12 Multirate Digital Signal Processing

12.01 Introduction

The digital signal processing systems discussed so far belong to the class of single-rate systems, as the sampling rate remains constant at the input, output, and all nodes. There are numerous applications where a signal sampled at one rate needs to be converted to a different rate. For instance, various sampling rates are utilized in digital audio: 32 kHz for broadcasting, 44.1 kHz for CDs, and 48 kHz for digital audio tapes (DATs). Frequently, there's a need to transition between these sample rates. Another example is pitch control in audio recordings. In analog tapes, this is achieved by adjusting the tape recorder speed. However, in digital processing systems, it necessitates altering the sample rate.

To alter the sampling rate of a digital signal, multirate digital signal processing systems employ two fundamental devices: a *downsampler* and an *upsampler*. Systems that operate with discrete-time signals having unequal sampling rates at different points are referred to as *multirate* systems.

12.02 Basic sampling rate alteration devices

First, let's examine the characteristics of the upsampler and downsampler in the time domain, and later, we'll explore them in the frequency domain.

12.02.1 Time domain characterization

An upsampler with an upsampling factor L, where $L \in \mathbb{N}$, generates an output sequence $x_u(n)$ with a sampling rate that is L times higher than that of the input sequence $x(n)$. The upsampling operation is executed by inserting $L-1$ zero-value samples between two consecutive samples of the input sequence $x(n)$, following the relation:

$$x_u(n) = \begin{cases} x(n/L), & n = 0, \pm L, \pm 2L, \dots, \\ 0, & \text{elsewhere.} \end{cases}$$

An illustration of the upsampling process is shown in the following figure:

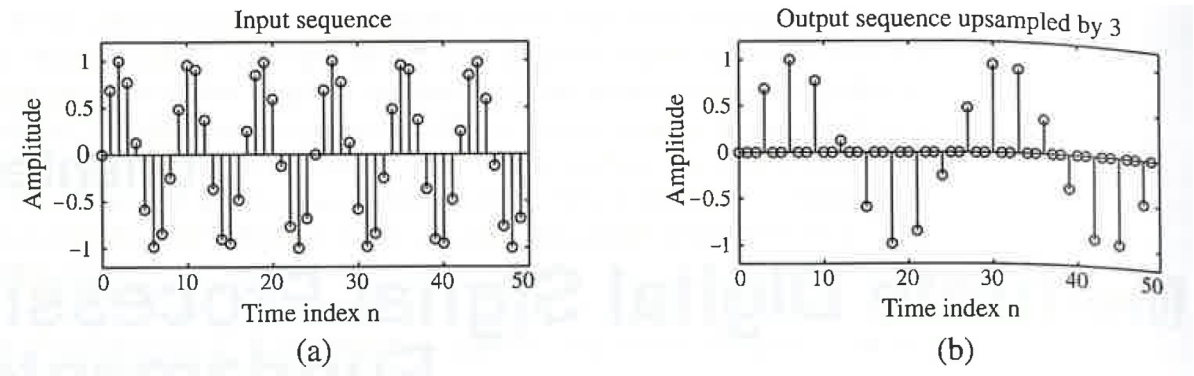

Figure 13.1: Illustration of the up-sampling process.

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

The block diagram representation of the upsampler, also known as a *sampling rate expander*, is shown in the following figure:

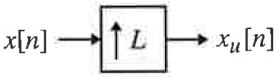

Figure 13.2: Block diagram representation of an up-sampler.

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

In practice, the zero-valued samples inserted by the upsampler are interpolated using some form of filtering process to eliminate any unnecessary spectral components from the higher-rate sequence. This process, known as *interpolation*, will be discussed later.

A downsampler with a downsampling factor M, where $M \in \mathbb{N}$, generates an output sequence $y(n)$ with a sampling rate that is $\frac{1}{M}$ of that of the input sequence $x(n)$. The downsampling operation is executed by retaining every Mth sample of the input sequence and discarding the remaining $M-1$ samples in-between. Therefore, the output sequence is generated according to the relation:

$$y(n) = x(nM).$$

All input samples with indexes equal to an integer multiple of M are retained in the output, while the others are discarded.

The downsampling process is illustrated in the following figure:

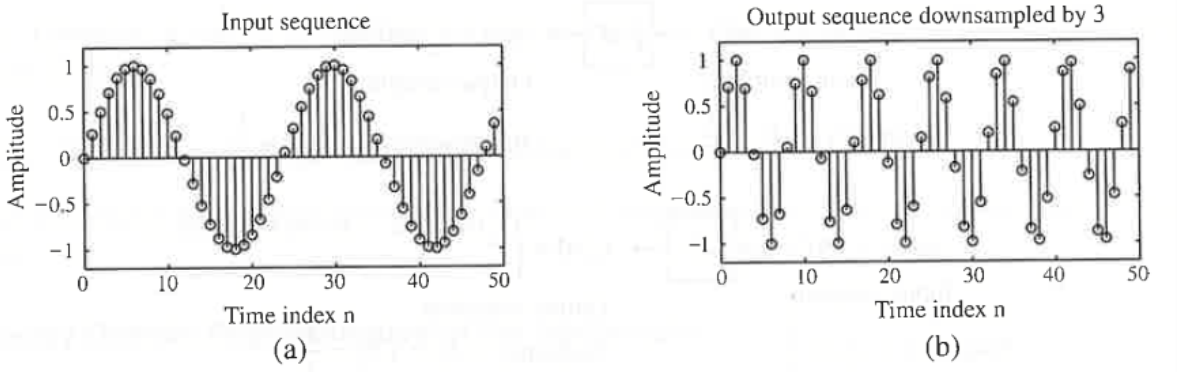

Figure 13.3: Illustration of the down-sampling process.

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

The block diagram representation of the *downsampler*, also known as a *sampling rate compressor*, is shown below:

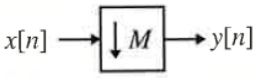

Figure 13.4: Block diagram representation of a down-sampler.

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

The sampling periods are typically not depicted in the block diagrams of the upsampler and downsampler. However, the following figure illustrates how the sampling rate changes in both cases:

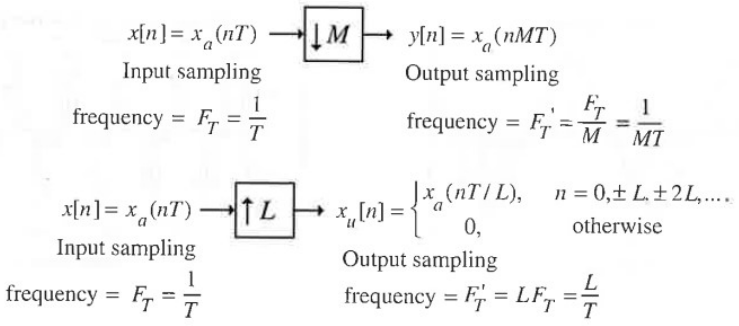

Figure 13.5: The sampling rate alteration building blocks with sampling rates explicitly shown.

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

The upsampler and downsampler building blocks are frequently used together in multirate signal processing applications. One example of using both types of devices is to achieve a sampling rate change by a rational number rather than an integer value.

12.02.2 Frequency domain characterization

We consider the case of the upsampler and derive the relations between the input and output spectra of a factor of 2 upsampler:

$$x_u(n) = \begin{cases} x(n/2), & n = 0, \pm 2, \pm 4, \dots, \\ 0, & \text{elsewhere.} \end{cases}$$

The z-transform is

$$X_u(z) = \sum_{n=-\infty}^{+\infty} x_u(z) z^{-n} = \sum_{\substack{n=-\infty \\ n \text{ even}}}^{+\infty} x(n/2) z^{-n} =$$

$$(n = 2m) = \sum_{m=-\infty}^{+\infty} x(m) z^{-2m} = X(z^2)$$

In a similar manner, we can show that for a factor of L upsampler, we have:

$$X_u(z) = X(z^L).$$

Let's examine what happens on the unit circle, i.e., for $z = e^{j\omega}$. Then, $X_u(e^{j\omega}) = X(e^{j\omega L})$. As ω ranges from 0 to 2π, $e^{j\omega L}$ will traverse around the unit circle L times, implying that the X-axis of the frequency response is compressed by a factor of L.
For instance, the following figure depicts, for simplicity, a real frequency response $X(e^{j\omega})$ with an asymmetric spectrum (hence $x(n) \in \mathbb{C}$), and the resulting upsampled spectrum $X_u(e^{j\omega})$ when $L = 2$:

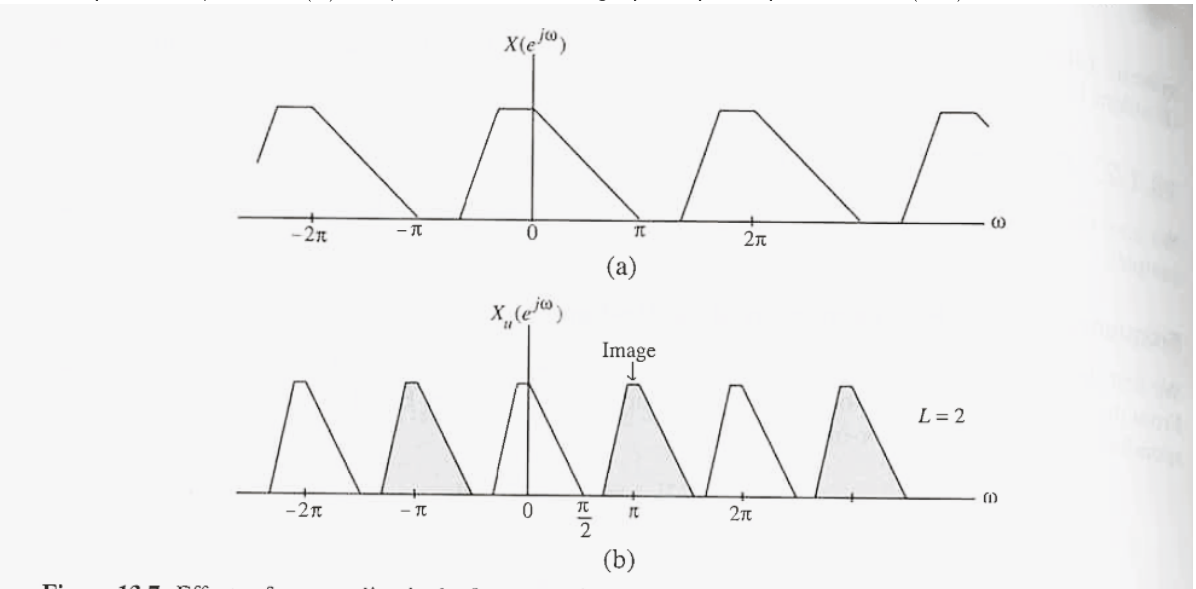

Figure 13.7: Effects of up-sampling in the frequency domain: (a) input spectrum and (b) output spectrum for $L = 2$.

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

We notice that the X-axis has been compressed by a factor of 2, and within the range $[0, 2\pi]$, we now observe two repetitions of $X(e^{j\omega})$. This process is termed imaging because we obtain additional images of the input spectrum. With a factor of L upsampling, we will obtain $L - 1$ additional images of this spectrum. Even if the original spectrum is bandlimited and lowpass, the upsampled spectrum

will not resemble a lowpass spectrum anymore due to the zero-valued samples inserted between the non-zero samples. We can eliminate the images of the original spectrum by employing a lowpass filter with a bandwidth lower than π/L. The effect of such filtering will be to "fill in" the zero-valued samples of $x_u(n)$ with interpolated sample values.
Let us now consider the down-sampler. We apply the z-transform to the input-output relationship, yielding:

$$Y(z) = \sum_{n=-\infty}^{+\infty} x(Mn)z^{-n}$$

This expression cannot be directly expressed in terms of $X(e^{j\omega})$. To address this, we introduce an intermediate signal:

$$x_{\text{int}}(n) = \begin{cases} x(n), & n = 0, \pm M, \pm 2M, \dots \\ 0 & \text{otherwise.} \end{cases}$$

Then, we have:

$$Y(z) = \sum_{n=-\infty}^{+\infty} x(Mn)z^{-n} = \sum_{n=-\infty}^{+\infty} x_{\text{int}}(Mn)z^{-n} =$$
$$(k = Mn) = \sum_{k=-\infty}^{+\infty} x_{\text{int}}(k)z^{-k/M} = X_{\text{int}}(z^{1/M}).$$

$x_{\text{int}}(n)$ can be related to $x(n)$, since

$$x_{\text{int}}(n) = c(n)x(n)$$

where

$$c(n) = \begin{cases} 1, & n = 0, \pm M, \pm 2M, \dots \\ 0 & \text{otherwise,} \end{cases}$$

and

$$c(n) = \frac{1}{M} \sum_{k=0}^{M-1} e^{j\frac{2\pi}{M}kn}.$$

Then, we have:

$$X_{\text{int}}(z) = \sum_{n=-\infty}^{+\infty} c(n)x(n)z^{-n} = \frac{1}{M} \sum_{n=-\infty}^{+\infty} \sum_{k=0}^{M-1} e^{j\frac{2\pi}{M}kn} x(n)z^{-n}$$
$$= \frac{1}{M} \sum_{k=0}^{M-1} \sum_{n=-\infty}^{+\infty} x(n)(e^{-j\frac{2\pi}{M}k}z)^{-n} = \frac{1}{M} \sum_{k=0}^{M-1} X(e^{-j\frac{2\pi}{M}k}z)$$

and

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

To understand the implications of this relation, let's consider a factor of 2 downsampler and the resulting output spectrum:

$$Y(e^{j\omega}) = \frac{1}{2} \sum_{k=0}^{1} X(e^{-j\frac{2\pi}{2}k}e^{j\omega/2}) = \frac{1}{2} \left[X(e^{j\omega/2}) + X(e^{j(\omega-2\pi)/2}) \right]$$

$X(e^{j\omega/2})$ represents the spectrum of the original signal, which has been stretched on the X-axis by a factor of 2 and is now periodic with a period of 4π. $X(e^{j(\omega-2\pi)/2})$ is the same spectrum but time-shifted by 2π. In general, these two spectra overlap, resulting in an aliasing error caused by this overlap.

The situation is illustrated in the following figure:

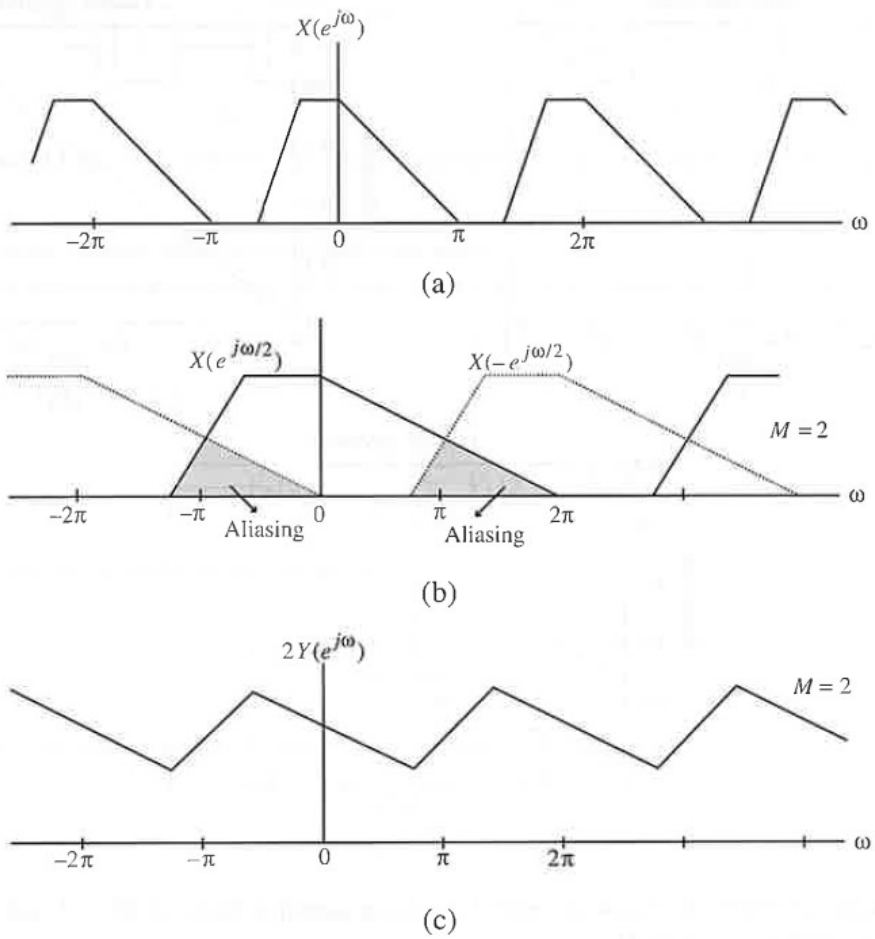

Figure 13.10: Illustration of the aliasing effect in the frequency domain caused by down-sampling.

(From S. K. Mitra, “Digital signal processing: a computer based approach”, McGraw Hill, 2011)

The absence of overlap, and hence no aliasing, occurs only if $X(e^{j\omega}) = 0$ for $|\omega| > \pi/2$.

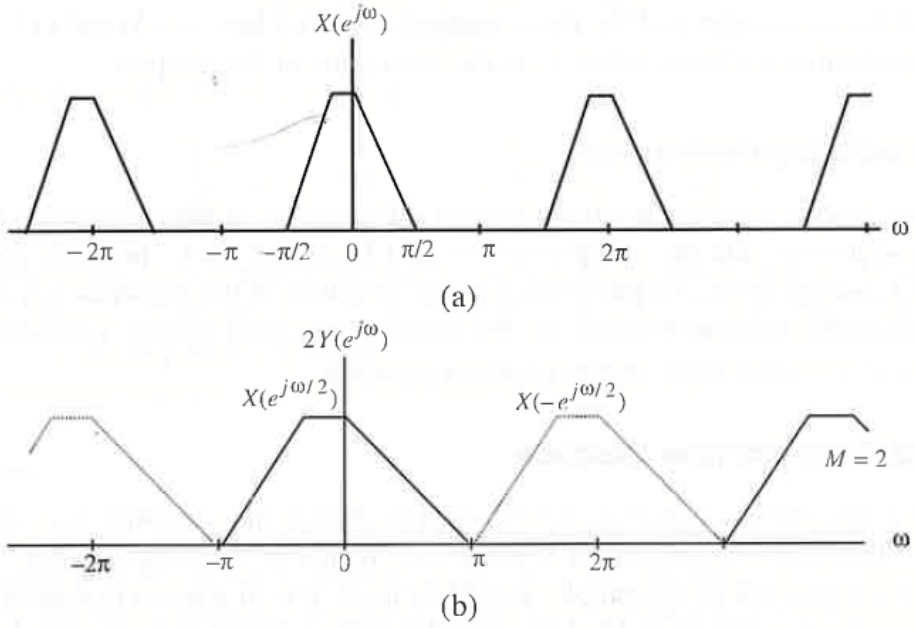

Figure 13.11: Effect of down-sampling in the frequency domain illustrating absence of aliasing.

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

For the general case, the situation remains the same. With a factor of M downsampler,

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

The spectrum $Y(e^{j\omega})$ consists of M uniformly shifted and stretched versions of $X(e^{j\omega})$, each stretched by a factor of M.
Aliasing is absent if and only if $X(e^{j\omega}) = 0$ for $|\omega| > \pi/M$.

12.02.3 Cascade equivalences

Complex multirate systems are formed by an interconnection of basic sampling rate alteration devices and the components of an LTI digital filter. In many applications, these devices appear in cascade form. Interchanging the positions of the branches in a cascade can often lead to a computationally efficient realization.
We will now consider some identities that we will exploit later on.

Up-sampler and down-sampler cascade

The basic sampling rate alteration devices can only up-sample or down-sample a signal by an integer factor. To implement a fractional change in sampling rate, a cascade of an upsampler and downsampler should be used. It is interesting to determine under which conditions the cascade of a factor M downsampler and a factor L upsampler are interchangeable, with no change in the input-output relation.

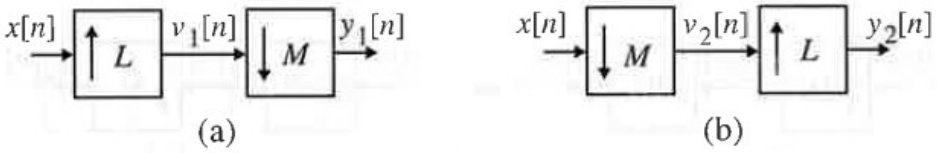

Figure 13.13: Two different cascade arrangements of a down-sampler and an up-sampler.

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

Consider first the case of the upsampler followed by the downsampler, denoted as case (a) in the figure:

$$\begin{aligned} V_1(z) &= X(z^L) \\ Y_1(z) &= \frac{1}{M}\sum_{k=0}^{M-1} V_1(z^{1/M}e^{-j\frac{2\pi}{M}k}) = \\ &= \frac{1}{M}\sum_{k=0}^{M-1} X(z^{L/M}e^{-j\frac{2\pi}{M}kL}) \end{aligned}$$

In the case of the cascade of the downsampler with the upsampler,

$$\begin{aligned} V_2(z) &= \frac{1}{M}\sum_{k=0}^{M-1} X(z^{1/M}e^{-j\frac{2\pi}{M}k}) \\ Y_1(z) &= V_2(z^L) = \\ &= \frac{1}{M}\sum_{k=0}^{M-1} X(z^{L/M}e^{-j\frac{2\pi}{M}k}) \end{aligned}$$

It follows that $Y_1(z) = Y_2(z)$ if

$$\sum_{k=0}^{M-1} X(z^{L/M}e^{-j\frac{2\pi}{M}kL}) = \sum_{k=0}^{M-1} X(z^{L/M}e^{-j\frac{2\pi}{M}k}).$$

The equality is satisfied if and only if M and L are *relatively prime*, meaning they do not have common factors greater than 1. Under this condition, $e^{-j\frac{2\pi}{M}kL}$ and $e^{-j\frac{2\pi}{M}k}$ take the same set of values for $k = 0, 1, \ldots, M-1$.

Noble identities

Two other simple cascade equivalence relations are depicted in the following figure:

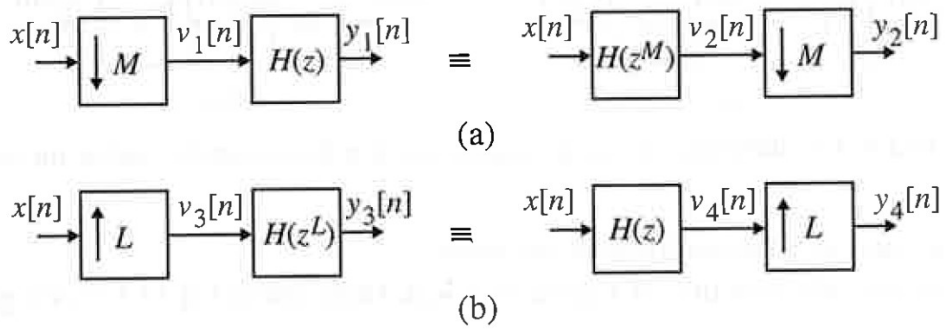

Figure 13.14: Cascade equivalences: (a) equivalence #1 and (b) equivalence #2.

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

These relations are commonly used in multirate networks to rearrange sampling rate alteration devices (downsamplers and upsamplers) into the most convenient positions.

12.03 Multirate structures for sampling rate conversion

The process of reducing the sample rate of a sequence is commonly known as *decimation*, and the corresponding system is called a *decimator*. Similarly, the process of increasing the sample rate is called *interpolation*, and the corresponding system is called an *interpolator*.

We know from the sampling theorem that the sampling rate of a critically sampled signal with a spectrum occupying the full Nyquist range cannot be reduced any further, as this would introduce aliasing. Therefore, the bandwidth of a critically sampled signal must first be reduced by lowpass filtering before the sampling rate is further reduced by the downsampler.

On the other hand, the zero-valued samples introduced by the upsampler must be interpolated for an effective sampling rate increase. We must remove the repetitions of the spectrum, the images, generated by the upsampler. Again, this operation can be performed with lowpass filtering.

Since a fractional sampling rate alteration can be obtained by concatenating an interpolator and a decimator, filters are also needed to design such multirate systems.

Here, we derive the input-output relations for the multirate structures used for sampling rate conversion.

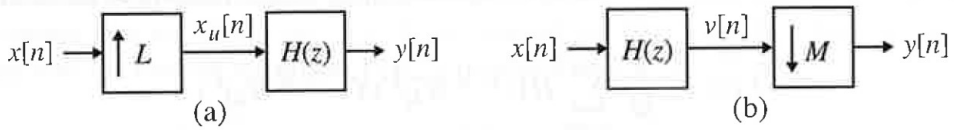

Figure 13.15: Sampling rate alteration systems for integer-valued conversion factors: (a) interpolator and (b) decimator.

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

We have observed that upsampling by a factor of L leads to the periodic repetition of the signal spectrum. Hence, the basic interpolator structure for integer-valued sampling rate increase comprises an

upsampler followed by a lowpass filter $H(z)$ with a cutoff frequency of π/L. $H(z)$ is referred to as the *interpolation filter*, and it is employed to eliminate the $L-1$ undesired images in the upsampled signal $x_u(n)$.

Since downsampling by a factor of M may lead to aliasing, the fundamental decimator structure comprises a lowpass filter $H(z)$ with a stop-band starting at π/M, followed by the downsampler. In this context, the lowpass filter is referred to as the *decimation filter* and is utilized to prevent aliasing by constraining the spectrum of the input signal to $|\omega| < \pi/M$ before downsampling.

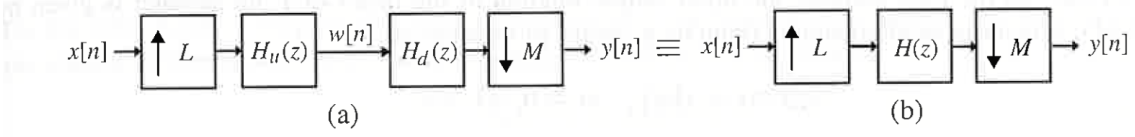

Figure 13.16: (a) General scheme for increasing the sampling rate by L/M and (b) an efficient implementation.

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

A fractional change in the sampling rate by a rational factor can be achieved by cascading an interpolator and a decimator. The interpolator must precede the decimator to ensure that the baseband of the intermediate signal is greater than that of the original signal and to prevent the loss of components. Both the interpolator and the decimator involve the use of a lowpass filter, denoted as $H_u(z)$ and $H_d(z)$, respectively, operating at the same sampling rate. These two lowpass filters can be replaced by a single lowpass filter $H(z)$, with the passband set to the lowest of the two, i.e., with a stopband starting at $\pi/\max(L,M)$.

Let us first develop the input-output relationship of the decimator. Let $h(n)$ be the impulse response of the decimation filter $H(z)$. Then the corresponding output is

$$v(n) = \sum_{r=-\infty}^{+\infty} h(n-r)x(r)$$

The output of the decimator is instead

$$y(n) = v(Mn)$$

which, combined with the previous one, gives

$$y(n) = \sum_{r=-\infty}^{+\infty} h(Mn-r)x(r).$$

Note that if the decimation filter is implemented in FIR form, we only need to compute its output once every M input samples, thereby reducing the computational complexity by a factor of M. However, with an IIR filter, we need to compute all output samples, and we do not achieve such a significant reduction in computation. Therefore, FIR filters are often preferred over IIR filters in multirate systems.

Next, we develop the input-output relation of the interpolator. In the time domain, the input-output relation of the upsampler with a factor of L is given by:

$$x_u(Lm) = x(m), \qquad m = 0, \pm 1, \pm 2, \ldots$$

and is 0 elsewhere.
Substituting this expression into the input-output relation of the interpolation filter:

$$y(n) = \sum_{r=-\infty}^{\infty} h(n-r)x_u(r)$$

which, with a change of variable, becomes:

$$y(n) = \sum_{m=-\infty}^{\infty} h(n-mL)x(m).$$

Again, using an FIR interpolation filter, we can reduce the number of operations to be performed for each sample by a factor of L, because only one every L sample of $h(n)$ is involved in the relationship. However, with IIR filters, we do not have such a significant computational advantage.
Eventually, for the fractional-rate sampling rate converter, the input-output relationship can be obtained in a similar manner and is given by:

$$y(n) = \sum_{m=-\infty}^{+\infty} h(Mn-Lm)x(m).$$

12.04 Multistage Design of Decimator and Interpolator

The decimator and interpolator structures we have seen are single-stage structures. Indeed, the basic scheme for implementation involves a single lowpass filter and a single sampling rate alteration device. However, if the interpolation factor L can be expressed as a product of two integers L_1 and L_2, the factor of L interpolation can be realized in two stages. Similarly, if the decimation factor M is the product of two integers M_1 and M_2, the factor of M decimation can also be realized in two stages:

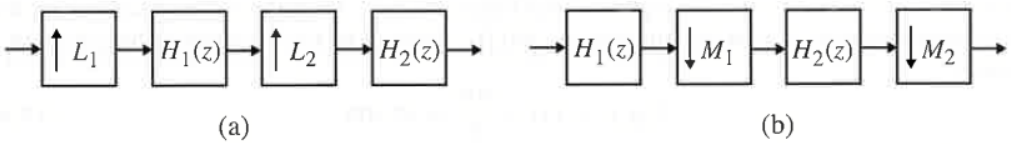

Figure 13.22: Two-stage implementation of sampling rate alteration systems: (a) interpolator and (b) decimator.

(From S. K. Mitra, “Digital signal processing: a computer based approach”, McGraw Hill, 2011)

The design can involve more than two stages, depending on the number of factors used to express L and M. In general, computation efficiency is significantly improved by designing the sampling rate alteration system as a cascade of several stages.
For example, consider a decimator for the reduction of the sampling rate from 12 kHz to 400 Hz, with a decimation factor of 30:

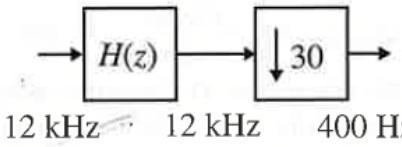

Figure 13.23: Block-diagram representation of the single-stage factor-of-30 decimator.

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

The specifications for the decimation filter are as follows: passband edge $F_p = 180$ Hz, stopband edge $F_s = 200$ Hz, passband ripple $\delta_p = 0.002$, and stopband ripple $\delta_s = 0.001$. Thus, $\omega_p = \frac{180}{12000} 2\pi$. If $H(z)$ is an equiripple linear-phase FIR filter, the order N_H (determined using `firpmord`) is

$$N_H = 1827$$

The number of multiplications per second (ignoring the filter symmetry) is

$$(N_H + 1) \times \frac{F_T}{30} = 1828 \times \frac{12\,000}{30} = 730\,800$$

Let us now consider the implementation of $H(z)$ as a cascade realization of two systems. The first system is a comb filter $F(z^{15})$ that realizes the lowpass filter we want with the requested transition band but also generates some upper bands. The other filter $I(z)$ is a pure lowpass filter that selects the lowest band but considers a larger transition band, since it takes into account the spectral gaps between the passbands of $F(z^{15})$. Since the cascade of the two systems has a passband ripple that is the sum in dB of the ripples of the two systems, to compensate we will require both $F(z)$ and $I(z)$ to have a passband ripple $\delta_p = 0.001$ (rather than $\delta_p = 0.002$). On the other hand, the stopband of $I(z)F(z^{15})$ is at least as good as $I(z)$ and $F(z)$, and therefore we will require $\delta_s = 0.001$.
We design $F(z)$ with the Parks-McClellan algorithm considering $F_p = 15 \times 180$ Hz and $F_s = 15 \times 200$ Hz. The transition band is 15 times larger than before, and the filter requires an order $N_f = 130$.
We also design $I(z)$ with the Parks-McClellan algorithm considering $F_p = 180$ Hz and $F_s = \frac{12\,000}{15} - 200 = 600$ Hz. The transition band is much larger than the 20 Hz of the first filter, and the order $N_I = 93$.

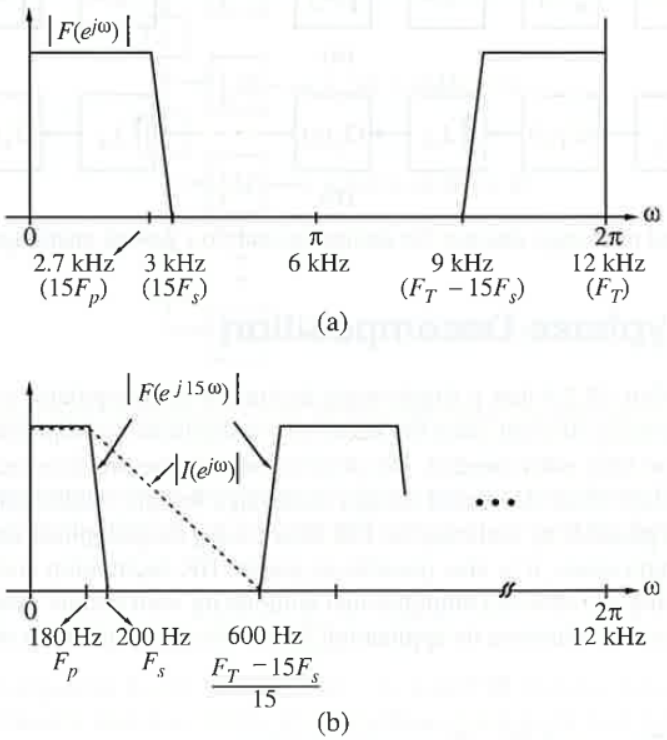

Figure 13.24: Decimation filter design based on the IFIR approach (frequency response plots not shown to scale).

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

We can now drastically reduce the computational complexity by making use of the cascade equivalence and performing multistage decimation.

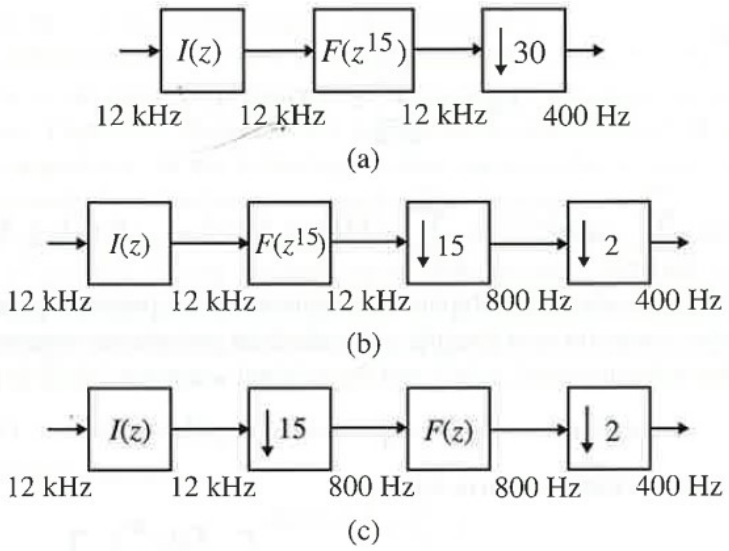

Figure 13.25: The steps in the two-stage realization of the decimator structure.

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

The implementation of $F(z)$ at sampling rate 800 Hz followed by a down-sampling by a factor 2 requires

$$131 \times \frac{800}{2} = 52\,400 \text{ multiplications per second.}$$

The implementation of $I(z)$ at sampling rate 12 kHz followed by a down-sampling by a factor 15 requires

$$94 \times \frac{12000}{15} = 75\,200 \text{ multiplications per second.}$$

The total complexity of this implementation becomes

$$52\,400 + 75\,200 = 127\,600 \text{ multiplications per second}$$

i.e. 5.7 time smaller than the direct single-stage implementation.

12.05 The polyphase decomposition

We have seen that a single-stage decimator and interpolator employing FIR lowpass filters can be computationally efficient, since the multiplications to compute the output can be carried out only when needed. We have also seen the computational requirements can be further decreased using a multi-stage design. We can further reduce the computational complexity by exploiting the polyphase decomposition of FIR filters.

To illustrate this approach, let's consider, for simplicity, a causal FIR transfer function $H(z)$ of length 9:

$$H(z) = h(0) + h(1)z^{-1} + h(2)z^{-2} + h(3)z^{-3} + h(4)z^{-4} + h(5)z^{-5} + h(6)z^{-6} + h(7)z^{-7} + h(8)z^{-8}$$

This transfer function can expressed as the sum of two terms:

$$\begin{aligned} H(z) &= \left(h(0) + h(2)z^{-2} + h(4)z^{-4} + h(6)z^{-6} + h(8)z^{-8}\right) \\ &+ \left(h(1)z^{-1} + h(3)z^{-3} + h(5)z^{-5} + h(7)z^{-7}\right) = \\ &= \left(h(0) + h(2)z^{-2} + h(4)z^{-4} + h(6)z^{-6} + h(8)z^{-8}\right) \\ &+ z^{-1}\left(h(1) + h(3)z^{-2} + h(5)z^{-4} + h(7)z^{-6}\right) \end{aligned}$$

and by using the notation

$$\begin{aligned} E_0(z) &= h(0) + h(2)z^{-1} + h(4)z^{-2} + h(6)z^{-3} + h(8)z^{-4} \\ E_1(z) &= h(1) + h(3)z^{-1} + h(5)z^{-2} + h(7)z^{-3} \end{aligned}$$

we can rewrite

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

In a similar manner, selecting every third term, we can express it in the form:

$$H(z) = E_0(z^3) + z^{-1}E_1(z^3) + z^{-2}E_1(z^3)$$

where now

$$\begin{aligned} E_0(z) &= h(0) + h(3)z^{-1} + h(6)z^{-2} \\ E_1(z) &= h(1) + h(4)z^{-1} + h(7)z^{-2} \\ E_2(z) &= h(2) + h(5)z^{-1} + h(8)z^{-2}. \end{aligned}$$

The decomposition of $H(z)$ in this form is called *polyphase decomposition*. In case of M branches, we have

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

where the polyphase component $E_k(z)$ is an FIR transfer function.
We will not consider the case of IIR filters, but I would like to mention the fact that polyphase decomposition is applicable also to IIR filters, and in that case, the polyphase component $E_k(z)$ is IIR. Moreover, for certain types of stable lowpass IIR transfer functions with cutoff at π/M, the polyphase decomposition takes the form:

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

with $A_k(z)$ being a stable allpass transfer function.
Let us consider the M-branch polyphase decomposition of $H(z)$. The following figure provides the direct realization and the transposed realization of

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

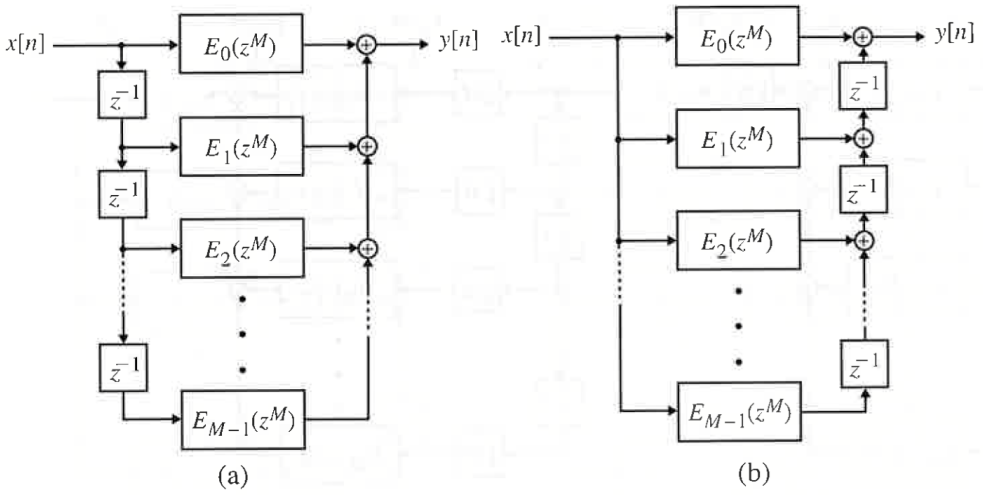

Figure 13.28: (a) Direct realization of an FIR filter based on a Type I polyphase decomposition and (b) its transpose.

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

The first polyphase decomposition is also called *Type I polyphase decomposition*. Very often, the transposed structure is represented in an alternative manner considering

$$R_l(z^M) = E_{M-1-l}(z^M).$$

The corresponding polyphase decomposition is thus given by

$$H(z) = \sum_{l=0}^{M-1} z^{-(M-1-l)} R_l(z^M).$$

This decomposition is called *Type II polyphase decomposition*.

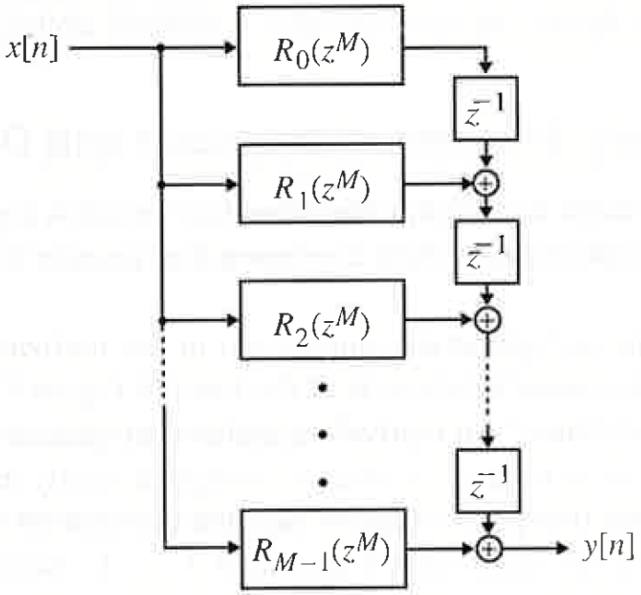

Figure 13.29: Realization of an FIR filter based on a Type II polyphase decomposition.

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

Computationally efficient decimator and interpolator structures employing lowpass filters can be derived by applying a polyphase decomposition to the corresponding transfer functions.

Consider first the use of the polyphase decomposition in the realization of the decimation filter. Using the first cascade equivalence, we can obtain a computationally more efficient realization.

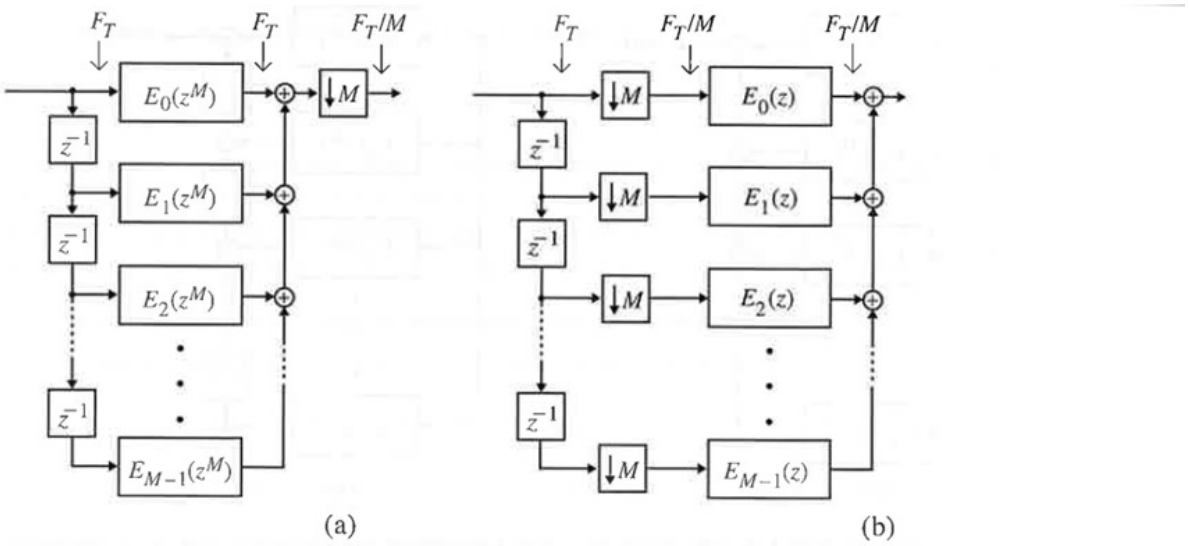

Figure 13.30: (a) Decimator implementation based on a Type I polyphase decomposition and (b) computationally efficient decimator structure. In the figures, the sampling rates of pertinent sequences are indicated by an arrow.

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

Assume first the use of a standard decimation filter $H(z)$, with length N. Since the decimator output $y(n)$ is obtained by down-sampling the filter output $v(n)$ by a factor M, we need to compute $v(n)$ only for $n = \ldots, -2M, -M, 0, M, 2M, \ldots$, the computational requirement is of N multiplications and $N-1$ addition per output sample. However, as n increases, the values memorized in the tapped delay line change. Thus, it is necessary to perform all operations in one input sampling period (i.e., at the highest rate) and in the following $M-1$ periods the arithmetic unit will remain idle.

Now consider the polyphase decomposition structure. If the length of the subfilter $E_k(z)$ is N_k, then $N = \sum_{k=0}^{M-1} N_k$. The overall computational cost is of $\sum_{k=0}^{M-1} N_k = N$ multiplications and $\sum_{k=0}^{M-1}(N_k - 1) + (M-1) = N-1$ additions per decimator output sample. The computational cost hasn't changed, but now all operations are performed at the lowest rate, with the arithmetic unit operative all instants of the output sampling period (which is M times that of the input sampling period).

Similar savings can be obtained in the case of the interpolator structure employing polyphase decomposition for the realization of a computationally efficient interpolator. In this case, it is convenient to use the second cascade equivalence:

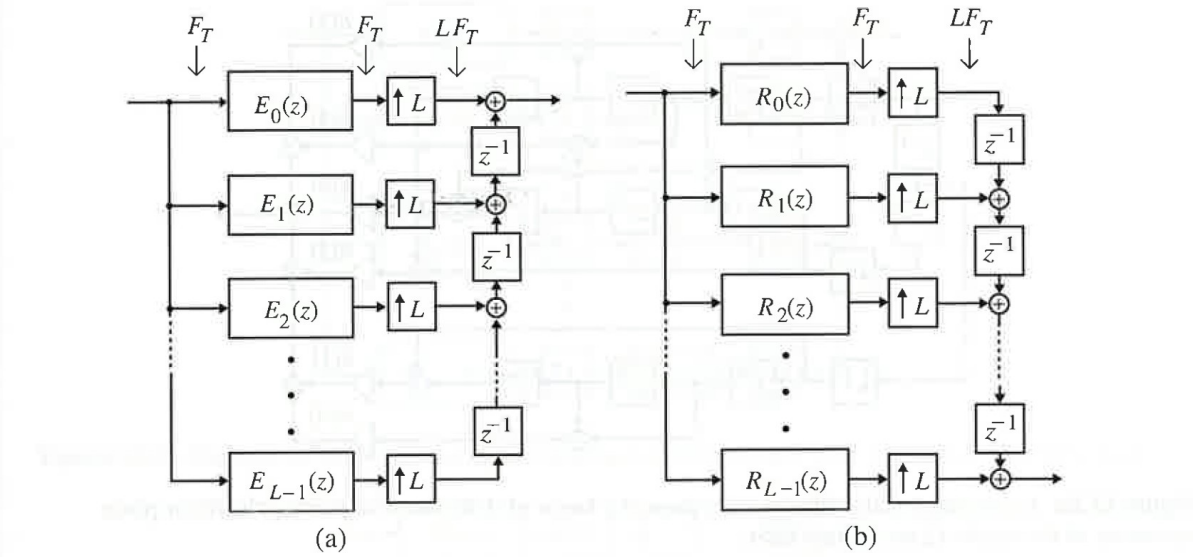

Figure 13.31: Computationally efficient interpolator structures: (a) Type I polyphase decomposition and (b) Type II polyphase decomposition. In the figures, the sampling rates of pertinent sequences are indicated by an arrow.

(From S. K. Mitra, "Digital signal processing: a computer based approach", McGraw Hill, 2011)

For more information read:

S. K. Mitra, "Digital Signal Processing: a computer based approach," 4th edition, McGraw-Hill, 2011

Chapter 13.1, pp. 740-749

Chapter 13.2, pp. 750-752,

Chapter 13.3, pp. 758-759,

Chapter 13.4, pp. 762-767,