \left(\frac{1}{2}\right)^{2n} u[2n], \\ y_1[n] &= x_1[2n] = x[2n-1] = \left(\frac{1}{2}\right)^{2n-1} u[2n-1], \\ y_2[n] &= x_2[2n] = x[2n-2] = \left(\frac{1}{2}\right)^{2n-2} u[2n-2] = \left(\frac{1}{2}\right)^{2(n-1)} u[2(n-1)] = y[n-1]. \end{aligned}$$

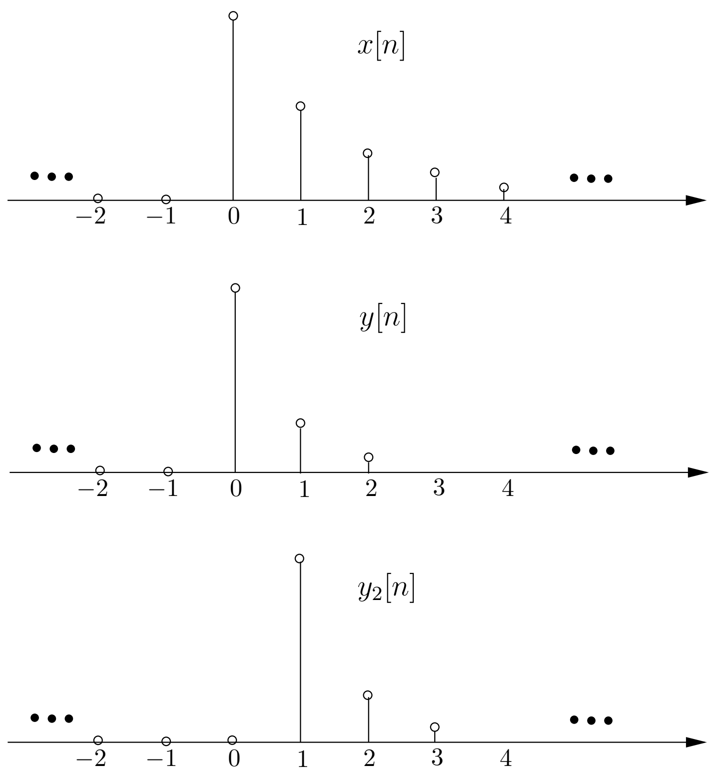


Figure 11.8: Sampling from $x[n] = \left(\frac{1}{2}\right)^n u[n]$ in Example 11.5.

If we denote by $D_M(\cdot)$, the operator that down-samples a signal by M, *i.e.,*

$D_M\,(x[n]) = x[Mn]$,

then we can state the following property.

Property: Down-sampling by M or $D_M(\cdot)$ is a linear, periodically time-varying operator with period M.

Proof: Since

$$D_M\left(\alpha x_1[n] = \beta x_2[n]\right) = \alpha x_1[Mn] + \beta x_2[Mn] = \alpha D_M\left(x_1[n]\right) + \beta D_M\left(x_2[n]\right),$$

clearly $D_M(\cdot)$ is a linear operator.

It has a *periodically time-varying* property because if a sequence is shifted by M, its down-sampled version is shifted by 1. More precisely,

$$\begin{aligned} D_M\left(\delta[n-kM]\right) &= \delta[n-k], \qquad k \in \mathbb{Z} \\ D_M\left(\delta[n-kM-\ell]\right) &= 0, \qquad \ell = 1, 2, \ldots, M-1. \end{aligned}$$

Hence, if $y[n] = D_M\left(x[n]\right)$, then $D_M\left(x[n-kM]\right) = y[n-k]$. ■

Therefore, for the down-sampling system the time-varying property implies that complex sinusoids are *no longer eigen-functions*. This can be seen from the following argument. Let

$$x[n] = e^{j\pi n} = (-1)^n.$$

If we down-sample by a factor 2,

$$y[n] = x[2n] = e^{j\pi 2n} = 1 \neq c \cdot x[n]$$

for any constant c. Hence the complex sinusoid is not an eigen-function for the down-sampling operator.

Recall that for a linear time-invariant system, the complex exponential was an eigen-function, *i.e.,* if

$$\mathcal{L}\left\{x[n]\right\} = \sum_m h[m]x[n-m],$$

then for $x[n] = e^{j\omega_0 n}$,

$$y[n] = \mathcal{L}\left\{e^{j\omega_0 n}\right\} = H\left(e^{j\omega_0}\right)x[n]$$

was a scaled version of $x[n]$ with scaling factor $H\left(e^{j\omega_0}\right)$ which is a constant independent of n.

We have now seen that the time-invariant property is crucial for this to be true. This is because $D_M(\cdot)$ is linear but *not* time-invariant and does not have the complex exponential as an eigen-sequence or eigen-function.

An important consideration is to understand what happens to the z-transform (or discrete-time Fourier transform) when one down-samples a signal. To be specific, we first consider down-sampling by a factor of 2 of a discrete-time signal with z-transform $X(z)$.

First let us define a new discrete-time sequence $x'[n]$ with z-transform $X'(z)$ as

$$\begin{aligned}
X'(z) &= \frac{1}{2}\left[X(z) + X(-z)\right] = \frac{1}{2}\sum_k x[k]z^{-k} + \frac{1}{2}\sum_k x[k](-1)^{-k}z^{-k} \\
&= \frac{1}{2}\sum_k \left\{x[k] + (-1)^{-k}x[k]\right\} z^{-k} \\
&= \sum_{k:\text{even}} x[k]z^{-k} + \underbrace{\sum_{k:\text{odd}} x[k](1-1)z^{-k}}_{0} \\
&= \sum_n x[2n]z^{-2n}.
\end{aligned}$$

Now, if

$$y[n] = x[2n],$$

then

$$Y(z) = \sum_n y[n]z^{-n} = \sum_n x[2n]z^{-n} = X'\left(z^{\frac{1}{2}}\right) = \frac{1}{2}\left[X\left(z^{\frac{1}{2}}\right) + X\left(-z^{\frac{1}{2}}\right)\right].$$

Evaluating it on the unit circle, if the ROC includes it, gives

$$Y\left(e^{j\omega}\right) = \frac{1}{2}\left[X\left(e^{j\frac{\omega}{2}}\right) + X\left(e^{-j\frac{\omega}{2}}\right)\right] = \frac{1}{2}\left[X\left(e^{j\frac{\omega}{2}}\right) + X\left(e^{j(\frac{\omega}{2}-\pi)}\right)\right].$$

Example 11.6 *Let*

$$x[n] = a^n u[n], \qquad 0 < a < 1.$$

Then

$$X(z) = \frac{1}{1-az^{-1}}, \qquad \textit{ROC:}\ \ |z| > a.$$

Now define

$$y[n] = x[2n] = a^{2n}u[2n] = \begin{cases} (a^2)^n, & n \geq 0 \\ 0 & \text{else.} \end{cases}$$

Hence

$$Y(z) = \frac{1}{1-a^2z^{-1}}, \qquad \textit{ROC:}\ \ |z| > a^2.$$

But we see that

$$\begin{aligned}\frac{1}{2}\left[X\left(z^{\frac{1}{2}}\right)+X\left(-z^{\frac{1}{2}}\right)\right] &= \frac{1}{2}\left[\frac{1}{1-az^{-\frac{1}{2}}}+\frac{1}{1+az^{-\frac{1}{2}}}\right] \\ &= \frac{1}{2}\left[\frac{2}{(1-az^{-\frac{1}{2}})(1+az^{-\frac{1}{2}})}\right]=\frac{1}{1-a^2z^{-1}} \\ &= Y(z)\end{aligned}$$

with ROC: $|z^{\frac{1}{2}}|>a$ *or* $|z|>a^2$.

This illustrates the following important points:

- The output is a function of not only $X(z)$ but also $X(-z)$ (which corresponds to the z-transform of a sequence $(-1)^n x[n]$, *i.e.,* a modulated version of $x[n]$).
- Since we have a time-scaling by a factor of 2 in going from $x'[n]$ to $y[n]$, it corresponds to $z \longrightarrow z^{\frac{1}{2}}$ or $e^{j\omega} \longrightarrow e^{j\omega/2}$ (contraction in time implies expansion in frequency).

Example 11.7 *Let* $x[n]$ *be a sequence with the spectrum* $X\left(e^{j\omega}\right)$ *given in Fig. 11.9. Let*

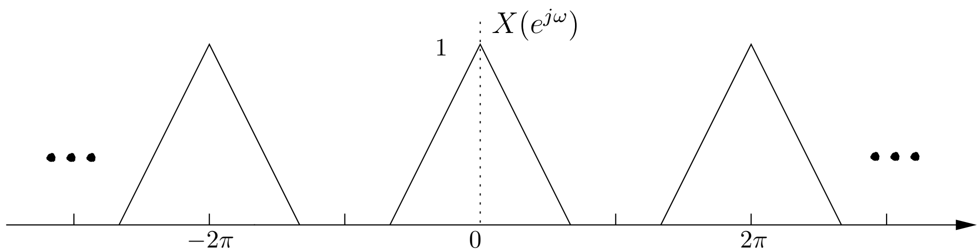


Figure 11.9: $X\left(e^{j\omega}\right)$.

$y[n]=x[2n]$ *be a down-sampled version of* $x[n]$ *with factor* 2. *Note that* $Y\left(e^{j\omega}\right)=X'\left(e^{j\frac{\omega}{2}}\right)$, *where*

$$X'\left(e^{j\omega}\right)=\frac{1}{2}\left[X\left(e^{j\omega}\right)+X\left(e^{j(\omega-\pi)}\right)\right].$$

Now consider the plot shown in Fig. 11.10, wherein $X\left(e^{j\omega}\right)$ *is shown in solid line and* $X\left(e^{j(\omega-\pi)}\right)$ *is in dashed line. Therefore* $X'\left(e^{j\omega}\right)$ *can be constructed as given in Fig. 11.11.*

Finally, we construct $Y\left(e^{j\omega}\right)$ *where* $Y\left(e^{j\omega}\right)=X'\left(e^{j\frac{\omega}{2}}\right)$ *means that it just expands the* ω *axis in going from* $X'\left(e^{j\omega}\right)$ *to* $Y\left(e^{j\omega}\right)$.

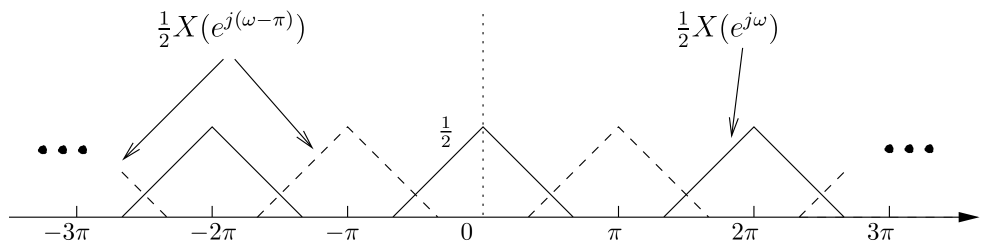


Figure 11.10: $\frac{1}{2}X\left(e^{j\omega}\right)$ in solid lines and $\frac{1}{2}X\left(e^{(j\omega-\pi)}\right)$in dashed lines.

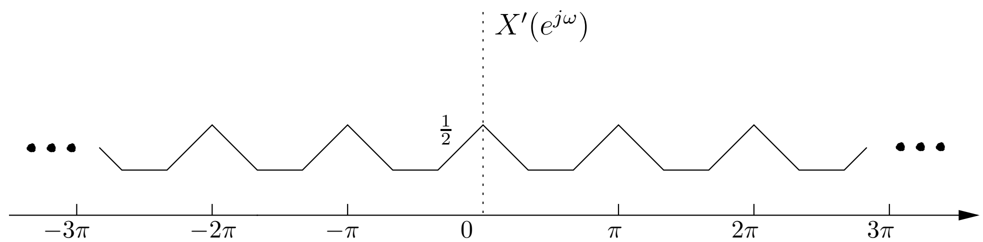


Figure 11.11: $X'\left(e^{j\omega}\right)$.

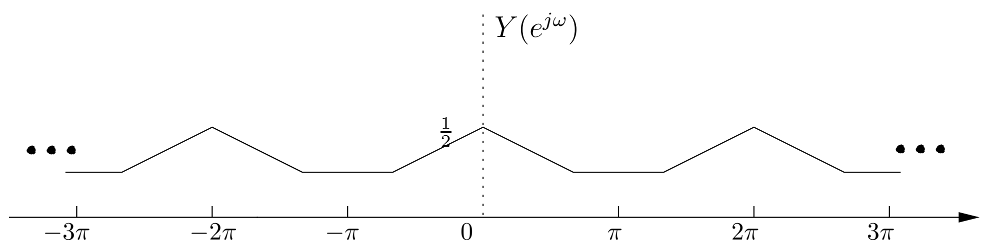


Figure 11.12: $Y\left(e^{j\omega}\right)$.

The result for down-sampling by 2 generalizes to arbitrary integer M as follows.

Proposition 11.1 *Given a signal $x[n]$ with z-transform $X(z)$, its down-sampled by factor M as*

$$y[n] = D_M\left(x[n]\right) = x[Mn]$$

has the following z-transform

$$Y(z) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(e^{-j\frac{2\pi}{M}k} z^{\frac{1}{M}}\right),$$

and if the unit circle is in the ROC then,

$$Y\left(e^{j\omega}\right) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(e^{j\left(\frac{\omega}{M}-\frac{2\pi}{M}k\right)}\right).$$

Proof. Using the same idea as done for the $M = 2$ case, let

$$\begin{aligned}
X'(z) &= \frac{1}{M}\sum_{k=0}^{M-1} X\left(e^{-j\frac{2\pi}{M}k} z\right) \\
&= \frac{1}{M}\sum_{k=0}^{M-1}\sum_{n} x[n]\left(e^{-j\frac{2\pi}{M}k} z\right)^{-n} \\
&= \sum_{n} x[n]\underbrace{\left\{\frac{1}{M}\sum_{k=0}^{M-1} e^{-j\frac{2\pi}{M}kn}\right\}}_{\delta[n-\ell M],\text{ for } \ell\in\mathbb{Z}} z^{-n} \\
&= \sum_{\ell} x[\ell M] z^{-\ell M}.
\end{aligned}$$

Then we see that

$$X'\left(z^{\frac{1}{M}}\right) = \sum_{\ell} x[\ell M] z^{-\ell} = Y(z).$$

11.2 Filtering and Down-sampling

As seen in Example. 11.7, we notice that the down-sampled version has aliasing. Because of this, it is usually better to low-pass filter a signal before sub-sampling. Let us do Example. 11.7 again along with this filtering.

Example 11.8 *Let $x[n]$ be a sequence for which the DTFT is given in Fig. 11.13. We pass it through a low-pass filter with cut-off frequency at $\frac{\pi}{2}$. Thus, we obtain $X_{LP}\left(e^{j\omega}\right)$ as shown in Fig. 11.14. Thus $X'\left(e^{j\omega}\right) = \frac{1}{2}\left[X'\left(e^{j\frac{\omega}{2}}\right) + X'\left(e^{j\left(\frac{\omega}{2}-\pi\right)}\right)\right]$ and $Y\left(e^{j\omega}\right)$, the DTFT of $y[n] = x[2n]$ will be obtained as in Fig. 11.15 and Fig. 11.16.*

Therefore, we see that

$$X_{LP}(z) = H(z)X(z)$$

where $H(z)$ is a low-pass filter operation. From here it is clear that for $y[n] = x_{LP}[2n]$,

$$\begin{aligned} Y(z) &= \frac{1}{2}\left[X_{LP}\left(z^{\frac{1}{2}}\right) + X_{LP}\left(-z^{\frac{1}{2}}\right)\right] \\ &= \frac{1}{2}\left[H\left(z^{\frac{1}{2}}\right)X\left(z^{\frac{1}{2}}\right) + H\left(-z^{\frac{1}{2}}\right)X\left(-z^{\frac{1}{2}}\right)\right]. \end{aligned}$$

This operator is presented in Fig. 11.17.

If the unit circle is in the ROC, then

$$Y\left(e^{j\omega}\right) = \frac{1}{2}\left[H\left(e^{j\frac{\omega}{2}}\right)X\left(e^{j\frac{\omega}{2}}\right) + H\left(e^{j\left(\frac{\omega}{2}-\pi\right)}\right)X\left(e^{j\left(\frac{\omega}{2}-\pi\right)}\right)\right].$$

Similar to Proposition. 11.1, we see that for a general filtering and down-sampling by M (see fig. 11.17), it follows that if $y[n] = x[Mn]$, then

$$Y(z) = \frac{1}{M}\sum_{k=0}^{M-1} H\left(e^{-j\frac{2\pi}{M}k}z^{\frac{1}{M}}\right)X\left(e^{-j\frac{2\pi}{M}k}z^{\frac{1}{M}}\right).$$

11.3 Upsampling: Increasing the Sampling Rate by an Integer Factor

We have seen that the reduction of the sampling rate of a discrete-time signal by an integer factor involves sampling the sequence in a manner analogous to sampling the continuous-time signal. A natural question to ask is whether we can *increase* the sampling rate. Clearly we cannot get more information by such an operation, but this is useful when we want to change the sampling rate by a rational factor as we will see later.

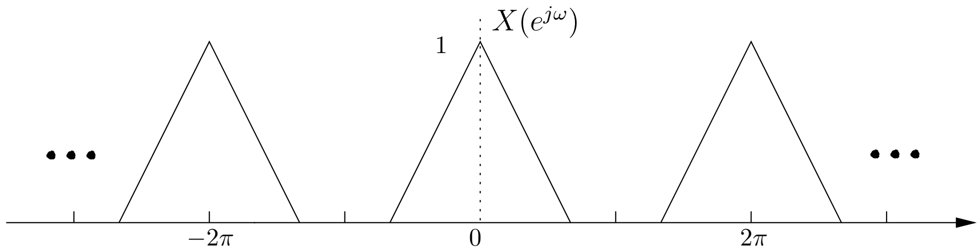


Figure 11.13: $X\left(e^{j\omega}\right)$.

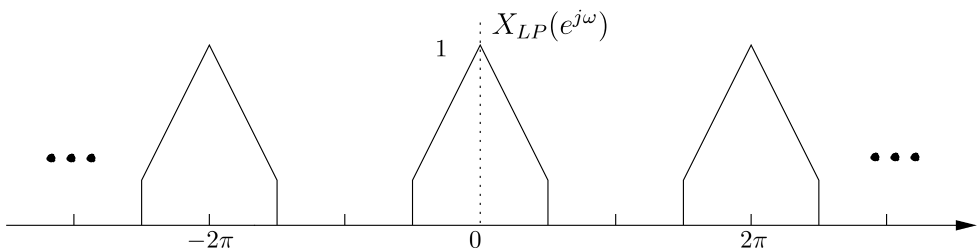


Figure 11.14: $X_{LP}\left(e^{j\omega}\right)$.

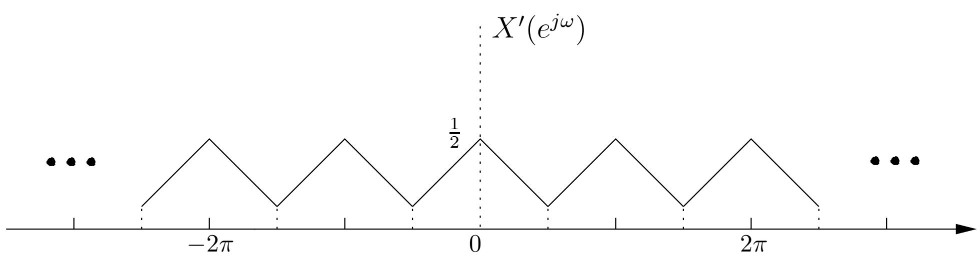


Figure 11.15: $X'\left(e^{j\omega}\right)$.

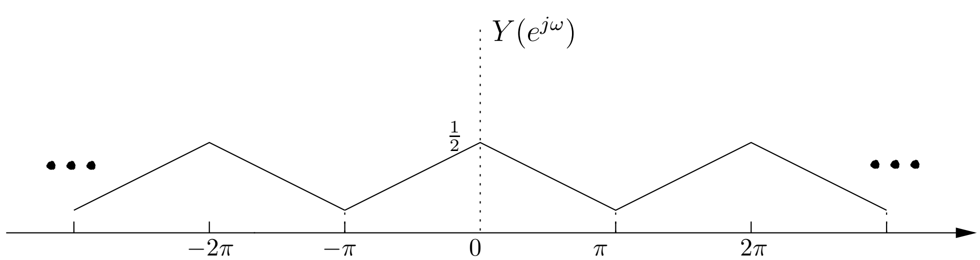


Figure 11.16: $Y\left(e^{j\omega}\right)$.

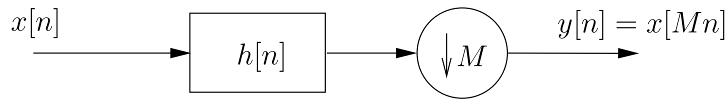


Figure 11.17: Filtering before subsampling. Typically filter is a low-pass filter with cut-off frequency $\frac{\pi}{M}$.

Suppose we have a sequence $x[n]$ whose sampling rate we wish to increase by a factor L. If we consider the underlying continuous-time signal $x_c(t)$, the objective is to obtain samples

$$x'[n] = x_c(nT_s{}'), \tag{11.1}$$

where $T_s{}' = \frac{T_s}{L}$, from the sequence of samples

$$x[n] = x_c(nT_s). \tag{11.2}$$

We will refer to the operation of increasing the sampling rate as *up-sampling*. From (11.1) and (11.2) it follows that

$$x'[n] = x\left[n/L\right] = x_c\left(nT_s/L\right) \text{ for } n = kL,\, k \in \mathbb{Z}.$$

For the samples $n \neq kL$, $k \in \mathbb{Z}$, we simply replace it with zero, *i.e.*, $y[n]$ is the up-sampled version of $x[n]$ if

$$y[n] = \begin{cases} x\left[n/L\right] & n = kL, k \in \mathbb{Z} \\ 0 & \text{otherwise.} \end{cases} \tag{11.3}$$

Equivalently,

$$y[n] = \sum_{k=-\infty}^{+\infty} x[k]\delta[n-kL]. \tag{11.4}$$

Example 11.9 *See Fig. 11.18.*

Let us now compute the z-transform of the upsampled version of a sequence. We denote the up-sampling operator by $\mathrm{U}_L(\cdot)$. Then we have,

$$\begin{aligned} Y(z) &= \sum_n y[n]z^{-n} = \sum_n \left(\sum_{k=-\infty}^{+\infty} x[k]\delta[n-kL] \right) z^{-n} \\ &= \sum_{k=-\infty}^{+\infty} x[k]z^{-kL} = X\left(z^L\right). \end{aligned} \tag{11.5}$$

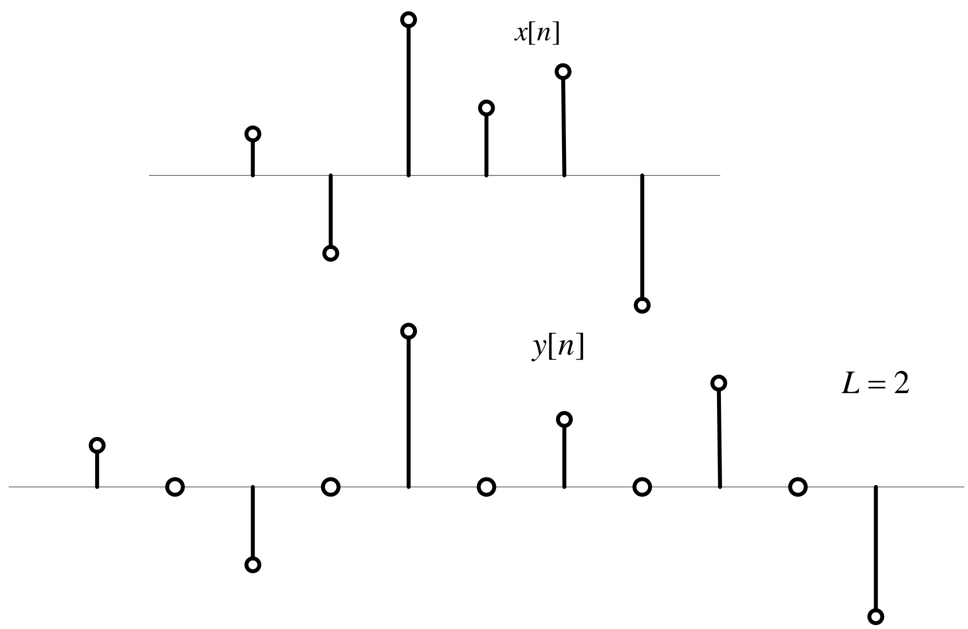


Figure 11.18: Example. 11.9

If the unit circle is in the ROC, then

$$Y\left(e^{j\omega}\right) = x\left(e^{j\omega L}\right).$$

Note that unlike downsampling, where there is a loss of information (and hence is not an invertible operation), upsampling can be easily inverted since,

$$\mathrm{D}_L\left(\mathrm{U}_L\left(x[n]\right)\right) = x[n]. \tag{11.6}$$

Example 11.10 *The up-sampling and its inverse are illustrated in Fig. 11.19.*

The up-sampling property is summarized in the following proposition.

Proposition 11.2 *If*

$$\begin{aligned} y[n] &= U_L\left(x[n]\right) = \begin{cases} x\left[n/L\right] & n = kL, k \in \mathbb{Z} \\ 0 & \textit{else}, \end{cases} \\ &= \sum_{k=-\infty}^{+\infty} x[k]\delta[n-kL], \end{aligned} \tag{11.7}$$

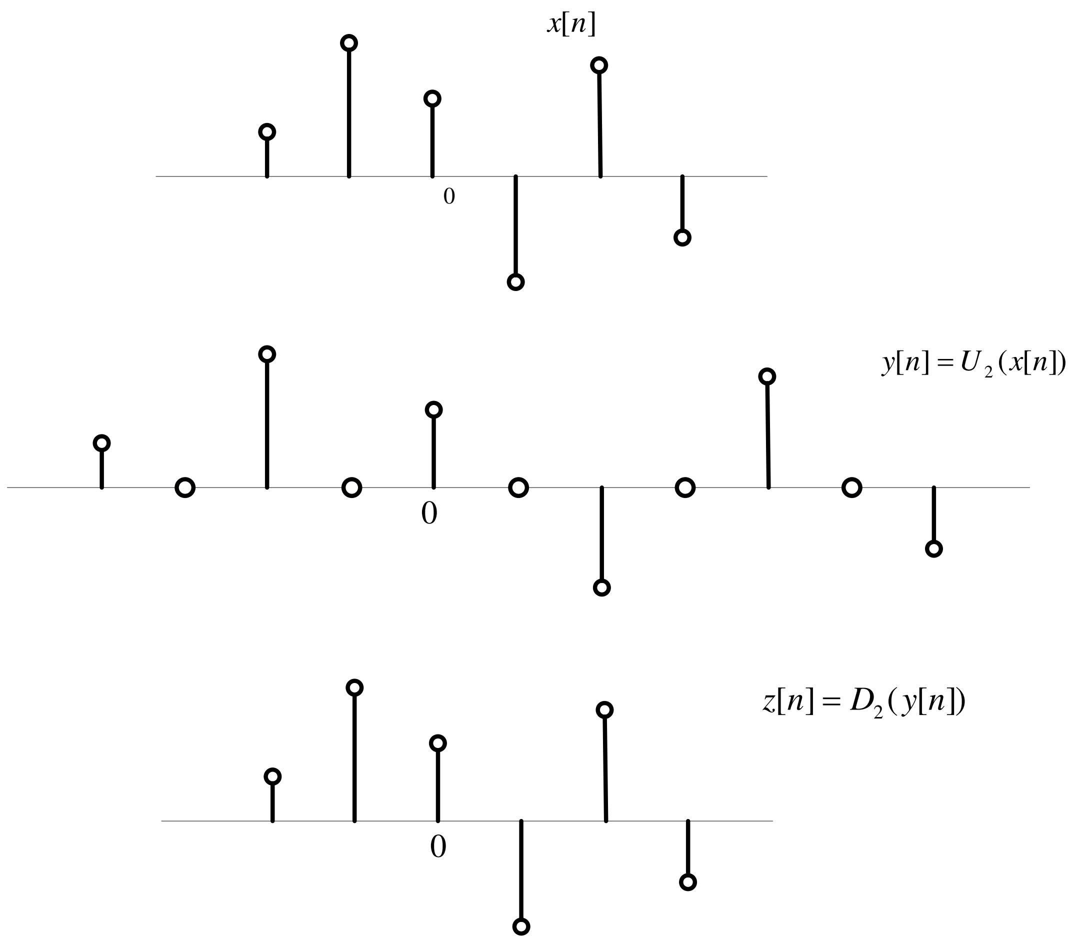


Figure 11.19: Up-sampling and down-sampling.

then

$$Y(z) = X\left(z^L\right), \tag{11.8}$$

and if $|z| = 1 \in \mathrm{ROC}_x$,

$$Y(e^{j\omega}) = X\left(e^{j\omega L}\right). \tag{11.9}$$

From (11.9) it is clear that up-sampling determines a contraction of the frequency axis.

Example 11.11 *Let* $x[n] = x_c(nT_s)$ *corresponds to the spectrum shown in Fig. 11.20, where* $T_s = 2\Omega_N$ *and* $y[n] = U_2\,(x[n])$.

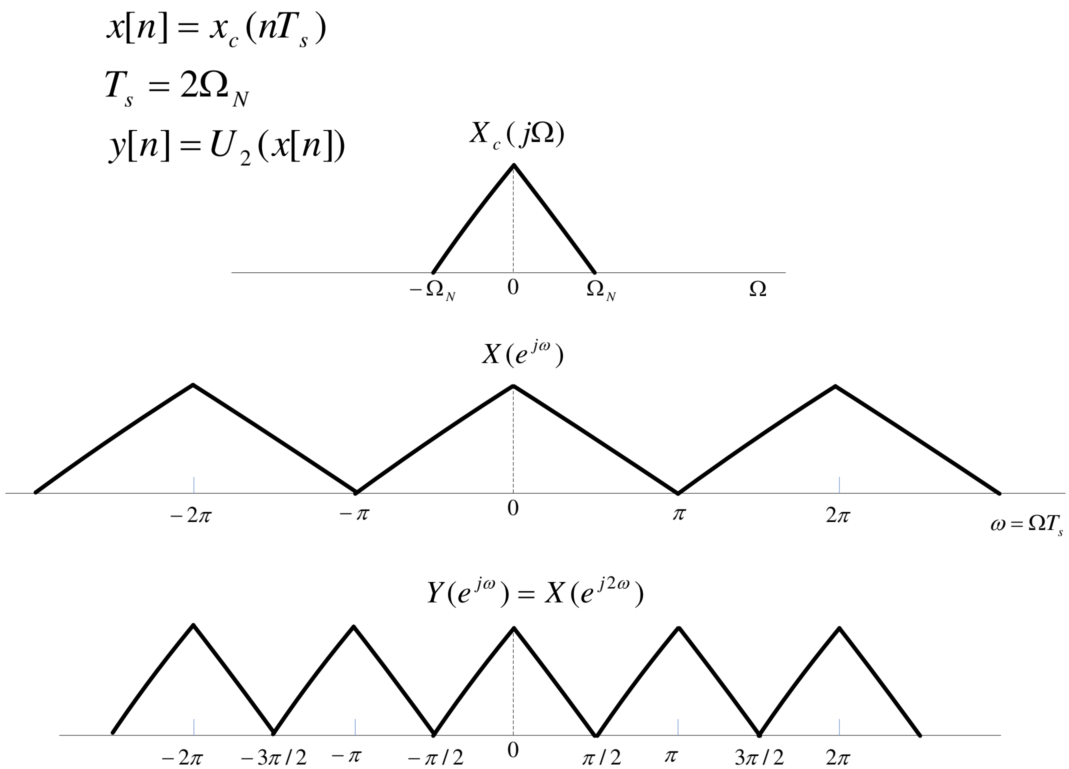


Figure 11.20: Up-sampling contracts the frequency axis.

Recall that the goal of up-sampling is to obtain samples at a higher rate than available. Clearly such a goal is impossible if we had not sampled the underlying continuous-time signal at the required sampling rate.

For example, if we had sample $x[k]$, we could find a reconstruction through optimal interpolation as

$$x_r(t) = \sum_k x[k]\mathrm{sinc}\left(\frac{t - kT_s}{T_s}\right).$$

Then we could resample this at sampling period $T_s{}'$ to produce

$$z[n] = x_r(nT_s{}') = \sum_k x[k]\mathrm{sinc}\left(\frac{nT_s{}' - kT_s}{T_s}\right).$$

If $\frac{T_s{}'}{T_s} = \frac{1}{L}$, then we have

$$z[n] = \sum_k x[k]\mathrm{sinc}\left(\frac{n}{L} - k\right) = \sum_k x[k]\mathrm{sinc}\left(\frac{n - kL}{L}\right).$$

Now

$$\mathrm{sinc}\left(\frac{n-kL}{L}\right) = 0 \quad \text{for} \quad n = kL \pm L, kL \pm 2L, \ldots$$

For $n = kL$,

$$\mathrm{sinc}\left(\frac{n-kL}{L}\right) = 1.$$

Therefore we see that for $n = mL, m \in \mathbb{Z}$,

$$z[mL] = \sum_k x[k]\mathrm{sinc}\left(\frac{mL - kL}{L}\right) = x[m].$$

But it also obtains all the samples in the middle.

Example 11.12 *Look at Fig. 11.21, wherein the sampling period has been changed from T_s to $T_s' = \frac{T_s}{L}$.*

Example 11.13 $$x_c(t) = \cos(4000\pi t)$$

$$T_s = \frac{1}{6000} \quad , \quad x[n] = x_c(nT_s) \quad , \quad \Omega_s = 2\pi \times 6000 = 12000\pi$$

$$x[n] = \cos\left(4000\pi \frac{n}{6000}\right) = \cos\left(\frac{2\pi}{3}n\right)$$

$$y[n] = \mathrm{U}_2\left(x[n]\right) = \begin{cases} x\left[n/2\right] & n = 0, \pm 2, \pm 4, \ldots \\ 0 & \textit{else.} \end{cases}$$

$$T_s{}' = \frac{1}{12000} \quad , \quad z[n] = x_c(nT_s{}')$$

$$z[n] = \cos\left(4000\pi \frac{n}{12000}\right) = \cos\left(\frac{\pi n}{3}\right).$$

The relationship between the spectrum of all the signals in this example is illustrated in Fig. 11.22.

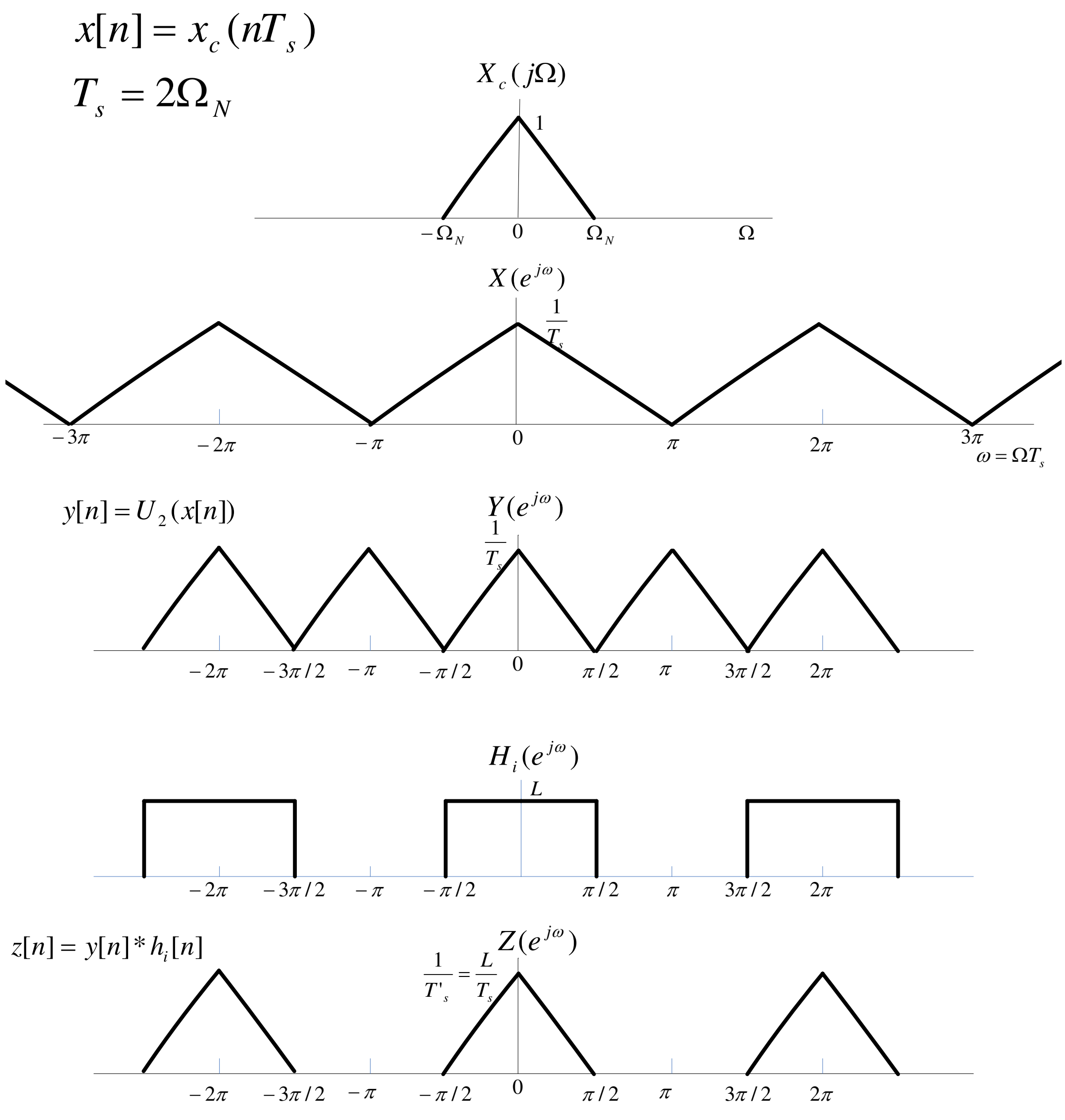


Figure 11.21: changing the up-sampling rate for $\frac{T'_s}{T_s} = \frac{1}{L}$.

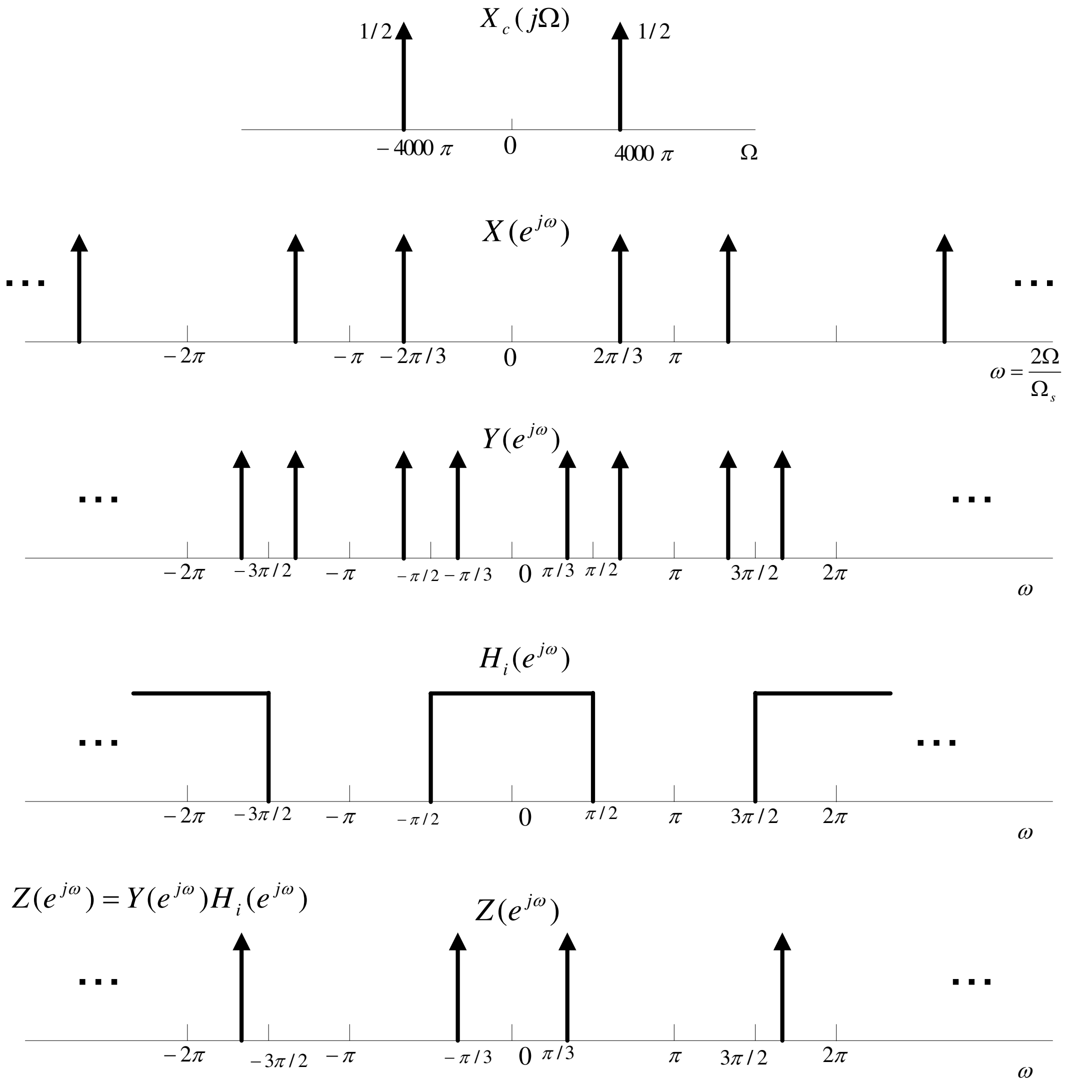


Figure 11.22: Example. 11.13

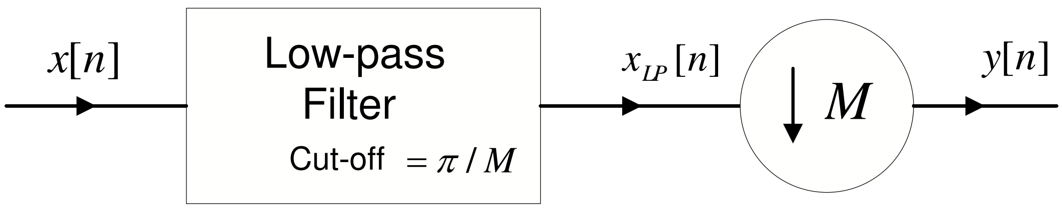


Figure 11.23: Filtering followed by downsampling.

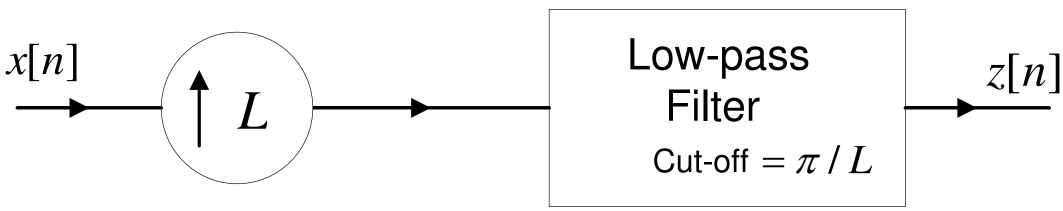


Figure 11.24: Upsampling followed by interpolation.

11.4 Changing Sampling Rate by a Rational Non-integer Factor

We have shown so far how to increase or decrease the sampling rate of a sequence by an integer factor. These operations are depicted in Figures 11.23 and 11.24. By combining downsampling and upsampling, it is possible to change the sampling rate by a non-integer (rational) factor.

Specifically we can cascade an upsampling and downsampling operation to produce a rational sampling frequency change as illustrated in Figure 11.25. This can be simplified as in Figure 11.26.

Example 11.14 *Let $x_c(t) = \cos(4000\pi t)$ and $T_s = \frac{1}{6000}$. Then the sampled signal is*

$$x[n] = x_c\left(n\frac{1}{6000}\right) = \cos\left(\frac{2\pi}{3}n\right).$$

We want to resample the signal at ${T_s}' = 1/8000$, i.e., $T_s/T_s' = 8000/6000 = 4/3$. That is, we finally want,

$$u[n] = x_c(n{T_s}') = \cos\left(\frac{4000\pi n}{8000}\right) = \cos\left(\frac{\pi}{2}n\right).$$

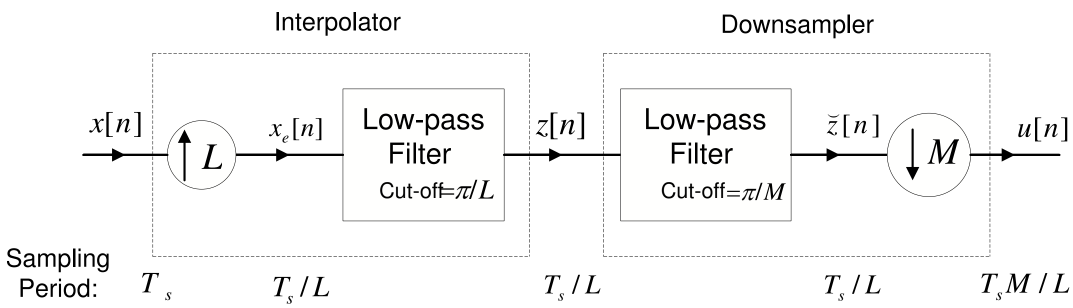


Figure 11.25: System for changing the sampling rate by a factor $\frac{M}{L}$.

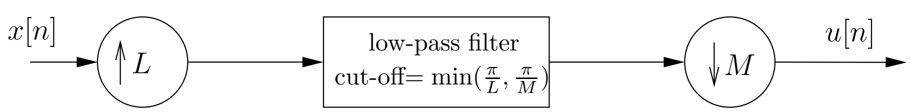


Figure 11.26: This is the same system as Figure 11.25, but simplified by combining the two low-pass filters into one.

Can we do this through discrete-time operations on $x[n]$?

Since $T_s{}' = 3/4T_s$, we have $M = 3$, $L = 4$ as down-sampling and up-sampling factors. This is shown in Figure 11.27.

Example 11.15 (Sampling rate conversion for digital audio.) *If we want to go from 44.1kHz sampling rate to 48.0kHz sampling rate we need:*

$$T_s = \frac{10^{-3}}{44.1} \qquad , \qquad T_s{}' = \frac{10^{-3}}{48}$$

$$\frac{T_s{}'}{T_s} = \frac{44.1}{48} = \frac{14.7}{16} \quad or \quad T_s{}' = T_s\left(\frac{147}{160}\right).$$

Therefore if $M = 147, L = 160$, we can go from 44.1kHz to 48kHz sampling rate.

Example 11.16 *Consider the example shown in Figure 11.28. A signal that has been sampled at the Nyquist rate is resampled by a factor $2/3$. Since the new sampling rate is lower than the sampling rate, the operation will result in losing the signal in the part of the spectrum outside the new Nyquist rate.*

11.5 Interchange of Filtering and Upsampling/Downsampling

In this section we examine an important property of the upsampling and downsampling process. Consider the two systems shown in Figure 11.29. We shall show that these two systems are equivalent, first through an example and then more formally.

Example 11.17 (Equivalence of the Systems on Figure 11.29.)

$$X_a\left(e^{j\omega}\right) = \frac{1}{2}\left\{X\left(e^{j\frac{\omega}{2}}\right) + X\left(e^{j\left(\frac{\omega}{2}-\pi\right)}\right)\right\}$$

$$Y_a\left(e^{j\omega}\right) = H\left(e^{j\omega}\right)X_a\left(e^{j\omega}\right) = \frac{1}{2}H\left(e^{j\omega}\right)\left\{X\left(e^{j\frac{\omega}{2}}\right) + X\left(e^{j\left(\frac{\omega}{2}-\pi\right)}\right)\right\}$$

$$X_b\left(e^{j\omega}\right) = H\left(e^{j2\omega}\right)X\left(e^{j\omega}\right)$$

$$\begin{aligned}
Y_b\left(e^{j\omega}\right) &= \frac{1}{2}\left\{X_b\left(e^{j\frac{\omega}{2}}\right) + X_b\left(e^{j\left(\frac{\omega}{2}-\pi\right)}\right)\right\} \\
&= \frac{1}{2}\left\{H\left(e^{j\omega}\right)X\left(e^{j\frac{\omega}{2}}\right) + H\left(e^{j\left(\frac{\omega}{2}-\pi\right)2}\right)X\left(e^{j\left(\frac{\omega}{2}-\pi\right)}\right)\right\} \\
&= \frac{1}{2}H\left(e^{j\omega}\right)\left\{X\left(e^{j\frac{\omega}{2}}\right) + X\left(e^{j\left(\frac{\omega}{2}-\pi\right)}\right)\right\}.
\end{aligned}$$

Hence we see that $Y_a\left(e^{j\omega}\right) = Y_b\left(e^{j\omega}\right)$.

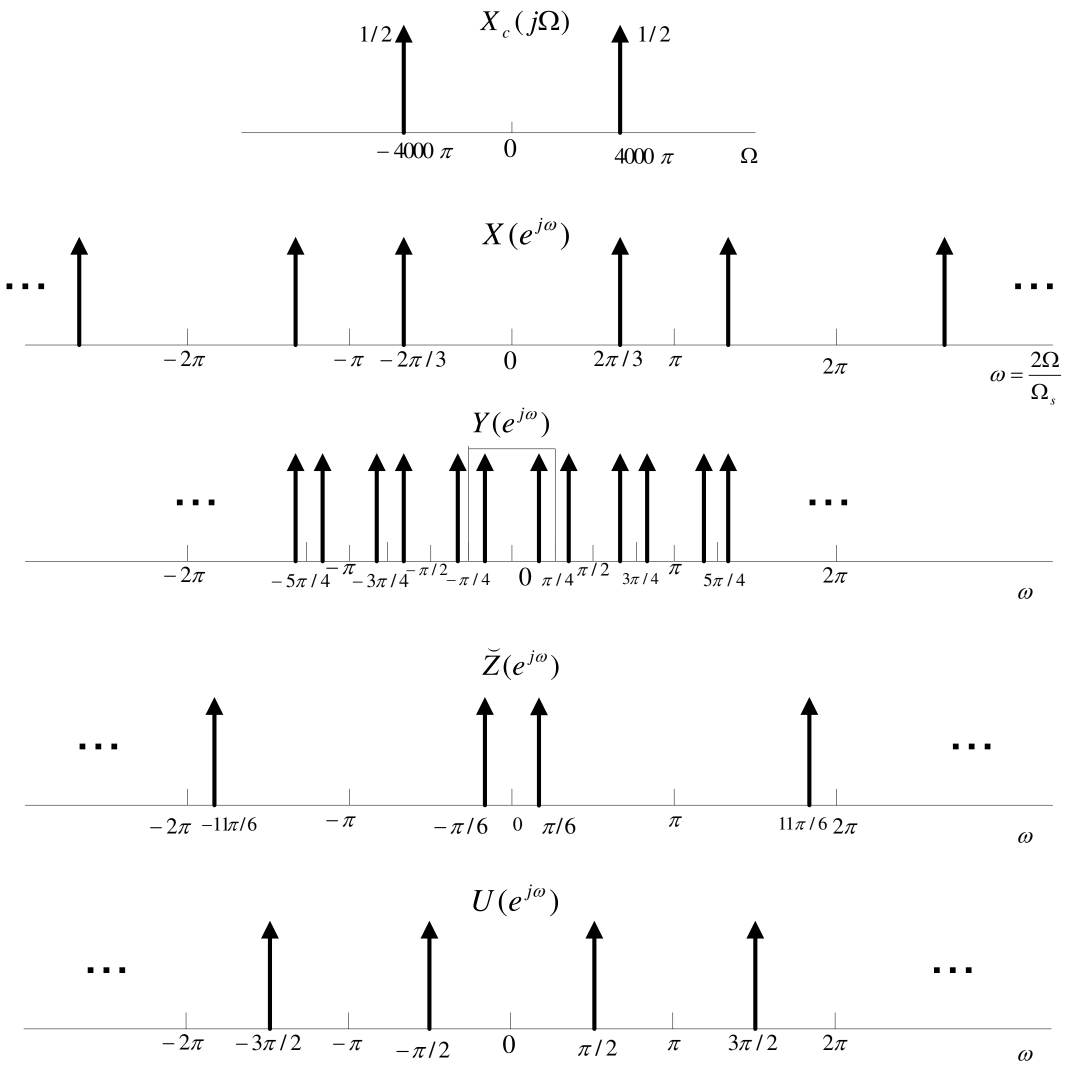


Figure 11.27: Changing the sampling rate by a rational non-integer factor.

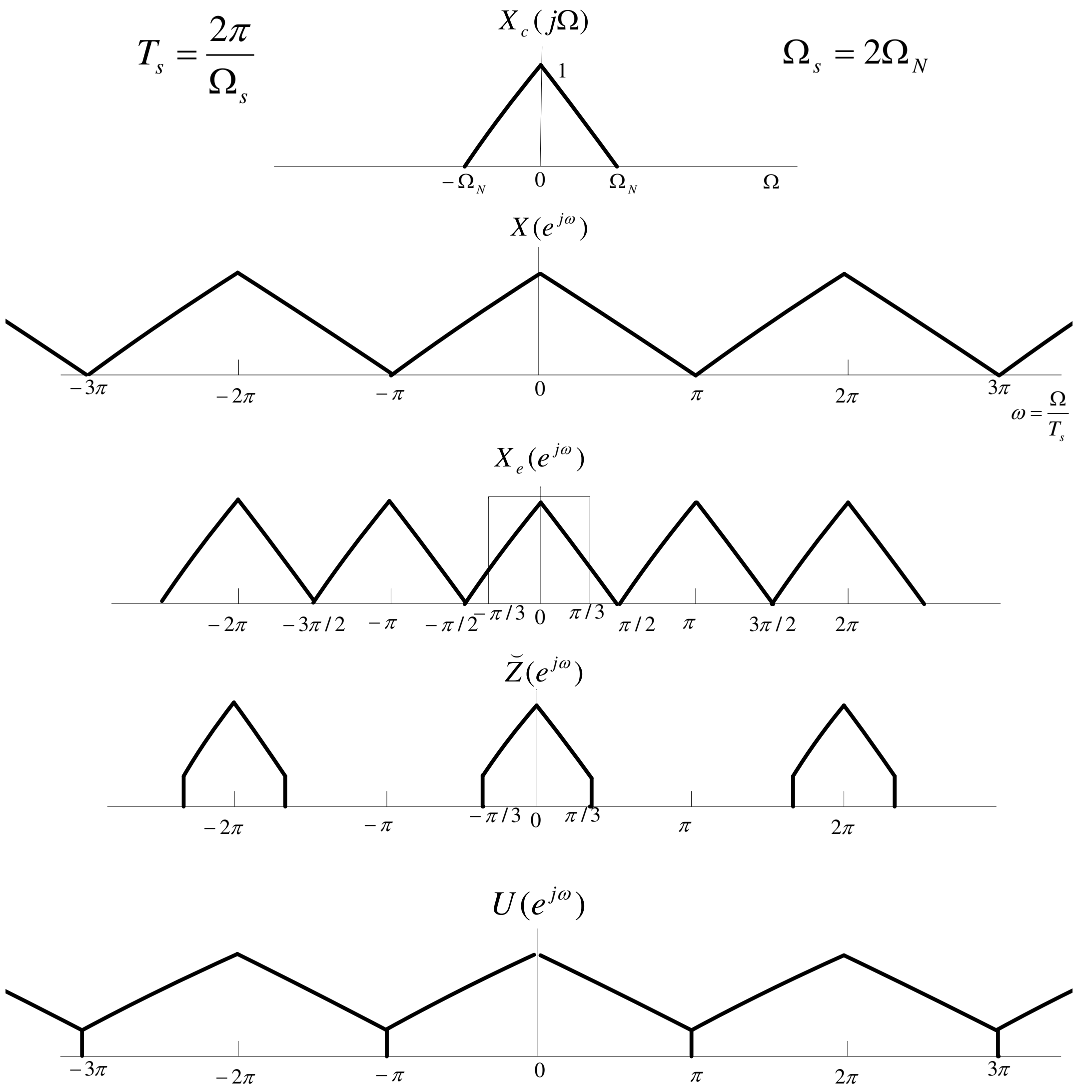


Figure 11.28: Example. 11.16: Resampling a signal to below the Nyquist rate.

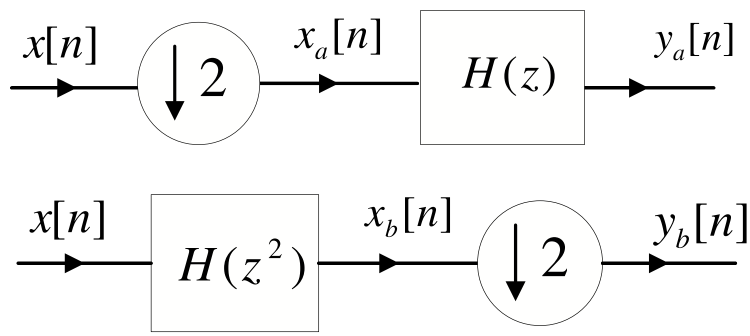


Figure 11.29: Two equivalent systems based on downsampling identities.

The example suggests the following simple formal proof for the equivalence of the two systems on Figure 11.29. We know that if

$$y[n] = \mathrm{D}_2\left(x_f[n]\right),$$

then

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

or if $|z| = 1$ is in the ROC,

$$Y\left(e^{j\omega}\right) = \frac{1}{2}\left[X_f\left(e^{j\frac{\omega}{2}}\right) + X_f\left(e^{j\left(\frac{\omega}{2}-\pi\right)}\right)\right].$$

Now since $x_a[n] = \mathrm{D}_2\left(x[n]\right)$,

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

and

$$Y_a(z) = H(z)X_a(z) = \frac{1}{2}H(z)\left\{X\left(z^{1/2}\right) + X\left(-z^{1/2}\right)\right\}. \tag{11.10}$$

Next,

$$X_b(z) = H\left(z^2\right)X(z),$$

therefore,

$$X_b\left(z^{1/2}\right) = H\left(\left(z^{1/2}\right)^2\right)X\left(z^{1/2}\right) = H(z)X\left(z^{1/2}\right),$$

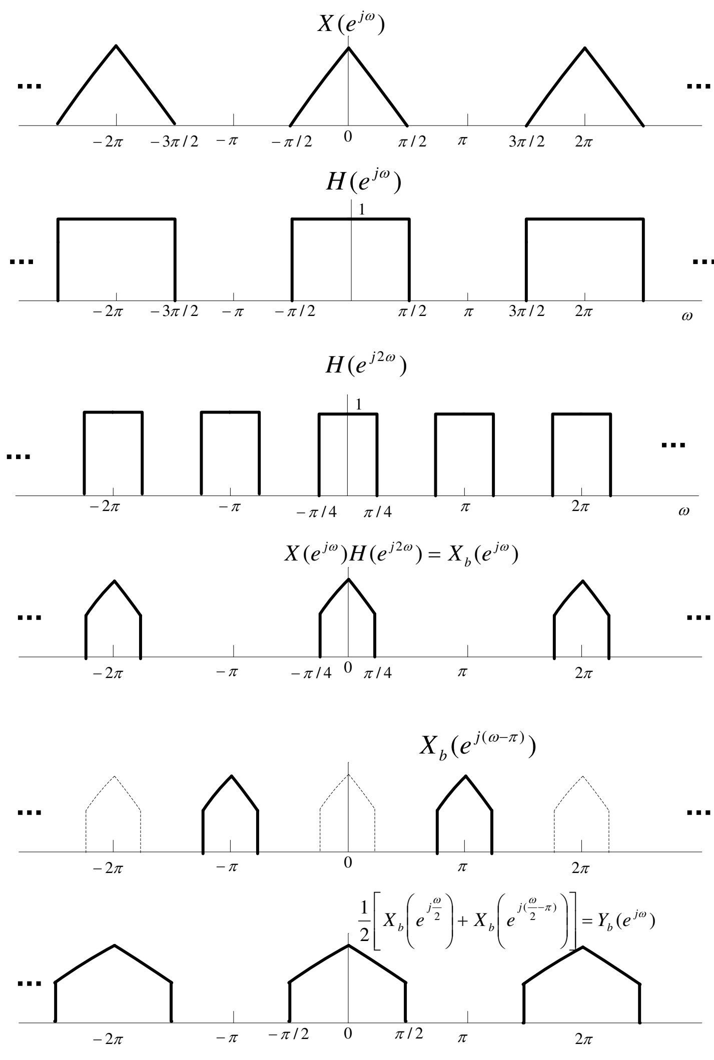


Figure 11.30: Example 11.17: Comparison of the spectrun of two equivalence systems.

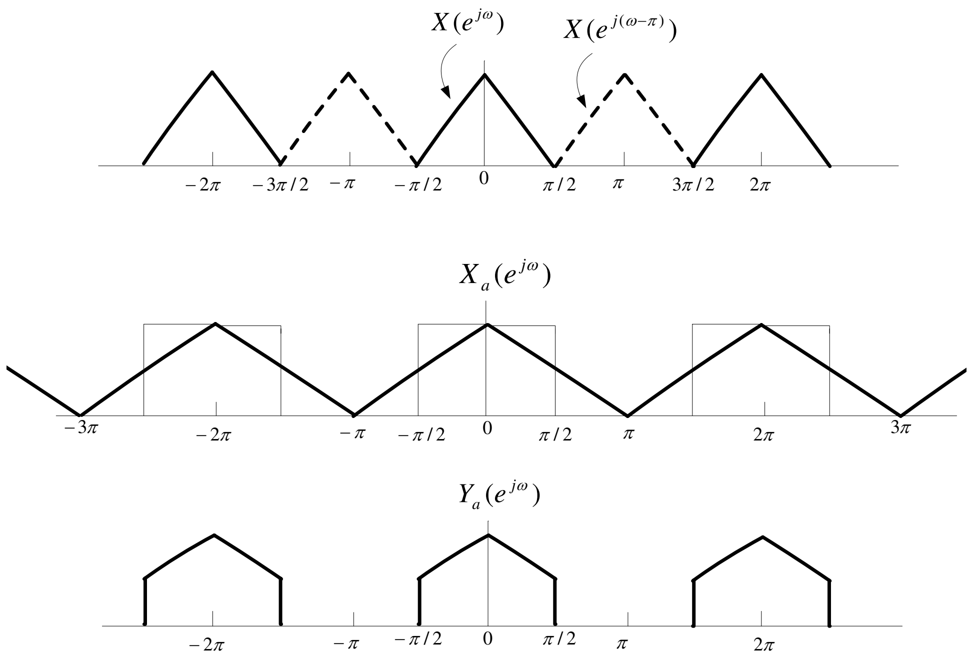


Figure 11.31: Continuation of Fig. 11.30 for Example 11.17.

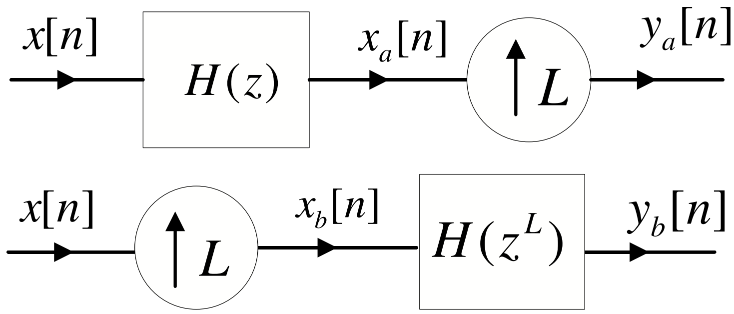


Figure 11.32: Two equivalent systems based on upsampling identities.

and

$$X_b\left(-z^{1/2}\right) = H\left(\left(-z^{1/2}\right)^2\right) X\left(-z^{1/2}\right) = H(z) X\left(-z^{1/2}\right).$$

Hence,

$$\begin{aligned} Y_b(z) &= \frac{1}{2}\left\{X_b\left(z^{1/2}\right) + X_b\left(-z^{1/2}\right)\right\} \\ &= \frac{1}{2} H(z) \left\{X\left(z^{1/2}\right) + X\left(-z^{1/2}\right)\right\} \\ &= Y_a(z). \end{aligned}$$

A similar identity applies to up-sampling as shown in Figure 11.32. This can be seen from the following.

$$Y_a(z) = X_a\left(z^L\right) = H\left(z^L\right) X\left(z^L\right),$$

and

$$Y_b(z) = H\left(z^L\right) X_b(z) = H\left(z^L\right) X\left(z^L\right).$$

Hence

$$Y_a(z) = Y_b(z),$$

and since the ROC's are the same,

$$y_a[n] = y_b[n].$$

The identities depicted on Figures 11.29 and 11.32 are called the "Noble Identities".

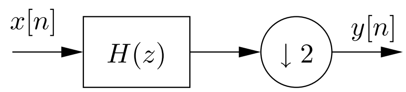


Figure 11.33: Downsampling system for Exercise 11.18.

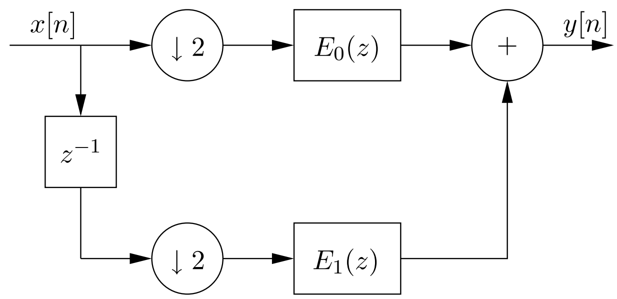


Figure 11.34: Equivalent system.

Example 11.18 (Polyphase Implementation of Downsampling) *Consider the downsampling system given in Figure 11.33, where $H(z)$ is an arbitrary filter with impulse response $h[n]$. We define*

$$e_0[n] = h[2n], \qquad \textit{and} \qquad e_1[n] = h[2n+1].$$

Prove that the system of Figure 11.34 is equivalent to the one given in Figure 11.33.

Solution:

Using one of the noble identities on Figure 11.34 results in Figure 11.35. Downsampling with a factor two is linear, so we can perform it after the addition. Now define $g_0[n]$ to be the impulse response corresponding to $E_0(z^2)$. We have

$$g_0[n] = \begin{cases} h[n], & n \text{ even}, \\ 0, & n \text{ odd}. \end{cases}$$

Similarly, defining $g_1[n]$ to be the impulse response corresponding to $E_1(z^2)$, we have

$$g_1[n] = \begin{cases} h[n+1], & n \text{ even}, \\ 0, & n \text{ odd}. \end{cases}$$

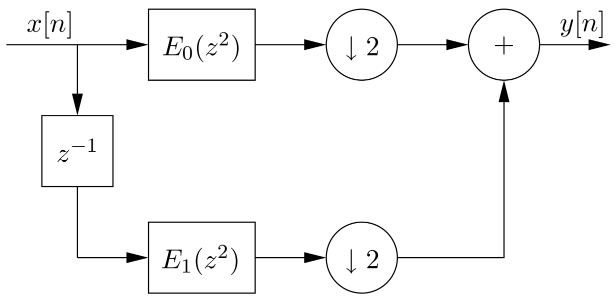


Figure 11.35: Equivalent system.

Now if we consider the filtering operation without downsampling, we have

$$\begin{aligned} y[n] &= \sum_{l=-\infty}^{\infty} h[l]x[n-l] \\ &= \sum_{k=-\infty}^{\infty} h[2k]x[n-2k] + \sum_{k=-\infty}^{\infty} h[2k+1]x[n-2k-1], \end{aligned}$$

which we see is exactly the operation performed by the system in Figure 11.35.

11.6 Sub-Band Decompositions

As we have seen, it is possible to change the sampling rate of a discrete-time signal by a combination of upsampling (with interpolation) and downsampling. Multirate techniques refer in general to utilizing upsampling, downsampling and filtering in a variety of ways to analyze and process a discrete-time signal. Filterbanks take a discrete-time signal and pass then through a parallel set *bank* of filtering and multirate operations. These ideas have become the corner-stone of modern signal processing techniques.

In digital audio (for example MP3), the representation of an audio signal is based on human auditory perception. The goal is to encode the audio signal in a manner that is *transparent* perceptually, i.e., to avoid annoying auditory effects. This is where the model of the auditory system as a *filterbank* is critically used. This also motivates the study of filterbanks, which are central to multirate signal processing.

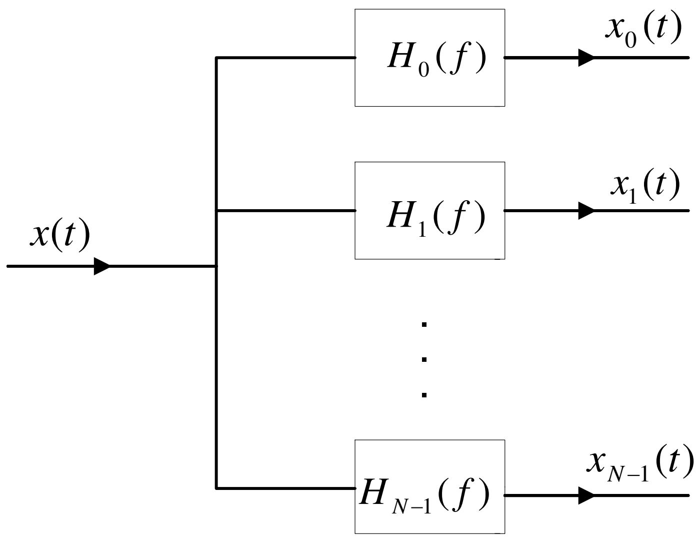


Figure 11.36: Filterbank model of human auditory system

11.6.1 Perceptual Models

As a first approximation, the human auditory systems analyzes the sounds by passing it through a bank of filters as shown in Fig. 11.36.

The auditory filters have been empirically (experimentally) characterized and are illustrated (approximately) in Fig. 11.37.

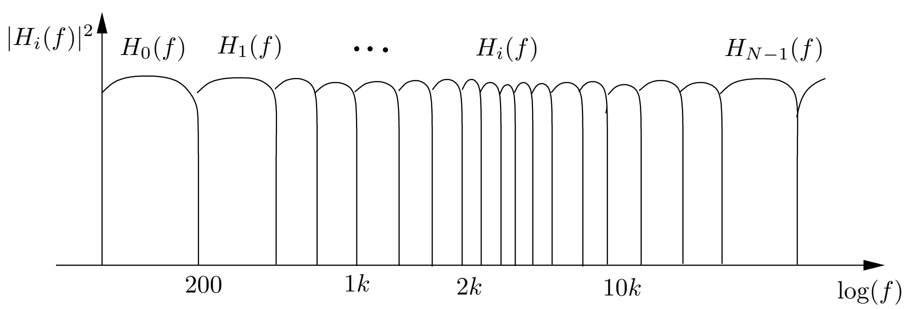


Figure 11.37: Approximate frequency responses of auditory filters

The human auditory system is able to distinguish signals across these filters but not within. That is, the output of each filter is a single entity with the dominant (highest

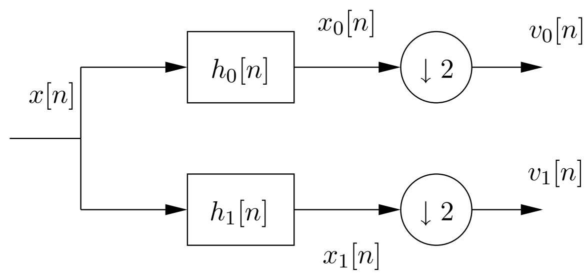


Figure 11.38: Analysis filter bank for Exercise 11.19.

energy) signal within that band masking all other signals in the same band. This masking process is quite complicated and is still not completely understood, but the basic principle of filterbanks have been very successfully used in digital audio coding. Therefore we will study the basic principles of filterbanks and in this sampling rate conversions are crucial. These ideas also form the basis for other signal representation and transform techniques such as *wavelets* which is a topic for an advanced class on signal processing.

Example 11.19 (Haar Decomposition) *In this example we consider a filter bank where the analysis and synthesis filters are not perfect low and high-pass filters.*

We start with the 2 component analysis filter bank given in Figure 11.38. The impulse responses of the filters $h_0[n]$ and $h_1[n]$, in this case are given by

$$h_0[n] = \begin{cases} 1, & \text{if } n = 0, 1 \\ 0, & \text{otherwise} \end{cases}, \qquad h_1[n] = \begin{cases} -1, & \text{if } n = 0 \\ 1, & \text{if } n = 1 \\ 0, & \text{otherwise.} \end{cases}$$

(a) *Consider*

$$\begin{bmatrix} v_0[n] \\ v_1[n] \end{bmatrix} = A \begin{bmatrix} x[2n-1] \\ x[2n] \end{bmatrix}.$$

Find A. Show that it is orthonormal.

(b) *Show that $x[2n-1]$ and $x[2n]$ can be recovered from $v_0[n]$ and $v_1[n]$.*

In the second part of this exercise we will find the synthesis filter bank that reconstructs $x[n]$ from $v_0[n]$ and $v_1[n]$. The structure of the synthesis filter bank is given in Figure 11.39. Let

$$g_0[n] = \frac{1}{2} h_0[-n+1], \qquad \text{and } g_1[n] = \frac{1}{2} h_1[-n+1].$$

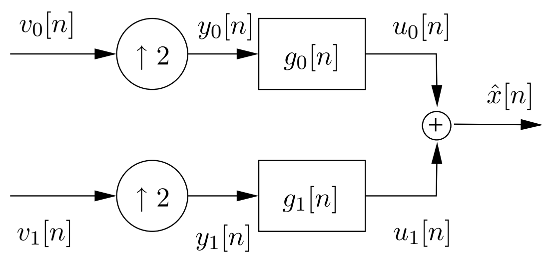


Figure 11.39: Synthesis filter bank for Example 11.19.

(c) *Find $u_0[n]$ and $u_1[n]$ and prove that $\hat{x}[n] = x[n-1]$.*

Solution:

(a)

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

The rows of A are orthogonal, since

$$\left\langle \begin{bmatrix} 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 \end{bmatrix} \right\rangle = 0.$$

(b) *Since the rows of A are orthogonal it is full rank and hence invertible. We have*

$$\begin{bmatrix} x[2n-1] \\ x[2n] \end{bmatrix} = A^{-1} \begin{bmatrix} v_0[n] \\ v_1[n] \end{bmatrix},$$

where

$$A^{-1} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Let

$$g_0[n] = \frac{1}{2} h_0[-n+1], \qquad \text{and } g_1[n] = \frac{1}{2} h_1[-n+1].$$

(c) *We have*

$$\begin{aligned} v_0[n] &= x[2n] + x[2n-1] \\ v_1[n] &= -x[2n] + x[2n-1]. \end{aligned}$$

$$g_0[n] = \frac{1}{2} h_0[-n+1] = \begin{cases} \frac{1}{2}, & n = 0 \\ \frac{1}{2}, & n = 1 \\ 0, & \text{otherwise}, \end{cases}$$

$$g_1[n] = \frac{1}{2} h_1[-n+1] = \begin{cases} \frac{1}{2}, & n = 0 \\ -\frac{1}{2}, & n = 1 \\ 0, & \text{otherwise}. \end{cases}$$

$$y_0[n] = \begin{cases} x[n] + x[n-1], & n \text{ even} \\ 0, & n \text{ odd}, \end{cases}$$

$$y_1[n] = \begin{cases} -x[n] + x[n-1], & n \text{ even} \\ 0, & n \text{ odd}. \end{cases}$$

$$u_0[n] = \frac{1}{2}\left(y_0[n] + y_0[n-1]\right) = \begin{cases} \frac{1}{2}\left(x[n] + x[n-1]\right), & n \textit{ even} \\ \frac{1}{2}\left(x[n-1] + x[n-2]\right), & n \textit{ odd}, \end{cases}$$

$$u_1[n] = \frac{1}{2}\left(y_1[n] - y_1[n-1]\right) = \begin{cases} \frac{1}{2}\left(-x[n] + x[n-1]\right), & n \textit{ even} \\ \frac{1}{2}\left(x[n-1] - x[n-2]\right), & n \textit{ odd}, \end{cases}$$

$$\hat{x}[n] = \begin{cases} x[n-1], & n \textit{ even} \\ x[n-1], & n \textit{ odd}. \end{cases}$$

11.7 Sub-band or filterbank decomposition of signals

As seen in the auditory perceptual model, there is value to analyzing and representing signals using a bank of filters. If the filter have (almost) non-overlapping responses (such as a low-pass and high-pass), then we are splitting the signals into frequency sub-bands and hence the terminology sub-band decomposition.

A basic structure for a sub-band or filterbank decomposition is illustrated in Fig. 11.40.

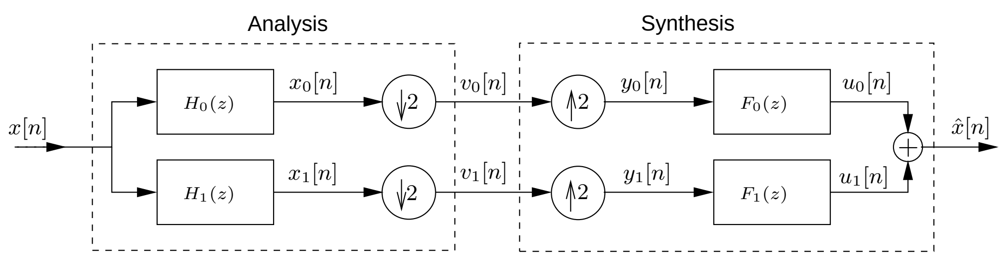


Figure 11.40: Analysis and synthesis of sub-band decomposition. This is also called the two-band quadrature mirror filter (QMF) bank.

Example 11.20 *In Figures 11.41, 11.42, 11.43, and 11.44, you find an example of sub-band decomposition.*

Hence, we see that this filterbank reconstructs $x[n]$ after the synthesis. Such a filterbank which synthesizes the original signal $x[n]$ (except perhaps sealing and delaying) is called a *perfect reconstruction filterbank*. In this example we achieved this by having *non-overlapping* frequency responses for the *analysis* filters $(H_0(z), H_1(z))$ and the synthesis filters $(F_0(z), F_1(z))$. However, such a perfect reconstruction also holds (surprisingly)

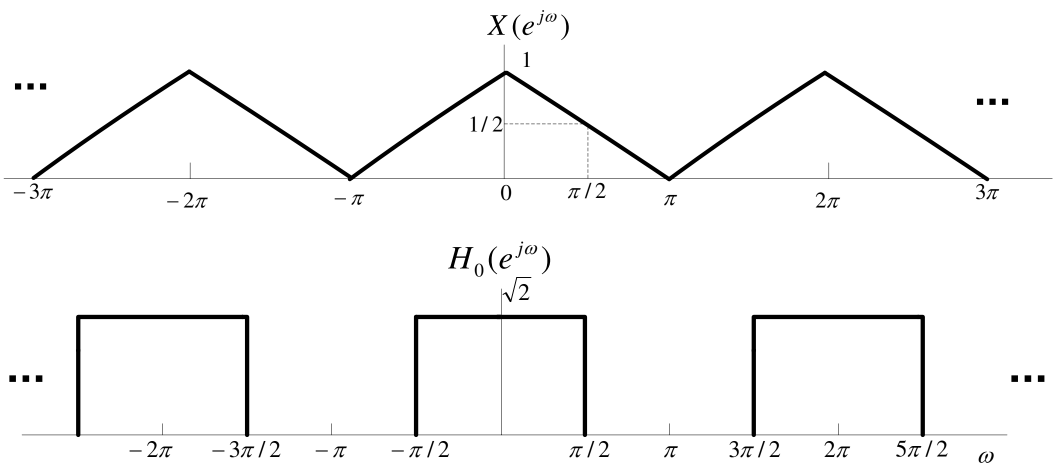


Figure 11.41: Example 11.20.

for carefully designed filters $H_0(z), H_1(z), F_0(z), F_1(z)$ which might have *overlapping* frequency responses.

To understand the implications of this example more thoroughly, notice that $v_0[n]$ and $v_1[n]$ were sufficient to reconstruct $x[n]$. Therefore, inherently we are *representing* $x[n]$ through $v_0[n]$ and $v_1[n]$. This is because $v_0[n]$ and $v_1[n]$ are downsampled versions of $x_0[n]$ and $x_1[n]$ and hence together have the same number of samples as $x[n]$.

Since $H_0\left(e^{j\omega}\right)$ and $H_1\left(e^{j\omega}\right)$ are orthogonal and span the whole space (i.e. cover all the frequencies), we can easily understand that $x_0[n]$ and $x_1[n]$ allow us to reconstruct $x[n]$. However, the non-trivial observation is that downsampled versions of $x_0[n]$ and $x_1[n]$ i.e. $v_0[n]$, $v_1[n]$ are also sufficient for reconstructing $x[n]$.

Since

$$\hat{X}\left(e^{j\omega}\right) = F_0\left(e^{j\omega}\right) Y_0\left(e^{j\omega}\right) + F_1\left(e^{j\omega}\right) Y_1\left(e^{j\omega}\right),$$

and

$$\hat{X}\left(e^{j\omega}\right) = X\left(e^{j\omega}\right),$$

we see that the decomposition is

$$X\left(e^{j\omega}\right) = F_0\left(e^{j\omega}\right) V_0\left(e^{j2\omega}\right) + F_1\left(e^{j\omega}\right) V_1\left(e^{j2\omega}\right).$$

and

$$\begin{aligned}\hat{X}\left(e^{j\omega}\right) &= F_0\left(e^{j\omega}\right)Y_0\left(e^{j\omega}\right)+F_1\left(e^{j\omega}\right)Y_1\left(e^{j\omega}\right)\\ &= F_0\left(e^{j\omega}\right)V_0\left(e^{j2\omega}\right)+F_1\left(e^{j\omega}\right)V_1\left(e^{j2\omega}\right).\end{aligned}$$

Now,

$$\begin{aligned}V_0\left(e^{j\omega}\right) &= \frac{1}{2}\left\{X_0\left(e^{j\omega/2}\right)+X_0\left(e^{j\left(\frac{\omega}{2}-\pi\right)}\right)\right\}\\ &= \frac{1}{2}\left\{X_0\left(e^{j\omega/2}\right)+X_0\left(-e^{j\omega/2}\right)\right\}\end{aligned}$$

Hence,

$$\begin{aligned}V_0\left(e^{j2\omega}\right) &= \frac{1}{2}\left\{X_0\left(e^{j\omega}\right)+X_0\left(-e^{j\omega}\right)\right\}\\ &= \frac{1}{2}\left\{H_0\left(e^{j\omega}\right)X\left(e^{j\omega}\right)+H_0\left(-e^{j\omega}\right)X\left(-e^{j\omega}\right)\right\}\end{aligned}$$

and similarly,

$$\begin{aligned}V_1\left(e^{j2\omega}\right) &= \frac{1}{2}\left\{X_1\left(e^{j\omega}\right)+X_1\left(-e^{j\omega}\right)\right\}\\ &= \frac{1}{2}\left\{H_1\left(e^{j\omega}\right)X\left(e^{j\omega}\right)+H_1\left(-e^{j\omega}\right)X\left(-e^{j\omega}\right)\right\}\end{aligned}$$

Therefore,

$$\begin{aligned}\hat{X}\left(e^{j\omega}\right) &= \frac{1}{2}F_0\left(e^{j\omega}\right)H_0\left(e^{j\omega}\right)X\left(e^{j\omega}\right)+\frac{1}{2}F_0\left(e^{j\omega}\right)H_0\left(e^{j(\omega-\pi)}\right)X\left(e^{j(\omega-\pi)}\right)\\ &+ \frac{1}{2}F_1\left(e^{j\omega}\right)H_1\left(e^{j\omega}\right)X\left(e^{j\omega}\right)+\frac{1}{2}F_1\left(e^{j\omega}\right)H_1\left(e^{j(\omega-\pi)}\right)X\left(e^{j(\omega-\pi)}\right).\end{aligned}$$

But $F_0\left(e^{j\omega}\right)H_0\left(e^{j(\omega-\pi)}\right)=0=F_1\left(e^{j\omega}\right)H_1\left(e^{j(\omega-\pi)}\right)$ since

$$F_0\left(e^{j\omega}\right)=H_0\left(e^{j\omega}\right)=\begin{cases}\sqrt{2} & |\omega|\le\pi/2\\ 0 & \text{else}\end{cases}$$

and

$$F_1\left(e^{j\omega}\right)=H_1\left(e^{j\omega}\right)=\begin{cases}\sqrt{2} & |\omega|>\pi/2\\ 0 & \text{else.}\end{cases}\quad .$$

Therefore,

$$\begin{aligned}\hat{X}\left(e^{j\omega}\right) &= \frac{1}{2}\left|H_0\left(e^{j\omega}\right)\right|^2X\left(e^{j\omega}\right)+\frac{1}{2}\left|H_1\left(e^{j\omega}\right)\right|^2X\left(e^{j\omega}\right)\\ &= \frac{1}{2}\left\{\left|H_0\left(e^{j\omega}\right)\right|^2+\left|H_1\left(e^{j\omega}\right)\right|^2\right\}X\left(e^{j\omega}\right)=X\left(e^{j\omega}\right).\end{aligned}$$

Note that in the example shown above, the analysis and synthesis filters were *ideal*, i.e. perfect low-pass and high-pass filters. This means that we need infinite length filters to implement such a system. A natural question to ask is whether we can obtain such a property by using *finite-length* filters. At first, this seems impossible since finite-length filters will definitely have overlapping spectra. However, this is possible and we illustrate this idea using a signal decomposition method developed by Haar in 1917, which was largely forgotten till a couple of decades ago.

We will look at the Haar signal decomposition in two ways. First through a filter bank and second through a basis expansion.

Consider the filter $h_0[n]$ with impulse response

$$h_0[n] = \begin{cases} \frac{1}{\sqrt{2}} & n = -1, 0 \\ 0 & \text{else.} \end{cases} \tag{11.11}$$

Note that such a filter is non-causal, but has a finite length and therefore can be implemented using a delay. Consider, as in Figure 11.45,

$$\begin{aligned} x_0[n] &= h_0[n] \star x[n] = h_0[0]x[n] + h_0[1]x[n+1] \\ &= \frac{1}{\sqrt{2}} \{x[n] + x[n+1]\}. \end{aligned} \tag{11.12}$$

Now consider the filter $h_1[n]$ with impulse response,

$$h_1[n] = \begin{cases} \frac{1}{\sqrt{2}} & n = 0 \\ -\frac{1}{\sqrt{2}} & n = -1 \\ 0 & \text{else.} \end{cases} \tag{11.13}$$

Again, such a filter is non-causal but FIR and hence implementable. We therefore obtain,

$$x_1[n] = h_1[n] \star x[n] = \frac{1}{\sqrt{2}} \{x[n] - x[n-1]\} \tag{11.14}$$

Observing that $v_0[n]$ and $v_1[n]$ are downsampled versions of $x_0[n]$ and $x_1[n]$ respectively (see Figure 11.45), we get

$$\begin{aligned} v_0[n] &= x_0[2n] = \frac{1}{\sqrt{2}} \{x[2n] + x[2n+1]\} \\ v_1[n] &= x_1[2n] = \frac{1}{\sqrt{2}} \{x[2n] - x[2n+1]\}. \end{aligned} \tag{11.15}$$

Now it is easy to see from (11.15) that $\{v_0[n]\}$ and $\{v_1[n]\}$ sequences are enough to retain *all* the information about $x[n]$ since we see

$$\begin{aligned} v_0[n] + v_1[n] &= \sqrt{2}x[2n] \\ v_0[n] - v_1[n] &= \sqrt{2}x[2n+1]. \end{aligned} \tag{11.16}$$

This observation implies that even by filtering through filters $H_0\left(e^{j\omega}\right)$ and $H_1\left(e^{j\omega}\right)$ with *overlapping* spectra and downsampling, we still retain all the information about the input sequence.

This picture can be completed formally by constructing the corresponding synthesis filter $g_0[n]$ and $g_1[n]$.

$$g_0[n] = h_0[-n] = \begin{cases} \frac{1}{\sqrt{2}} & n = 0, 1 \\ 0 & \text{else.} \end{cases} \tag{11.17}$$

$$g_1[n] = h_1[-n] = \begin{cases} \frac{1}{\sqrt{2}} & n = 0 \\ -\frac{1}{\sqrt{2}} & n = 1 \\ 0 & \text{else,} \end{cases} \tag{11.18}$$

which are now *causal* FIR filters.

Now we have (see Figure 11.45),

$$y_0[n] = \begin{cases} v_0\left[n/2\right], & n \text{ even} \\ 0, & \text{else} \end{cases} = \begin{cases} \frac{1}{\sqrt{2}}\left(x[n] + x[n+1]\right), & n \text{ even} \\ 0, & \text{else.} \end{cases}$$

$$y_1[n] = \begin{cases} v_1\left[n/2\right], & n \text{ even} \\ 0, & \text{else} \end{cases} = \begin{cases} \frac{1}{\sqrt{2}}\left(x[n] - x[n+1]\right), & n \text{ even} \\ 0, & \text{else.} \end{cases} \tag{11.19}$$

Therefore, we get using (11.17) and (11.18) in (11.19),

$$\begin{aligned} u_0[n] &= \frac{1}{\sqrt{2}}\left(y_0[n] + y_0[n-1]\right) = \begin{cases} \frac{1}{\sqrt{2}} y_0[n], & n \text{ even} \\ \frac{1}{\sqrt{2}} y_0[n-1], & n \text{ odd} \end{cases} \\ u_1[n] &= \frac{1}{\sqrt{2}}\left(y_1[n] + y_1[n-1]\right) = \begin{cases} \frac{1}{\sqrt{2}} y_1[n], & n \text{ even} \\ -\frac{1}{\sqrt{2}} y_1[n-1], & n \text{ odd} \end{cases} \end{aligned} \tag{11.20}$$

yielding

$$\begin{aligned} u_0[n] &= \begin{cases} \frac{1}{2}\left(x[n] + x[n+1]\right), & n \text{ even} \\ \frac{1}{2}\left(x[n-1] + x[n]\right), & n \text{ odd} \end{cases} \\ u_1[n] &= \begin{cases} \frac{1}{2}\left(x[n] - x[n+1]\right), & n \text{ even} \\ \frac{1}{2}\left(x[n] - x[n-1]\right), & n \text{ odd.} \end{cases} \end{aligned} \tag{11.21}$$

Therefore, we get for $\hat{x}[n]$ (see Figure 11.45),

$$\hat{x}[n] = u_0[n] + u_1[n] = \begin{cases} x[n] & n \text{ even} \\ x[n] & n \text{ odd} \end{cases} = x[n]. \tag{11.22}$$

Therefore we get perfect reconstruction.

This illustrates the fact that we can get perfect reconstruction even with analysis (and synthesis) filters with finite length. This immediately means that the spectra of the filters $H_0\left(e^{j\omega}\right)$ and $H_1\left(e^{j\omega}\right)$ are overlapping as shown in Figure 11.48.

$$
\begin{aligned}
H_0\left(e^{j\omega}\right) &= \frac{1}{\sqrt{2}}\left(1+e^{-j\omega}\right) \\
H_1\left(e^{j\omega}\right) &= \frac{1}{\sqrt{2}}\left(1-e^{-j\omega}\right) = \frac{1}{\sqrt{2}}\left(1+e^{-j(\omega-\pi)}\right) \\
\left|H_0\left(e^{j\omega}\right)\right| &= \left|H_0\left(e^{j(\omega-\pi)}\right)\right| = [1+\cos\omega]^{1/2}
\end{aligned}
$$

An alternate interpretation of this property arises by viewing this through basis functions. This viewpoint is quite general and powerful since it leads to the idea of wavelets of which the Haar decomposition is a simple example.

Consider Haar *basis* functions as,

$$
\varphi_{2k}[n] = \begin{cases} \frac{1}{\sqrt{2}} & n = 2k, 2k+1 \\ 0 & \text{else.} \end{cases} \tag{11.23}
$$

$$
\varphi_{2k+1}[n] = \begin{cases} \frac{1}{\sqrt{2}} & n = 2k \\ -\frac{1}{\sqrt{2}} & n = 2k+1 \\ 0 & \text{else.} \end{cases} \tag{11.24}
$$

We notice that

$$
\varphi_{2k}[n] = \varphi_0[n-2k] \qquad , \qquad \varphi_{2k+1}[n] = \varphi_1[n-2k].
$$

Moreover,

$$
\langle \varphi_{2k}, \varphi_{2\ell} \rangle = \sum_n \varphi_{2k}^*[n]\varphi_{2\ell}[n] = \delta[k-\ell].
$$

$$
\langle \varphi_{2k}, \varphi_{2k+1} \rangle = \sum_n \varphi_{2k}^*[n]\varphi_{2k+1}[n] = \frac{1}{\sqrt{2}}\{1-1\} = 0,
$$

and for $\ell \neq k$

$$
\langle \varphi_{2k}, \varphi_{2\ell+1} \rangle = \sum_n \varphi_{2k}^*[n]\varphi_{2\ell+1}[n] = 0.
$$

Hence we see that $\{\varphi_\ell\}_\ell$ form an orthonormal set, i.e.

$$
\langle \varphi_p, \varphi_q \rangle = \delta[p-q]. \tag{11.25}
$$

Now the Haar analysis function (see Figure 11.45) can be interpreted as a projection by noticing that,

$$v_0[k] = x_0[2k] = \langle \varphi_{2k}, x \rangle = \frac{1}{\sqrt{2}} \{x[2k] + x[2k+1]\}. \tag{11.26}$$

$$v_1[k] = x_1[2k] = \langle \varphi_{2k+1}, x \rangle = \frac{1}{\sqrt{2}} \{x[2k] - x[2k+1]\}. \tag{11.27}$$

Therefore from this point-of-view, it is not surprising that $\{v_0[k]\}$,$\{v_1[k]\}$ sequences are *equivalent* to the original sequence $x[n]$, since they are just representation of $x[n]$ in the basis $\{\varphi_\ell\}_\ell$. Therefore, the reconstruction is also quite simple,

$$\begin{aligned} x[n] &= \sum_{k\in\mathbb{Z}} x_0[2k]\varphi_{2k}[n] + \sum_{k\in\mathbb{Z}} x_1[2k]\varphi_{2k+1}[n] \\ &= \sum_{k\in\mathbb{Z}} v_0[k]\varphi_{2k}[n] + \sum_{k\in\mathbb{Z}} v_1[k]\varphi_{2k+1}[n], \end{aligned} \tag{11.28}$$

as is usual for any orthonormal basis.

We see that (11.28) is clearly equivalent to (11.22), just expressed in the form of basis functions instead of filters. More concretely we see that

$$\begin{aligned} \sum_{k\in\mathbb{Z}} v_0[k]\varphi_{2k}[n] &= \sum_{k} \frac{1}{\sqrt{2}} \{v_0[k]\delta[n-2k] + v_0[k]\delta[n-2k-1]\} \\ &= \frac{1}{\sqrt{2}} \underbrace{\sum_{k\in\mathbb{Z}} v_0[k]\delta[n-2k]}_{y_0[n]=U_2(v_0[n])} + \frac{1}{\sqrt{2}} \underbrace{\sum_{k\in\mathbb{Z}} v_0[k]\delta[n-2k-1]}_{y_0[n-1]} \\ &= \frac{1}{\sqrt{2}} \{y_0[n] + y_0[n-1]\} = u_0[n]. \end{aligned}$$

$$\begin{aligned} \sum_{k\in\mathbb{Z}} v_1[k]\varphi_{2k+1}[n] &= \frac{1}{\sqrt{2}} \underbrace{\sum_{k\in\mathbb{Z}} v_1[k]\delta[n-2k]}_{y_1[n]=U_2(v_1[n])} - \frac{1}{\sqrt{2}} \underbrace{\sum_{k\in\mathbb{Z}} v_1[k]\delta[n-2k-1]}_{y_1[n-1]} \\ &= \frac{1}{\sqrt{2}} \{y_1[n] - y_1[n-1]\} = u_1[n]. \end{aligned}$$

Therefore (11.28) just expresses the same relationship as in (11.22) as

$$\begin{aligned} x[n] &= u_0[n] + u_1[n] \\ &= \frac{1}{\sqrt{2}} \{y_0[n] + y_0[n-1]\} + \frac{1}{\sqrt{2}} \{y_1[n] - y_1[n-1]\}, \end{aligned} \tag{11.29}$$

which was done through a filterbank interpretation. Figures 11.46 and 11.47 show the synthesis operation for an example.

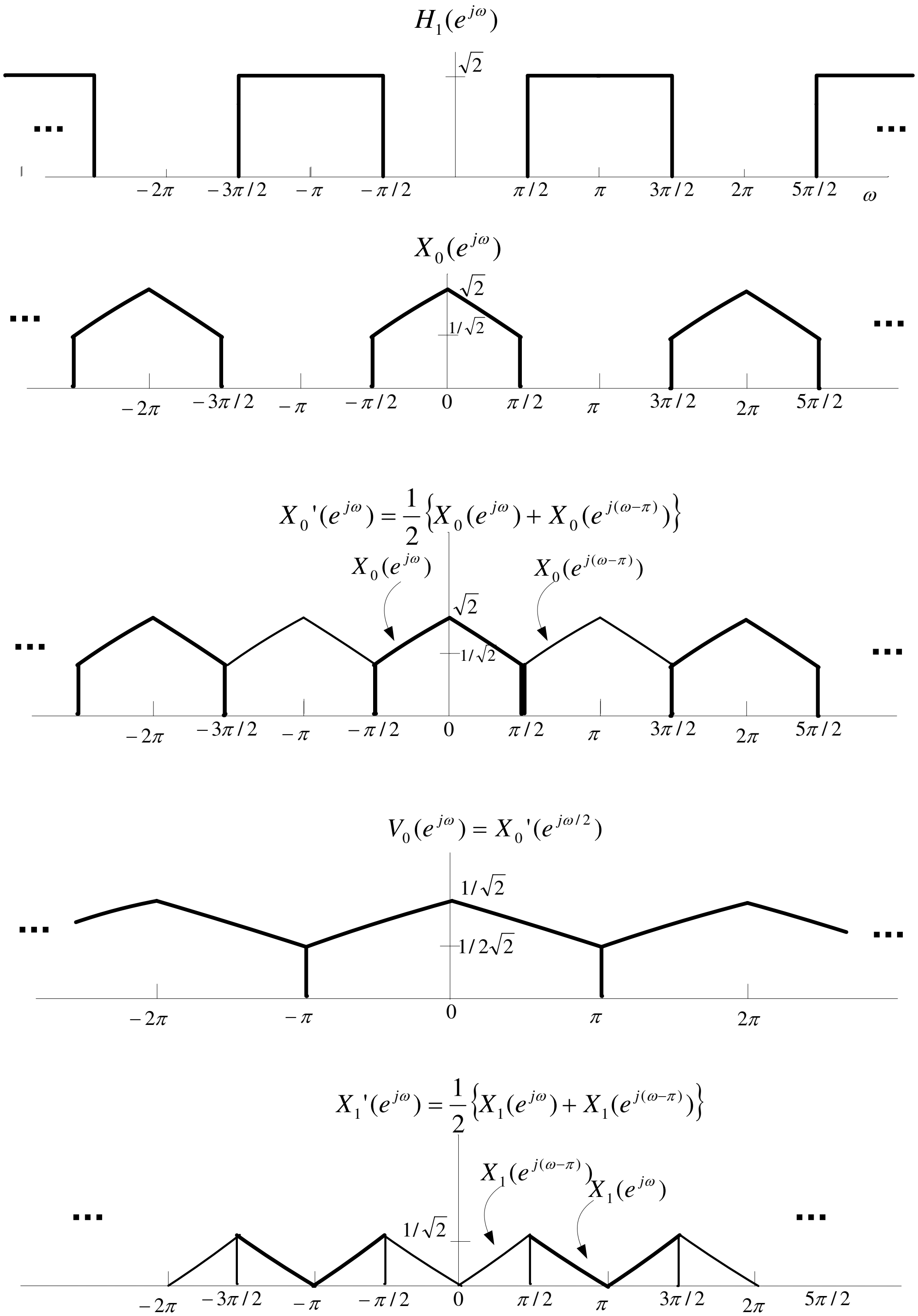


Figure 11.42: Continuation of Fig.11.41: Example 11.20.

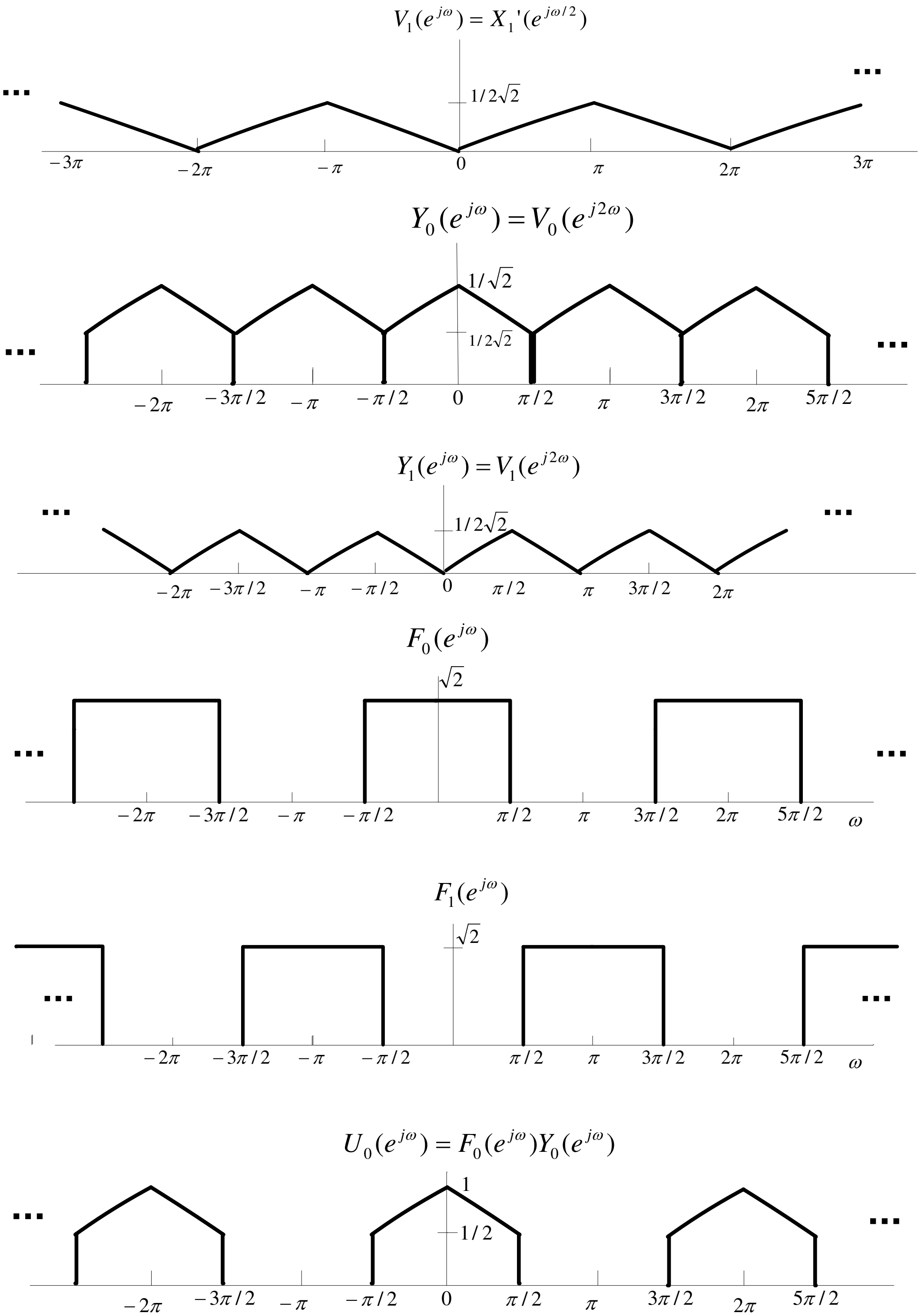


Figure 11.43: Continuation of Fig.11.42: Example 11.20.

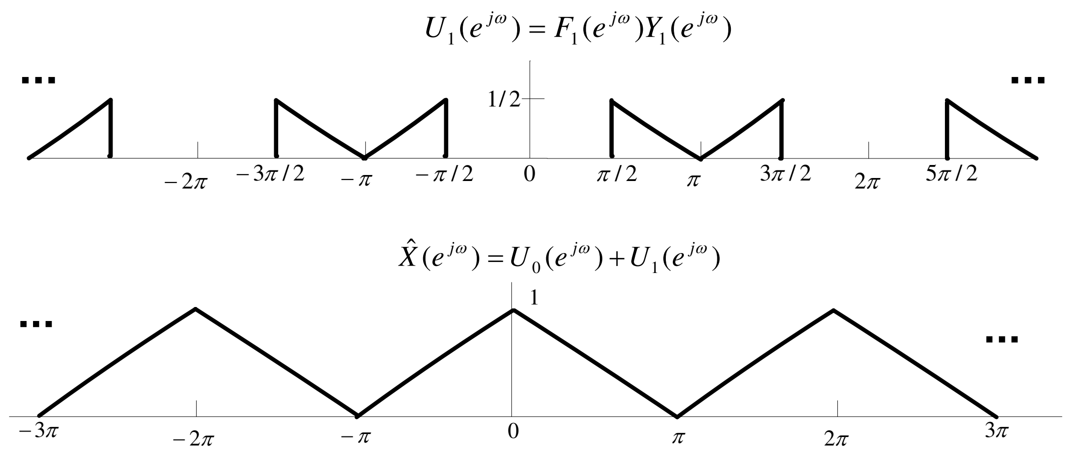


Figure 11.44: Continuation of Fig.11.43: Example 11.20.

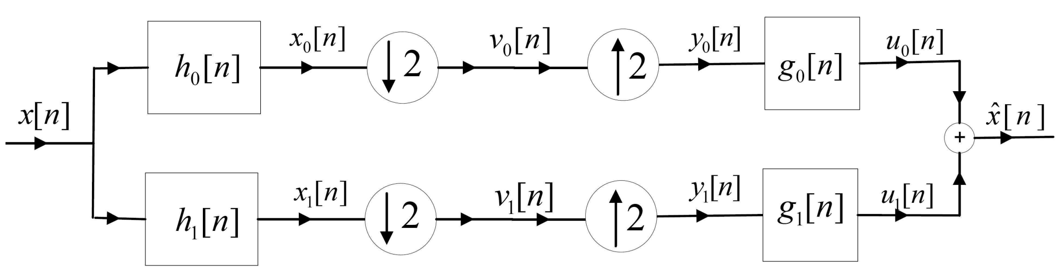


Figure 11.45: Two-channel filter bank representation of the Haar decomposition

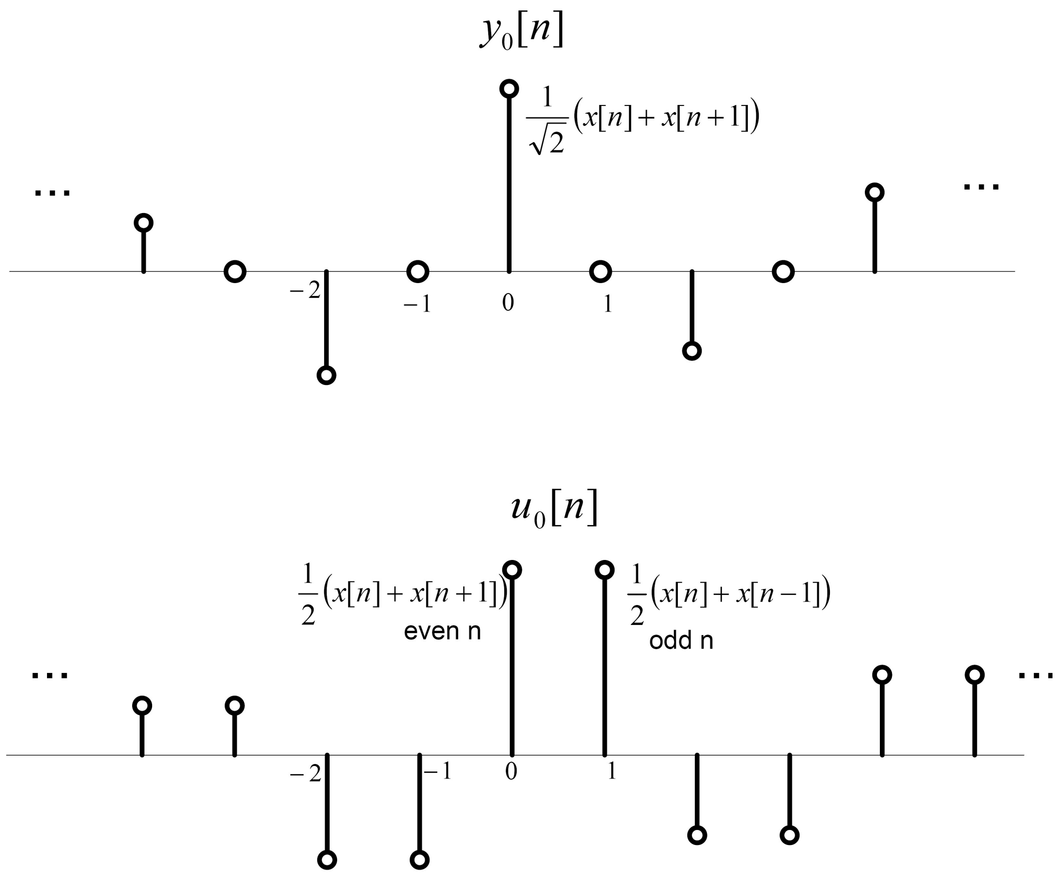


Figure 11.46: Illustration of synthesis filter operations.

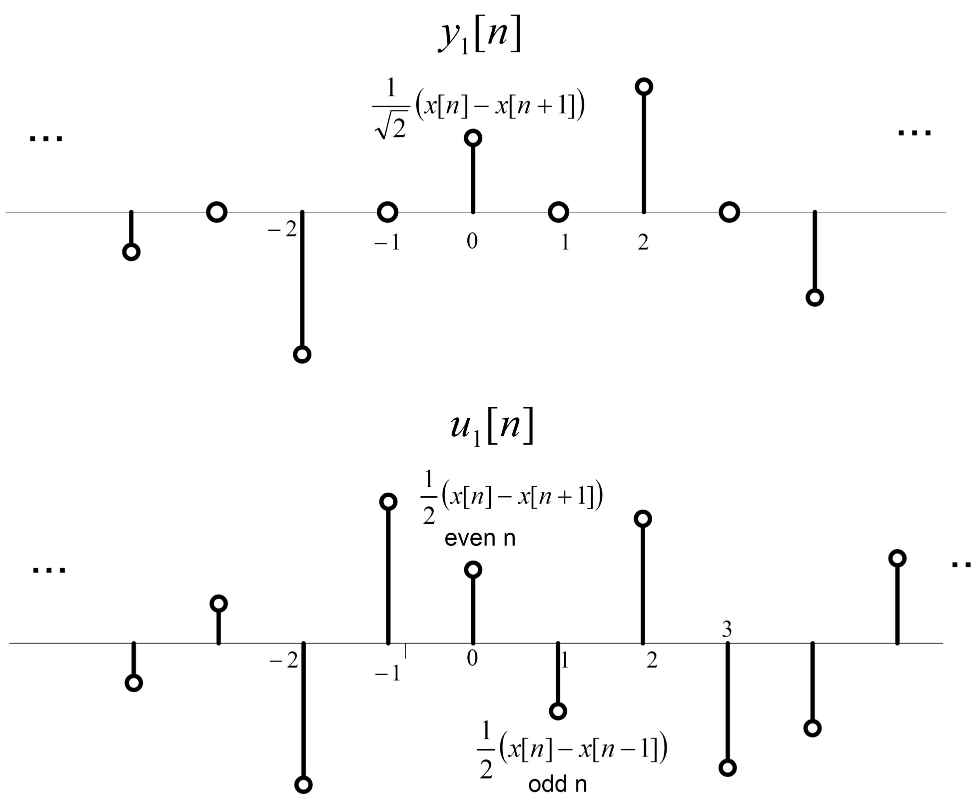


Figure 11.47: Continuation to Fig. 11.46: Illustration of synthesis filter operations.

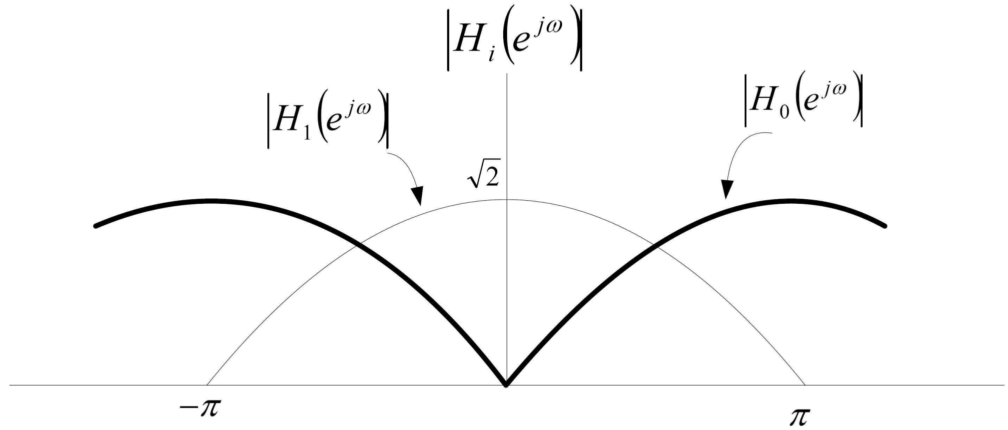


Figure 11.48: Overlapping spectra of analysis filters.

11.8 Problems

Problem 11.1 *Prove the equivalence of the two down-sampling and up-sampling with interpolator configurations shown in Fig. 11.49. These equivalent relations are called the "noble identities".*

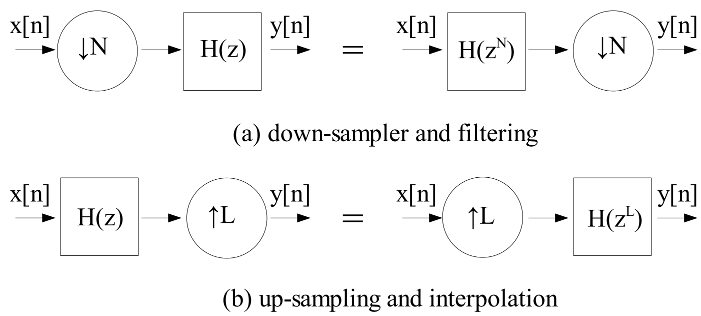


Figure 11.49: Show these two equivalences

Problem 11.2 *The system shown in Fig 11.50 approximately interpolates the sequence $x[n]$ by a factor of L. Suppose that the linear filter has impulse response $h[n]$ such that $h[n] = h[-n]$ and $h[n] = 0$ for $|n| > (RL-1)$, where R and L are integers,* i.e., *the impulse response is symmetric and of length $(2RL-1)$.*

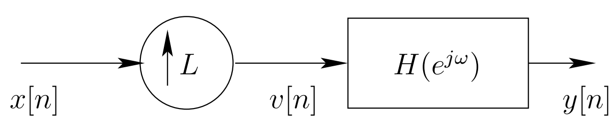


Figure 11.50: Interpolating system

(a) What condition must be satisfied by $h[n]$ in order that $y[n] = x[n/L]$ for $n = 0, \pm L, \pm 2L, \dots$? Note that $h[n] = \delta[n]$ is a trivial example for this condition. Can you give any other example?
Hint: *Note that there is no condition on $y[n]$ where n is not divisible by L.*

(b) Let $z[n]$ be the downsampled version of $y[n]$ by a sampling factor of $2L$. Find the spectrum of $z[n]$ and compare it to the spectrum of $x[n]$ which is shown in Fig. 11.51. What is the relationship between $x[n]$ and $z[n]$?

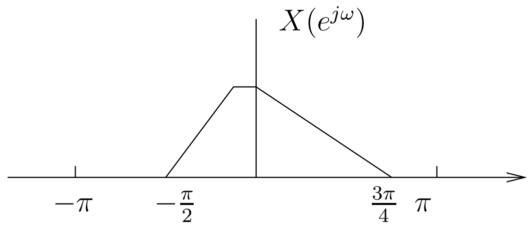


Figure 11.51: Spectrum of $x[n]$

Problem 11.3 (Downsampling in MATLAB) *In this problem we will consider the effect of aliasing in an audio file. This will be an actual experiment on using prefiltering for sampling. Download the file* hw6.wav *from the course website and load it in Matlab in the variable* `a`

```
>>[a,fs]=wavread('hw6.wav');
```

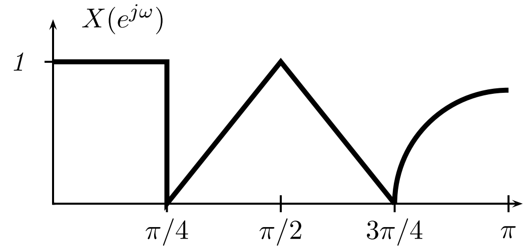
(a) *Listen to the file:*
```
>>soundsc(a,fs);
```

(b) *Compute the DFT of* `a` *and save it in* `A`*:*
```
>>A=fft(a);
```
Read the Matlab help of `wavread`*:*
```
>>help wavread;
```
and find the sampling frequency of the file. Estimate the the bandwidth of the original continuous time file, i.e., $\omega = \frac{2\Omega}{\Omega_s}$, *where* ω *and* Ω *respectively denote the frequency in transform of the discrete-time and continuous-time signals,.*
Hint: *Recall the relationship between the DFT and DTFT and note that after the sampling process, the original spectrum will be repeated at the multiples of the sampling frequency,* $k\Omega_s$, *and then it is filtered by the appropriate low-pass filter.*

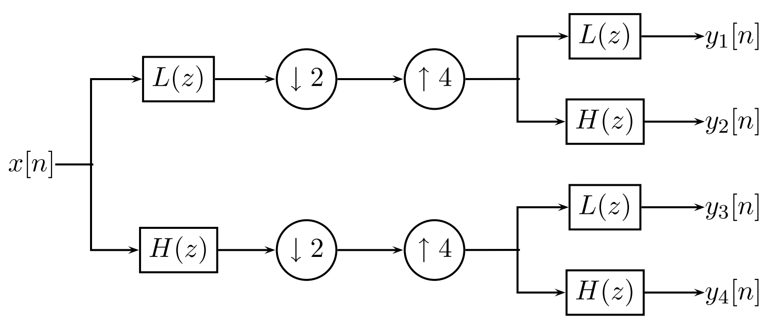
(c) *Read the Matlab help for* `downsample`*:*
```
>>help downsample;
```
Produce the sequence `b` *which is the down-sampled version of* `a` *with down-sampling factor* 3, i.e., $b[n] = a[3n]$. *Listen to* `b`*:*
```
>>soundsc(b,fs/3);
```
Does it sound like the original audio file?

(d) *Plot the spectrum of* `b`*:*
```
>>B=fft(b);
>>plot(abs(B));
```

Problem 11.4 *Consider a real discrete-time signal $x[n]$ with the following spectrum:*



Note that the spectrum of a real signal is symmetric, i.e., $X(e^{j\omega}) = X(e^{-j\omega})$*, and hence we just plotted it for $\omega \geq 0$. Now consider the following multirate processing scheme in which $L(z)$ is an ideal* lowpass *filter with cutoff frequency $\pi/2$ and $H(z)$ is an ideal* highpass *filter with cutoff frequency $\pi/2$:*



Plot the four spectra $Y_1(e^{j\omega}), Y_2(e^{j\omega}), Y_3(e^{j\omega}), Y_4(e^{j\omega})$ for $\omega \geq 0$.

Problem 11.5 *Consider the system given in Figure 11.52.*

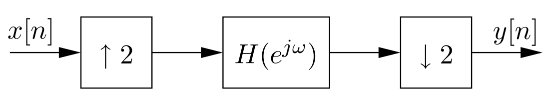


Figure 11.52: System for Problem 1.

(a) *Let $H(e^{j\omega})$ and $X(e^{j\omega})$ be as given in Figures 11.53 and 11.54 respectively. We assume that $\angle H(e^{j\omega}) = 0$. Draw $Y(e^{j\omega})$.*

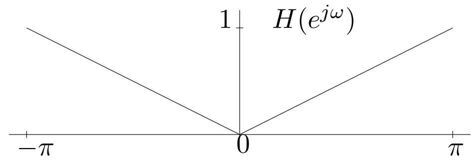


Figure 11.53: Problem 1. $H(e^{j\omega})$ to be used for Parts (a) and (c).

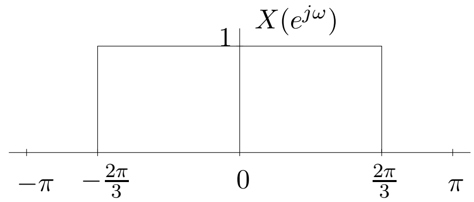


Figure 11.54: Problem 1. $X(e^{j\omega})$ to be used for Parts (a) and (c).

(b) *Consider the system given in Figure 11.55. For arbitrary $h[n]$ find an expression for*

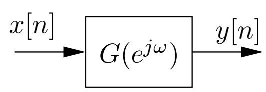


Figure 11.55: Equivalence to be shown for Problem 1.

$g[n]$ in terms of $h[n]$, such that the systems of Figures 11.52 and 11.55 are equivalent.

(c) *Using the results from Part (b), compute $Y(e^{j\omega})$ for $H(e^{j\omega})$ and $X(e^{j\omega})$ as given in Part (a).*

Problem 11.6 *Consider the system in Fig. 11.56 with $H_0(z)$, $H_1(z)$, and $H_2(z)$ as the transfer functions of LTI systems. Assume that $x[n]$ is an arbitrary stable complex signal without any symmetry properties.*

(a) *Let $H_0(z) = 1$, $H_1(z) = z^{-1}$, and $H_2(z) = z^{-2}$. Can you reconstruct $x[n]$ from $y_0[n]$, $y_1[n]$, and $y_2[n]$? If so, how? If not, justify your answer.*

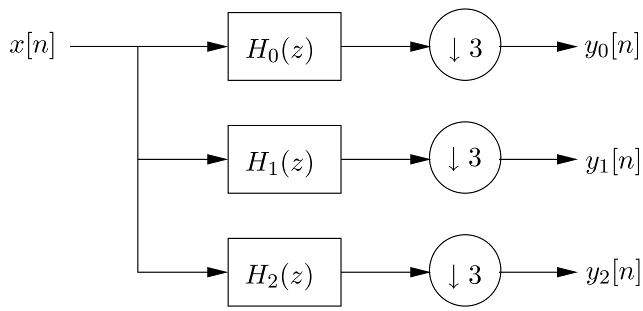


Figure 11.56: Sampling system 1.

(b) *Assume that $H_0\left(e^{j\omega}\right)$, $H_1\left(e^{j\omega}\right)$, and $H_2\left(e^{j\omega}\right)$ are as follows:*

$$H_0\left(e^{j\omega}\right) = \begin{cases} 1, & |\omega| \leq \frac{\pi}{3}, \\ 0, & \textit{otherwise}, \end{cases}$$

$$H_1\left(e^{j\omega}\right) = \begin{cases} 1, & \frac{\pi}{3}| < \omega| \leq \frac{2\pi}{3}, \\ 0, & \textit{otherwise}, \end{cases}$$

$$H_2\left(e^{j\omega}\right) = \begin{cases} 1, & \frac{2\pi}{3} < |\omega| \leq \pi, \\ 0, & \textit{otherwise}. \end{cases}$$

Can you reconstruct $x[n]$ from $y_0[n]$, $y_1[n]$, and $y_2[n]$? If so, how? If not, justify your answer.

(c) *Now consider the system in Fig. 11.57.*

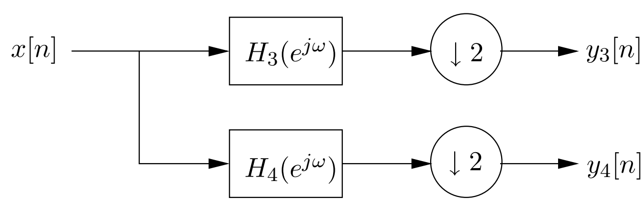


Figure 11.57: Sampling system 2.

Let $H_3\left(e^{j\omega}\right) = 1$ and

$$H_4\left(e^{j\omega}\right) = \begin{cases} 1, & 0 \leq \omega < \pi, \\ -1, & -\pi \leq \omega < 0. \end{cases}$$

Can you reconstruct $x[n]$ from $y_3[n]$ and $y_4[n]$? If so, how? If not, justify your answer.

Problem 11.7 *In your grandmother's attic you just found a treasure: a collection of rare 78rpm vinyl jazz records. The first thing you want to do is to transfer the recordings to compact discs, so you can listen to them without wearing out the originals. Your idea is obviously to play the record on a turntable and use an A/D converter to convert the line-out signal into a discrete-time sequence, which you can then burn onto a CD. The problem is, you only have a "modern" turntable, which plays records at 33rpm. Since you're a DSP wizard, you know you can just go ahead, play the 78rpm record at 33rpm and sample the output of the turntable at 44.1 KHz. You can then manipulate the signal in the discrete-time domain so that, when the signal is recorded on a CD and played back, it will sound right.*

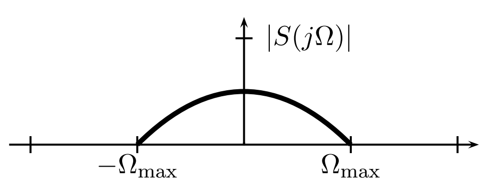
Design a system which performs the above conversion. If you need to get on the right track, consider the following:

- *Call $s(t)$ the continuous-time signal encoded on the 78rpm vinyl (the jazz music).*
- *Call $x(t)$ the continuous-time signal you obtain when you play the record at 33rpm on the modern turntable.*
- *Let $x[n] = x(nT_s)$, with $T_s = 1/44100$.*

and answer the following questions:

(a) *Express $x(t)$ in terms of $s(t)$.*

(b) *Sketch the Fourier transform $X(j\Omega)$ when $S(j\Omega)$ is as in the following figure. The highest nonzero frequency of $S(j\Omega)$ is $\Omega_{\max} = (2\pi) \cdot 16000$ Hz (old records have a smaller bandwidth than modern ones).*



(c) *Design a system to convert $x[n]$ into a sequence $y[n]$ so that, when you interpolate $y[n]$ to a continuous-time signal $y(t)$ with interpolation period T_s, you obtain $Y(j\Omega) = S(j\Omega)$.*