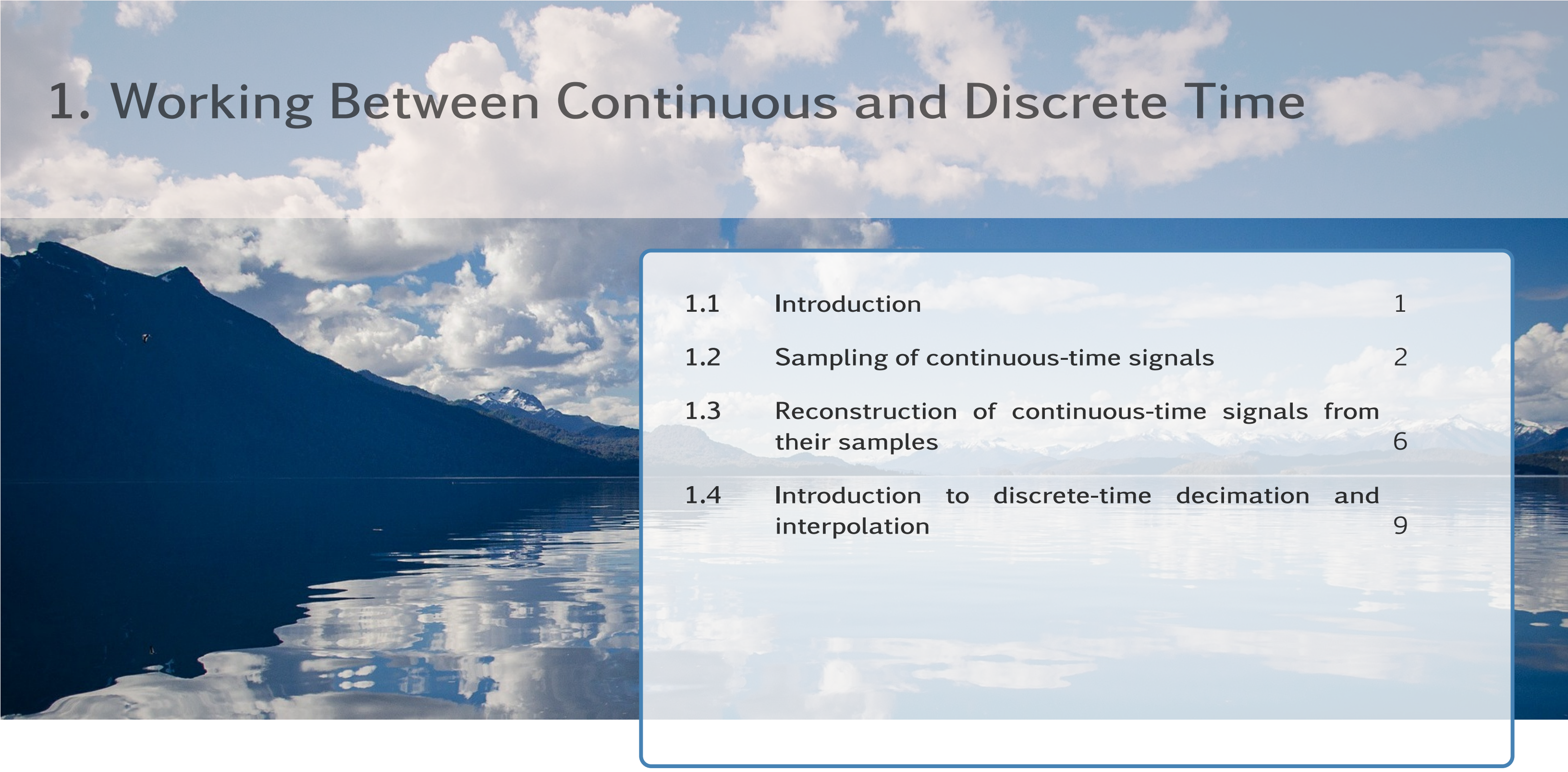
1. Working Between Continuous and Discrete Time



In signal processing we need to be able to move fluently between the continuous-time and discrete-time worlds. Natural signals, and our perception of them, nearly always occur in the continuous-time world. Nevertheless, almost all signal processing operations are implemented in the discrete-time world. Hence we need to be able to understand how we move from one world to the other, and to understand the implications that sampling has on the nature of the signals in time and frequency.

1.1 Introduction

In this chapter we review the mathematics that are the basis for sampling of continuous-time signals and their reconstruction. We will also discuss the closely-related mathematics that underly discrete-time decimation and interpolation that enable changes in the effective sampling rate of a discrete-time signal.

1.2 Sampling of continuous-time signals

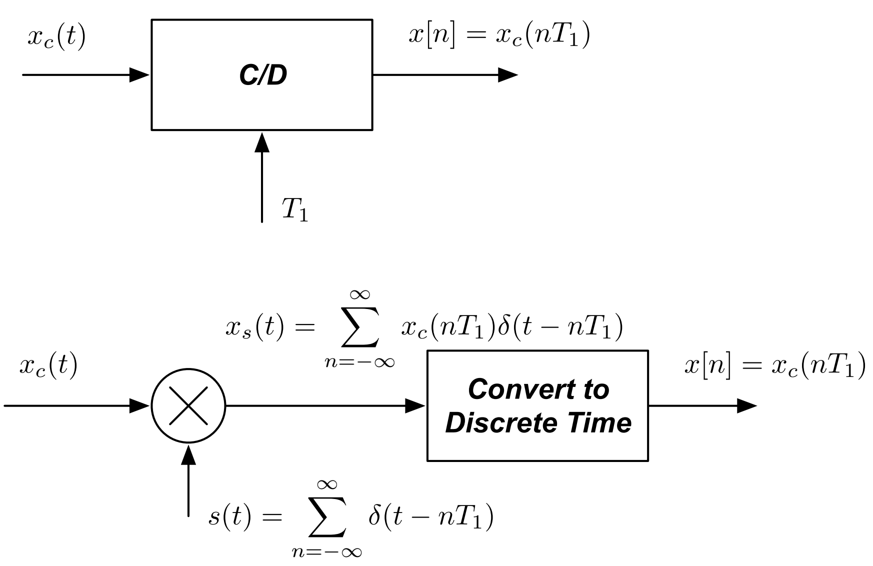

$x_c(t)$
C/D
$x[n] = x_c(nT_1)$
T_1
$x_s(t) = \sum_{n=-\infty}^{\infty} x_c(nT_1)\delta(t - nT_1)$
$x_c(t)$
Convert to Discrete Time
$x[n] = x_c(nT_1)$
$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_1)$

Figure 1.1: Upper panel: block diagram of C/D conversion. Lower panel: Specification of the underlying signal processing.

The block diagram in the upper panel of Fig. 1.1 shows the basic structure of the ideal *continuous-to-discrete-time converter* or C/D converter. The input is a continuous-time function $x_c(t)$, which we assume to be a bandlimited time function such that its continuous-time Fourier transform $X(j\Omega)$ equals zero for $|\Omega| > W$.[1] The output is a discrete-time function $x[n]$ that is equal to $x_c(t)$ evaluated at times $t = nT_1$, where T_1 is the sampling period in seconds.

In order to understand the mathematical representation of sampling and its implications, we use the more detailed description in the lower panel of Fig. 1.1. Specifically, the input is first multiplied by $s(t)$, an infinite train of delta functions of area 1, which are separated by an interval of T_1 seconds. The function $x_s(t)$ is the product of $x_c(t)$ and $s(t)$, and is a train of delta functions $x_s(t)$ that are separated in time by intervals of T_1 seconds and have areas equal to the amplitude of the original signal $x_c(nT_1)$ at the times at which the impulses occur. This sequence is then converted magically (in our mathematical model) into discrete time pulses $x[n]$ which have *amplitudes* equal to the *areas* of the corresponding delta functions in $x_s(t)$, so $x[n]$ will be equal to $x_c(nT_1)$.

Time-domain representations. It is helpful to be able to visualize these signals in both time and frequency. The left side of Fig. 1.2 depicts, for an arbitrary time function and its spectrum, the various continuous-time functions and discrete-time functions depicted in Fig. 1.1. Their Fourier transforms are shown on the right side of the figure. More specifically,

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_1) \tag{1.1}$$

[1] We will adopt the notational convention of using the Greek letter Ω for continuous-time frequency in radians per second and using ω for discrete-time frequency in radians.

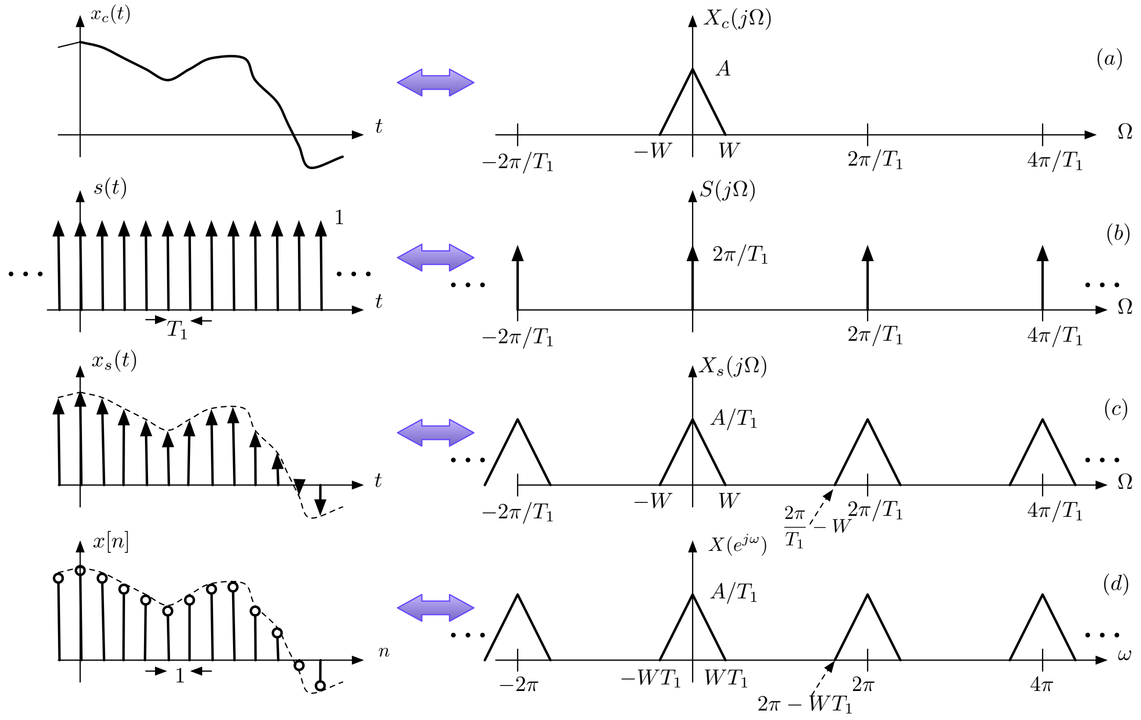


Figure 1.2: Left column: the time functions (a) $x_c(t)$, (b) $s(t)$, (c) $x_s(t)$, and (d) $x[n]$. Right column: The corresponding Fourier transforms in continuous and discrete time.

As described above,

$$x_s(t) = x_c(t)s(t) = \sum_{n=-\infty}^{\infty} x_c(nT_1)\delta(t - nT_1) \tag{1.2}$$

The function $x[n]$ is a train of discrete-time samples that have *amplitudes* that are equal to the *areas* of the corresponding impulses of $x_s(t)$. We also note that $x_s(t)$ is a function of t, which is a real variable, while $x[n]$ is a function of n, which is meaningful only when it is an integer.

$$x[n] = \sum_{l=-\infty}^{\infty} x_c(lT_1)\delta[n - l] \tag{1.3}$$

Frequency-domain representations. Considering now the corresponding functions in the frequency domain, let us assume that $X_c(j\Omega)$, the continuous-time Fourier transform (CTFT) of $x_c(t)$, is of arbitrary shape, nonzero only for $|\Omega| \leq W$, and with a maximum amplitude equal to A, as depicted in Fig. 1.2. It can be shown that $S(j\Omega)$, the CTFT of $s(t)$, is also an infinite train of delta functions:

$$S(j\Omega) = \frac{2\pi}{T_1}\sum_{k=-\infty}^{\infty} \delta\left(\Omega - \frac{2\pi k}{T_1}\right) \tag{1.4}$$

Now, because $x_s(t) = x_c(t)s(t)$ in the time domain, we obtain in the frequency domain

$$X_s(j\Omega) = \frac{1}{2\pi}X_c(j\Omega) * S(j\Omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X_c(j\theta)S(j(\Omega - \theta))d\theta \tag{1.5}$$

where the symbol $*$ indicates convolution. Because convolving $X_c(j\Omega)$ with an infinite train of delta functions merely replicates and shifts the spectrum, we obtain

$$X_s(j\Omega) = \frac{1}{2\pi} X_c(j\Omega) * \sum_{k=-\infty}^{\infty} \frac{2\pi}{T_1}\delta\left(\Omega - \frac{2\pi k}{T_1}\right) = \frac{1}{T_1}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\Omega - \frac{2\pi k}{T_1}\right)\right) \tag{1.6}$$

as shown in Row (c) of Fig. 1.2.

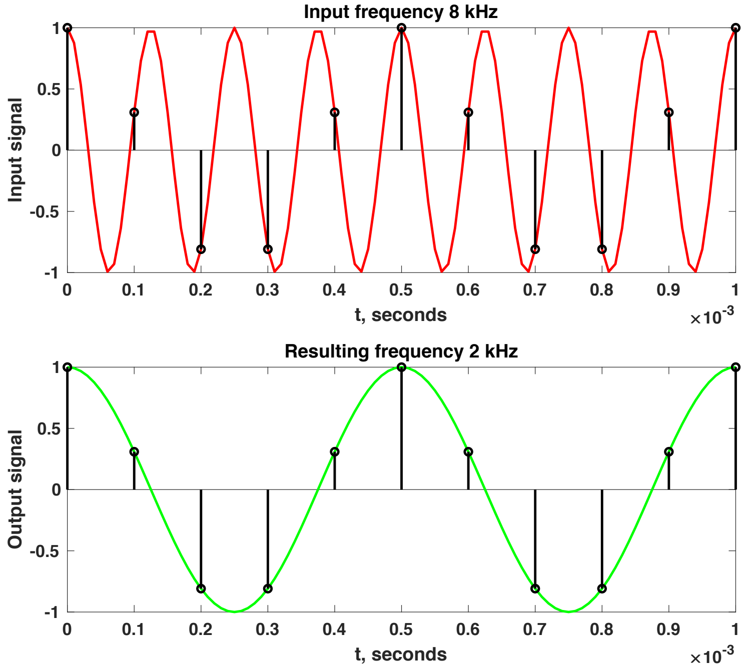


Figure 1.3: Example of aliasing distortion, with an input signal of 8 kHz (upper panel) sampled at 10 kHz producing an output signal of 2 kHz (lower panel).

Avoiding aliasing distortion. In order for the continuous-time signal to ultimately be recovered without distortion, it is necessary that the replications of the original spectrum in Fig. 1.2 not overlap one another. It can easily be seen that the separation of the replications is related to the sampling period T_1 while the width of the replications is twice the signal bandwidth W. It is clear from Fig. 1.2 that overlap will be avoided if

$$\frac{2\pi}{T_1} - W > W \text{ or } W < \frac{\pi}{T_1} \tag{1.7}$$

In other words, the sampling frequency must be at least twice as great as the bandwidth of the incoming signal to avoid aliasing distortion. This, of course, is merely a restatement of the Nyquist constraint that determines the minimum sampling frequency for distortionless recovery of the continuous-time signal.

Figure 1.3 shows an example of the aliasing that occurs when the sampling frequency is not great enough. The upper panel is an 8-kHz sinusoid with sample points superimposed by sampling at 10 kHz, which imposes an upper bound of 5 kHz for sampling without aliasing. The lower panel of the figure depicts the function that emerges from the sampling and reconstruction process: a 2-kHz signal. Note that the discrete-time sample points fit both the 8-kHz input signal and the 2-kHz output signal equally well. To prevent the effects of aliasing, it is common to precede a C/D converter by a lowpass filter with gain ideally equal to 1 and a cutoff frequency of $\Omega = \pi/T_1$.

Relating continuous-time and discrete-time frequency. The relationship between $X_s(j\Omega)$, the CTFT of $x_s(t)$, and $X(e^{j\omega})$, the discrete-time Fourier transform (DTFT) of $x[n]$, is a bit subtle. From the definition of the DTFT, $X(e^{j\omega})$ is

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} = \sum_{n=-\infty}^{\infty} x_c(nT_1)e^{-j\omega n} \tag{1.8}$$

Writing the definition of $X_s(j\Omega)$ and incorporating Eq. (1.2) produces

$$X_s(j\Omega) = \int_{-\infty}^{\infty} x_s(t)e^{-j\Omega t}dt = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x_c(nT_1)\delta(t-nT_1)e^{-j\Omega t}dt \tag{1.9}$$

Interchanging the sum and the integral and applying the procedures for integrating expressions with delta functions (see the Appendix for a detailed discussion of integration with delta functions) produces

$$X_s(j\Omega) = \sum_{n=-\infty}^{\infty} x_c(nT_1)\int_{-\infty}^{\infty} \delta(t-nT_1)\, e^{-j\Omega t}dt = \sum_{n=-\infty}^{\infty} x_c(nT_1)e^{-j\Omega nT_1} \tag{1.10}$$

We note that the final terms of Eqs. (1.8) and (1.10) are identical, except that

$$\omega = \Omega T_1 \tag{1.11}$$

where again T_1 is the sampling period. This means that $X(e^{j\omega})$, the DTFT of $x[n]$, is identical to $X_s(j\Omega)$, the CTFT of $x_s(t)$, except that the frequency axis is scaled by the sampling period T_1.[2] (Although it is easy to confuse discrete-time and continuous-time frequency in expressions like Eq. (1.11), keep in mind the dimensional analysis that discrete-time frequency in radians is equal to continuous-time frequency in radians per second times the sampling period in seconds.)

[2] If $X_s(j\Omega)$ includes delta functions, the areas of the delta functions would be scaled according to the relationship $\delta(at) = (1/|a|)\delta(t)$, as discussed in the Appendix.

1.3 Reconstruction of continuous-time signals from their samples

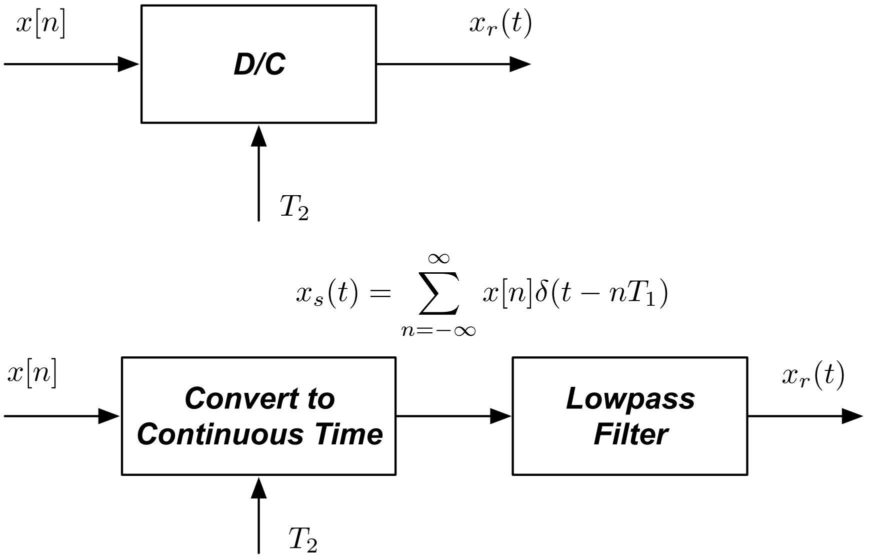

$x[n]$
D/C
$x_r(t)$
T_2
$x_s(t) = \sum_{n=-\infty}^{\infty} x[n]\delta(t - nT_1)$
$x[n]$
Convert to Continuous Time
Lowpass Filter
$x_r(t)$
T_2

Figure 1.4: Upper panel: block diagram of D/C conversion. Lower panel: Specification of the underlying signal processing.

Let us now turn our attention to the reconstruction process, which is in some ways a reversal of the sampling process. Figure 1.4 summarizes the major processing steps. The discrete-time sequence $x[n]$ is converted into the continuous-time sequence of delta functions $x_s(t)$, in which the areas of the impulses in continuous time are equal to the corresponding amplitudes of the discrete-time samples:

$$x_s(t) = \sum_{n=-\infty}^{\infty} x[n]\delta(t - nT_2) \tag{1.12}$$

This sequence of delta functions is passed through an ideal lowpass filter with transfer function

$$H_{LP}(j\Omega) = \begin{cases} T_2, & |\Omega| < \pi/T_2 \\ 0, & \text{otherwise} \end{cases} \tag{1.13}$$

The gain factor of T_2 in $H_{LP}(j\Omega)$ ensures that if the original signal is sampled with a great enough sampling frequency to avoid aliasing, and if the sampling period T_1 equals the time T_2 between the samples used in reconstruction, then the continuous-time recovered signal $x_r(t)$ will be identical to the original input $x_c(t)$. In effect, when $T_1 = T_2$ the gain factor of T_2 in the lowpass filter in the D/C converter is intended to compensate for the factor of $1/T_1$ that incurred in the C/D conversion process. If $T_2 \neq T_1$, the signal passing through will nominally incur a gain of T_2/T_1.

The reconstruction process is illustrated in the frequency domain in Fig. 1.5 which can be seen to retrace most of the functions shown in Fig. 1.2 in reverse order. In this figure we assume that the reconstruction sample period T_2 is equal to the original sampling period T_1, although this is not always the case. The discrete-time sequence $x[n]$, depicted in the upper panel of Fig. 1.5 is first converted into the continuous-time sequence of

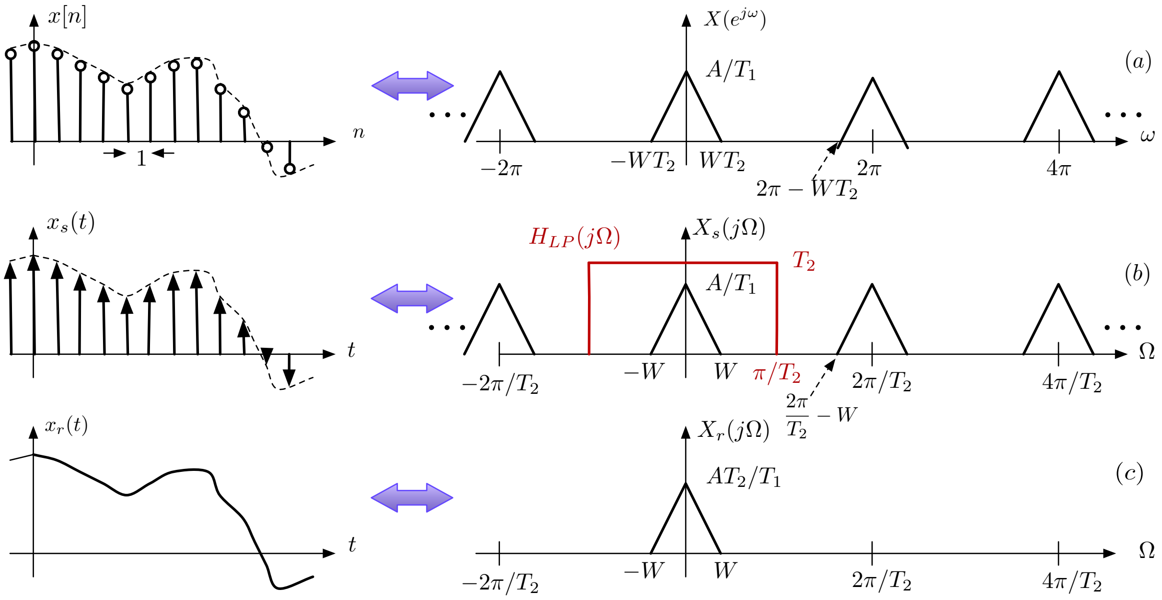


Figure 1.5: Reconstruction of a continuous-time function from its samples. Left side: the time functions (a) $x[n]$, (b) $x_s(t)$, and (d) $x_r(t)$. Right side: The corresponding Fourier transforms in discrete and continuous time. The ideal lowpass filter $H_{LP}(j\Omega)$ is also shown in panel (b) in red. In plotting, we assume that $T_2 = T_1$ in Fig. 1.2.

delta functions, $x_s(t)$, which is depicted in the central panel of Fig 1.5. As discussed in conjunction with Eq. (1.8) through Eq. (1.11), $X_s(j\Omega)$, the CTFT of $x_s(t)$, is identical to $X(e^{j\omega})$, the DTFT of $x[n]$ except that frequencies are scaled according to the relationship that $\omega = \Omega T_2$, which produces an infinite train of replications of the original frequency response, as in the central panel of Fig. 1.5. The lowpass reconstruction filter is depicted in red in the central panel of Fig. 1.5, and the product of $X_s(j\Omega)$ and $H_{LP}(j\Omega)$ is the single-mode spectrum $X_r(j\Omega)$ which is (ideally) equal to the original input spectrum $X_c(j\Omega)$ depicted in Fig. 1.2.

It is helpful to consider what is going on in the time domain to understand the nature of the reconstruction of the continuous-time output $x_r(t)$ from the sequence of delta functions $x_s(t)$. The output in the time domain is (as usual) the convolution of the input with the unit impulse response of the filter:

$$x_r(t) = x_s(t) * h_{LP}(t) \tag{1.14}$$

The unit impulse response of the filter is easily obtained directly:

$$h_{LP}(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H_{LP}(j\Omega)e^{j\Omega t}d\Omega = \frac{1}{2\pi}\int_{-\pi/T_2}^{\pi/T_2} T_2 e^{j\Omega t}d\Omega = \frac{\sin(\pi t/T_2)}{\pi t/T_2} \tag{1.15}$$

This is a continuous-time sinc function with amplitude equal to 1 at $t = 0$ and regularly-spaced zero crossings at $t = nT_2$. Because the convolution of any function with a train of impulses produces replication of the original function at the times of the impulses, weighted by the impulse areas, we obtain

$$x_r(t) = h_{LP}(t) * x_s(t) = h_{LP}(t) * \sum_{n=-\infty}^{\infty} x[n]\delta(t - nT_2) = \sum_{n=-\infty}^{\infty} x[n]\frac{\sin(\pi(t-nT_2)/T_2)}{\pi(t-nT_2)/T_2} \tag{1.16}$$

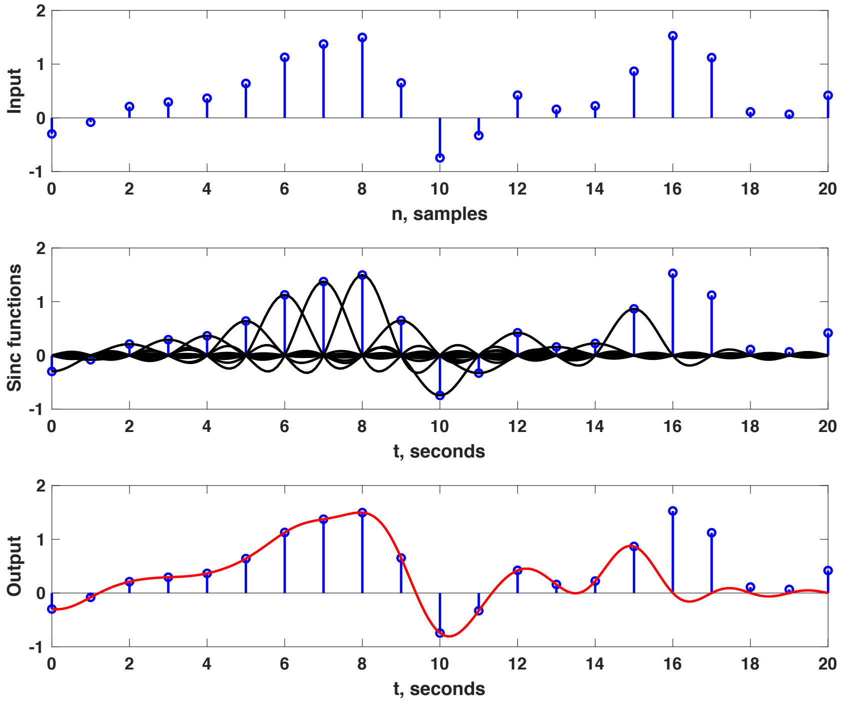


Figure 1.6: Upper panel: a segment of a discrete-time signal. Center panel: representation of that signal in continuous time by a linear combination of sinc functions. Lower panel: the sum of the sinc functions as of $t = 15$ seconds.

In other words, the reconstructed continuous-time function $x_r(t)$ can be represented as the sum of an infinite train of sinc functions, delayed by intervals of T_2 and scaled according to the corresponding value of $x[n]$. This is illustrated in Fig. 1.6, which uses the convenient but unrealistic value of $T_2 = 1$ second. The upper panel of Fig. 1.6 depicts a short segment of a discrete-time signal. The central panel shows its reconstruction as a series of weighted and delayed sinc functions. Note that the reconstruction is exact at the time of the original samples as the sinc function centered each sample point has an amplitude of 1 while the other sinc functions are all zero. The values of the reconstructed signal (lower panel) between the sample points are obtained by summing all the sinc functions together. While there is an infinite number of sinc functions, they decay away fairly rapidly. The lower two panels of Fig. 1.6 are computed only through 15 seconds. It can be seen that the reconstruction of $x_r(t)$ appears to be accurate except for values of t that are near to or greater than 15 seconds.

In practice, of course, the actual lowpass filter to reconstruct the continuous-time signal would not be ideal for multiple reasons including the fact that all ideal filters are fundamentally unrealizable. Nevertheless, the reconstructed signal would still be an infinite linear combination of the unit impulse response of the lowpass filter that is actually used.

1.4 Introduction to discrete-time decimation and interpolation

It is frequently necessary or desirable to change the effective sampling rate of discrete-time signals. We consider three standard ways of manipulating the effective sampling frequency in discrete time:

- *Decimation* or *downsampling* the signal by an integer factor of M, which decreases the effective sampling rate by a factor of M
- *Interpolation* or *upsampling* the signal by an integer factor L, which increases the effective sampling rate by a factor of L
- A combination of interpolation and decimation which changes the effective sampling rate of the signal by the rational factor of L/M.

There are multiple potential motivations for changing the sampling rate. For example, the sampling rate may be needed to combine signals that had been recorded at different different sampling rates, as would be necessary if audio from a CD recorded at the standard rate of 44.1 kHz were used as background music for a film, which uses 48 kHz as the standard sampling rate. Multi-rate signal processing is useful for delaying a signal by a fractional number of samples. For example, a delay of 1/4 sample can be accomplished by upsampling a signal by a factor of 4, delaying by a single sample, and then downsampling by a factor of 4. Finally, multi-rate signal processing techniques are frequently useful in the design of filters that must have a very narrow passband (or stopband) with narrow transition bands.

Figure 1.7 illustrates downsampling and upsampling for a short segment of a discrete-time speech signal. The original signal is depicted in the upper panel, while the central panel shows the signal after downsampling by a factor of 3, and the lower panel shows the signal after upsampling by a factor of 2. The solid curve in red shows the original continuous-time signal from which the discrete-time signal was derived. The left column shows the three functions with the horizontal axes scaled to maintain the continuous-time envelope unchanged, while the right column plots the three discrete-time samples in a fashion that preserves the spacing between the samples. It can be seen that the decimation and interpolation operations can be interpreted either as a decrease or increase of the effective sampling rate of the discrete-time functions relative to the continuous-time functions (as in the left column) or as an intrinsic compression or expansion in time of the discrete-time functions (as in the right column).

In the following sections we describe and discuss the fundamental mathematics of discrete-time decimation, interpolation, and change of sample rate by a rational factor. We will find that the operations associated with decimation and interpolation are closely related to ideal C/D and D/C conversion, respectively, that had been described in Secs. 1.2 and 1.3 above. We will discuss computationally-efficient ways of realizing these operations in Chapter 2 below.

1.4.1 Decimation: reducing the sampling rate by an integer factor M

In continuous-time signal processing, Fourier transform properties relate compression in time to expansion in frequency and vice-versa. Specifically, if

$$x(t) \Leftrightarrow X(j\Omega) \tag{1.17}$$

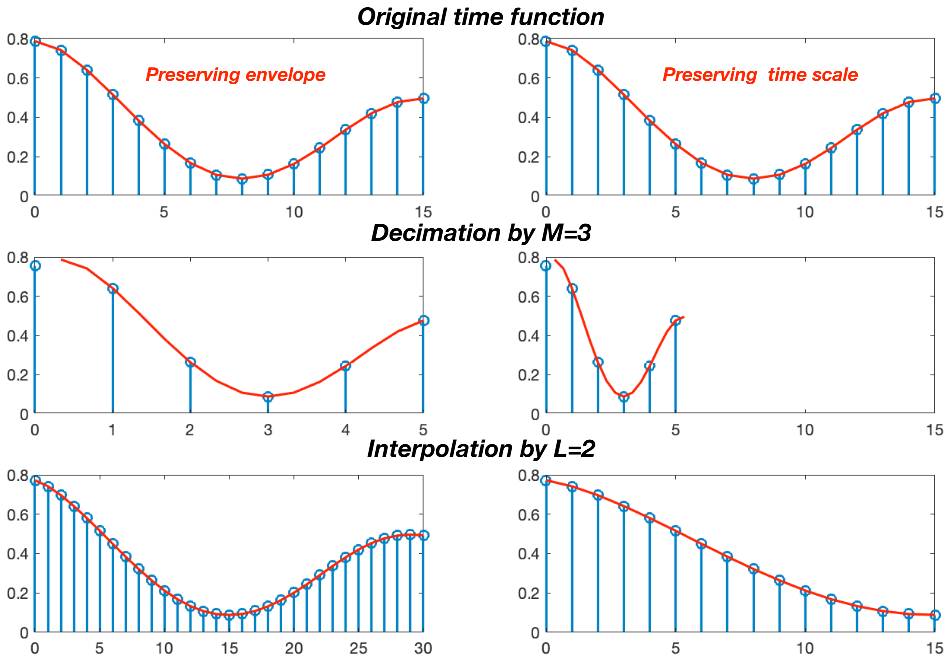


Figure 1.7: Comparison of an original discrete-time function (upper row), the function after decimation by $M = 3$ (central row), and the function after interpolation by $L = 2$ (lower row). Left column: the spacing of the samples is scaled according to the change in sample rate. Right column: the spacing of the samples is constant across all three sample rates.

then

$$x(at) \Leftrightarrow \frac{1}{|a|} X\left(j \frac{\Omega}{|a|}\right) \tag{1.18}$$

As we noted above, we can think of decimation as a process by which we compress in the signal along the time axis. Nevertheless, the time-compression is not straightforward because of the constraint that discrete-time signals are meaningful only for sample values that are integer. For example, if we were downsampling by a factor of 2, each of the odd samples of the original time function would disappear after decimation. In order to represent the impact of the samples that do not "survive" the decimation process on the frequency response, we must set those samples to zero explicitly in our mathematical analysis of the processing. Consequently, we mathematically describe the decimation process as in lower panel of the block diagrams in Fig. 1.8. The upper panel of this figure shows the basic structure of a system that performs decimation or downsampling by a factor of M. The input is a discrete-time function $x[n]$, which we assume to be a bandlimited time function such that its discrete-time-time Fourier transform $X(e^{j\omega})$ equals zero for $W < |\omega| \leq \pi$. The output is the decimated discrete-time function $x_d[m]$ which is equal to $x[n]$ evaluated at times $n = mM$, where M is the downsampling ratio. (We use the index m rather than n to call attention to the fact that the effective time base changes after decimation.)

The mathematical representation of the decimation process is specified in more detail in

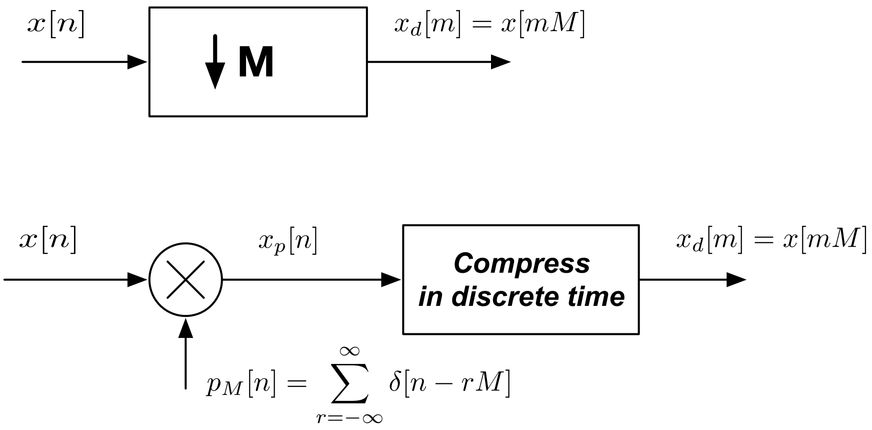


Figure 1.8: Upper panel: block diagram of decimation by M. Lower panel: Specification of the underlying signal processing.

the lower panel of Fig. 1.8. Specifically, the input is first multiplied by $p_M[n]$, an infinite train of delta functions in discrete time of amplitude 1, which are separated by an interval of M samples. The function $x_p[n]$ is the product of $x[m]$ and $p_M[n]$, and is a train of delta functions of amplitude $x[nM]$ which are separated in time by intervals of M samples, with the values of $x_p[n]$ set equal to zero when n is not an integer multiple of M. This sequence is then compressed in time so that $x_d[m] = x[mM]$.

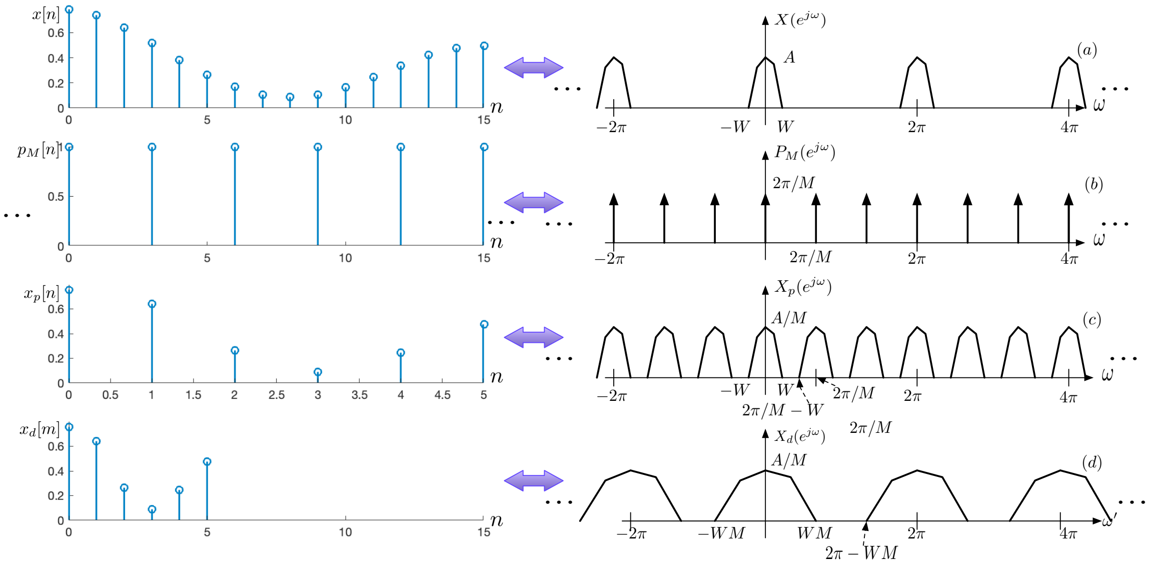


Figure 1.9: Representative time functions and their spectra in discrete-time decimation, illustrated for $M = 3$. Left side: the time functions (a) $x[n]$, (b) $p_M[n]$, (c) $x_p[n]$, and (d) $x_d[m]$. Right side: The corresponding discrete-time Fourier transforms.

Time-domain representations. It is helpful to be able to visualize these signals in both time and frequency. The left side of Fig. 1.9 depicts, for an arbitrary time function and its spectrum, the various discrete-time functions depicted in Fig. 1.8 for the downsampling ratio $M = 3$. The corresponding Fourier transforms are shown on the right side of the figure. More specifically,

$$p_M[n] = \sum_{r=-\infty}^{\infty} \delta[n - rM] \tag{1.19}$$

$$x_p[n] = x[n]p_M[n] = \begin{cases} x[n], & n = rM \\ 0, & \text{otherwise} \end{cases} \tag{1.20}$$

The functions $x[n]$, $p_M[n]$, and $x_p[n]$ are depicted in panels (a), (b), and (c), respectively in the left column of Fig. 1.9. The function $x_p[n]$ consists of the samples of $x[n]$ that will survive the decimation process. The output of the decimation process, $x_d[m]$, consists of these same nonzero samples, but the time axis is now compressed so that the nonzero delta functions are now found at successive values of the new time index m.

Frequency-domain representations. Considering now the corresponding functions in the frequency domain, we will assume that $X(e^{j\omega})$, the DTFT of $x[n]$, is of arbitrary shape, bandlimited such that $X(e^{j\omega}) = 0$ for $W < |\omega| \leq \pi$, and with a maximum amplitude equal to A, as depicted in Fig. 1.9. Like all DTFTs, $X(e^{j\omega})$ is periodic with period 2π.

It is easy to demonstrate by computing the inverse DTFT that $P_M(e^{j\omega})$, the DTFT of $p_M[n]$, is also an infinite train of delta functions:

$$P_M(e^{j\omega}) = \frac{2\pi}{M} \sum_{k=-\infty}^{\infty} \delta\left(\omega - \frac{2\pi k}{M}\right) = \frac{2\pi}{M} \sum_{r=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \delta\left(\omega - \frac{2\pi l}{M} - 2\pi r\right) \tag{1.21}$$

$P_M(e^{j\omega})$ is expressed in double-sum form to emphasize the fact that the DTFT is periodic with period 2π and that there are M equally-spaced delta functions within each periodic cycle, separated by $\omega = 2\pi/M$, and each with amplitude $2\pi/M$. Because $x_p[n] = x[n]p_M[n]$ in the time domain, the corresponding DTFTs are

$$X_p(e^{j\omega}) = \frac{1}{2\pi} X(e^{j\omega}) \circledast P_M(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\theta}) P_M(e^{j(\omega-\theta)}) d\theta \tag{1.22}$$

where the symbol $\circledast$ indicates circular convolution. Because convolving $X(e^{j\omega})$ with the limited train of delta functions within a span of $\omega = 2\pi$ for $P_M(e^{j\omega})$ merely replicates and shifts the spectrum $M - 1$ additional times, we obtain

$$x_p(e^{j\omega}) = \frac{1}{2\pi} X(e^{j\omega}) \circledast \frac{2\pi}{M} \sum_{k=-\infty}^{\infty} \delta\left(\omega - \frac{2\pi k}{M}\right) = \frac{1}{M} \sum_{r=0}^{M-1} X(e^{j(\omega - \frac{2\pi r}{M}))}) \tag{1.23}$$

as shown in Row (c) of Fig. 1.9.

Avoiding aliasing distortion. In order for the downsampled discrete-time signal to be ultimately be recovered without distortion, it is necessary that the replications of the original spectrum in Fig. 1.9 not overlap one another. It can easily be seen that the separation of the replications is related to the downsampling ratio of M while the width of the replications is twice the signal bandwidth W. It is clear from Fig. 1.9 that overlap will be avoided if

$$\frac{2\pi}{M} - W > W \text{ or } W < \frac{\pi}{M} \tag{1.24}$$

In other words, the sampling frequency must be at least $2M$ as great as the bandwidth of the signal to avoid aliasing distortion. Most practical downsampling systems include a lowpass filter of gain 1 and a nominal cutoff frequency of π/M to avoid distortion by aliasing.

Changing the sampling rate. The final step in the decimation process is changing the time scale of the discrete-time function so that the non-zero samples have successive indices. Specifically, we define a new function and a new time axis $x_d[m] = x_p[mM]$. The DTFT of $x_d[m]$ can be computed directly:

$$X_d(e^{j\omega'}) = \sum_{m=-\infty}^{\infty} x_d[m]e^{-j\omega' m} = \sum_{m=-\infty}^{\infty} x_p[mM]e^{-j\omega' m} \tag{1.25}$$

Letting $l = mM$ or $m = l/M$ produces

$$X_d(e^{j\omega'}) = \sum_{l=-\infty}^{\infty} x_p[l]e^{-j\omega'(\frac{l}{M})} = \sum_{l=-\infty}^{\infty} x_p[l]e^{-j(\frac{\omega'}{M})l} = X_p\left(\frac{\omega'}{M}\right) \tag{1.26}$$

In other words, the output $X_d(e^{j\omega'})$ is identical to the DTFT $x_p(e^{j\omega})$, except that it is stretched in frequency to be wider by a factor of M. We use the symbol ω' to represent the frequency after downsampling to recognize the fact that the frequency scale has changed because the time scale has changed. This function is depicted in the right column of the bottom panel of Fig. 1.9. While you may be concerned that $m = l/M$ is not integer for some values of m, these are exactly the values of m for which the function $x_p[m]$ is equal to zero.

1.4.2 Interpolation: increasing the sampling rate by an integer factor L

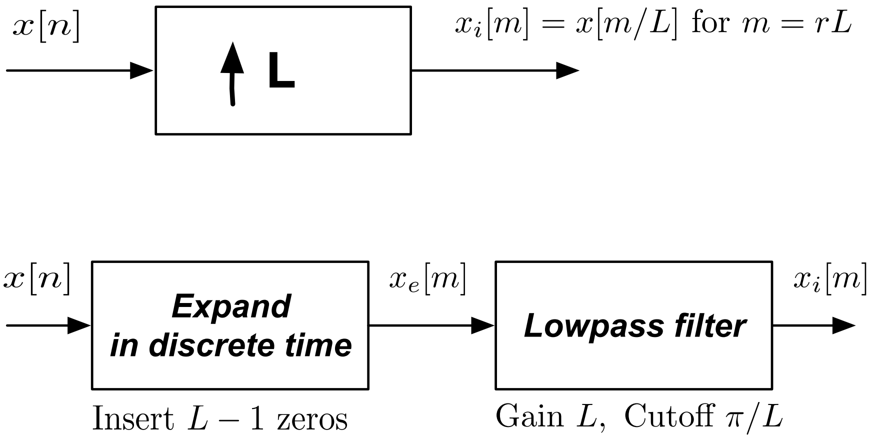


Figure 1.10: Upper panel: block diagram of interpolation by L. Lower panel: Specification of the underlying signal processing.

Just as decimation or downsampling is similar in some ways to C/D conversion, discrete-time interpolation or upsampling has some similarities to D/C conversion, which is at the end of the day itself an interpolation process in continuous time. Interpolation by a factor of L includes two major steps: (1) "expanding" the discrete-time function by inserting $L-1$ samples of amplitude zero between each successive sample of the input and (2) lowpass filtering the resulting signal using an ideal filter with gain L and cutoff frequency π/L.

Expansion in time. Figure 1.10 depicts the major functions involved with interpolation in time and frequency. Using $x[n]$ to designate the input, we will represent the expanded time function as

$$x_e[m] = \begin{cases} x[m/L], & m/L \text{ integer} \\ 0, & \text{otherwise} \end{cases} \tag{1.27}$$

to produce the interpolated sequence $x_i[m]$ as the filter output. As stated above, we will use the variable m rather than n for the time index as a reminder that the time scale has changed.

The discrete-time sequence $x_e[m]$ that represents the expanded signal can also be written as

$$x_e[m] = \sum_{l=-\infty}^{\infty} x[l]\delta[m - lL] \tag{1.28}$$

This sequence of weighted delta functions is passed through an ideal lowpass filter with transfer function

$$H_{LP}(e^{j\omega}) = \begin{cases} L, & |\omega| < \pi/L \\ 0, & \pi/L < |\omega| \leq \pi \end{cases} \tag{1.29}$$

The gain factor of L in $H_{LP}(e^{j\omega})$ ensures that if an original signal is upsampled and then downsampled by the same ratio, the original signal will be recovered without distortion and with the same amplitude. There is no need to be concerned with aliasing distortion in the interpolation process because no information is lost.

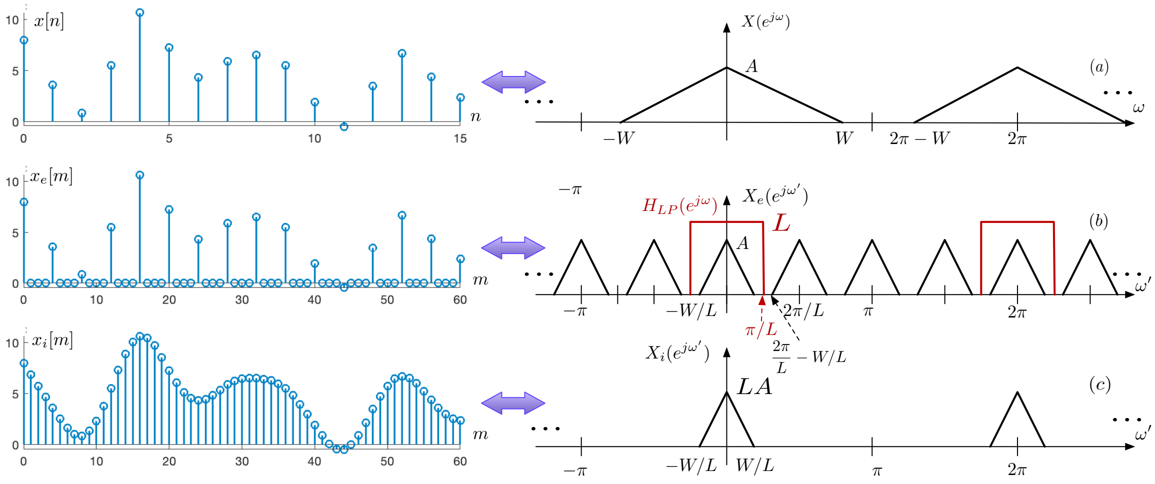


Figure 1.11: Functions involved in process of interpolation by a factor of $L = 4$ in the time and frequency domains. Left side: the time functions (a) $x[n]$, (b) $x_e[m]$, and (c) $x_r[m]$. Right side: The corresponding Fourier transforms in discrete time. The ideal lowpass filter $H_{LP}(e^{j\omega})$ is also shown in panel (b) in red.

The upsampling process is illustrated in the time and frequency domains in Fig. 1.11, which can be seen to retrace most of the functions shown in Fig. 1.9 in reverse order. The discrete-time sequence $x[n]$, depicted in the upper panel of Fig. 1.11 is first converted into the expanded sequence of delta functions $x_e[m]$, which is depicted in the central panel of Fig 1.9, as describe above. It is easy to demonstrate that the $X_e(e^{j\omega'})$, the DTFT of $x_e[m]$ is identical to $X(e^{j\omega})$, the DTFT of $x[m]$ but contracted in frequency by a factor of L:

$$X_e(j\omega') = \sum_{m=-\infty}^{\infty} x_e[m]e^{-j\omega' m} = \sum_{m=-\infty, m=rL}^{\infty} x_d[m/L]e^{-j\omega' m} \tag{1.30}$$

Letting $l = m/L$ or $m = lL$, we obtain

$$X_e(j\omega') = \sum_{m=-\infty, m=rL}^{\infty} x[m/L]e^{-j\omega' m} = \sum_{l=-\infty}^{\infty} x[l]e^{-j\omega' lL} = \sum_{l=-\infty}^{\infty} x[l]e^{-j(\omega' L)l} = X(e^{j\omega' L}) \tag{1.31}$$

Lowpass smoothing. The output of the decimation process is obtained by passing the expanded signal $x_e[m]$ through the lowpass filter. In the time domain we convolve $x_e[m]$ with the unit sample response of the filter:

$$x_i[m] = x_e[m] * h_{LP}[m] \tag{1.32}$$

The unit sample response of the filter is easily obtained directly:

$$h_{LP}[m] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_{LP}(e^{j\omega})e^{j\omega n}d\omega = \frac{1}{2\pi}\int_{-\pi/L}^{\pi/L} Le^{j\Omega m}d\Omega = L\frac{\sin(\pi m/L)}{\pi m/L} \tag{1.33}$$

This is a discrete-time sinc function with amplitude equal to L at $m = 0$ and regularly-spaced zero crossings at $m = rL$. Because the convolution of any function with a train of impulses produces replication of the original function at the times of the impulses, weighted by the impulse areas, we obtain

$$x_i[m] = h_{LP}[m] * x_e[m] = h_{LP}[m] * \sum_{r=-\infty}^{\infty} x[r]\delta(m - rL) = L\sum_{r=-\infty}^{\infty} x[r]\frac{\sin(\pi(m - rL)/L)}{\pi(m - rL)/L} \tag{1.34}$$

In other words, the interpolated discrete-time function $x_i[m]$ can be represented as the sum of an infinite train of sinc functions in discrete time, delayed by intervals of L and scaled according to the corresponding value of $x[m]$. The interpolation process works exactly the same way as the reconstruction of continuous-time functions in D/C conversion, as illustrated in Fig. 1.11, except that the resulting function $x_i[m]$ is discrete in time rather than continuous in time.

As in C/D conversion, the actual lowpass filter to reconstruct the interpolated signal would not be ideal. Nevertheless, the interpolated signal would still be an infinite linear combination of delayed and scaled unit sample response of the lowpass filter that is actually used.

In the frequency domain, the the lowpass filter limits the frequencies passed to $|\omega'| \le \pi/L$, as depicted in red in the central panel of Fig. 1.11. As depicted in the lower panel of Fig. 1.11, the output frequency response is

$$X_i(e^{j\omega'}) = \begin{cases} X(e^{j\omega'}), & |\omega'| \le \pi/L \\ 0, & \pi/L < |\omega'| \le \pi \end{cases} \tag{1.35}$$

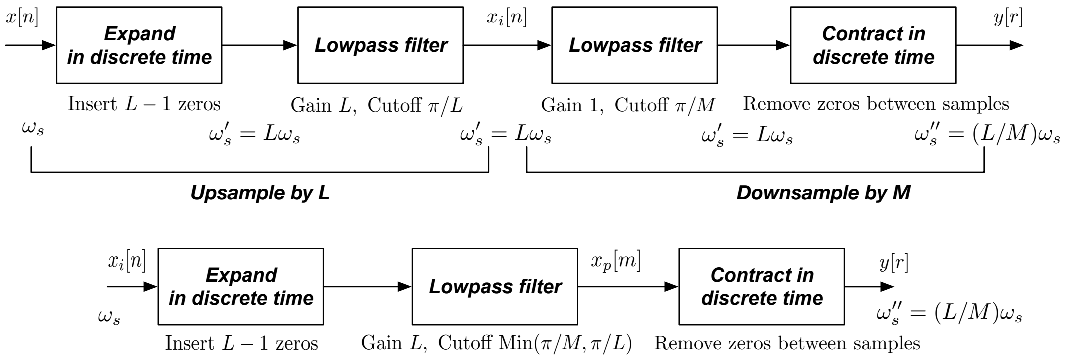


Figure 1.12: Block diagram of a system that changes the sampling rate by the rational fraction L/M. Upper panel: direct implementation by cascading interpolation by L with decimation by M. Lower panel: integrated system in which the cascade of two lowpass filters is replaced by a single lowpass filter.

1.4.3 Changing the sampling rate by a rational factor L/M

Changing the effective sampling rate by a rational factor of L/M can be thought of as first upsampling by L and then downsampling by M. For example, to change the sample rate of a speech signal from 20 kHz to 16 kHz, we would first upsample by $L = 4$ to 80 kHz and then downsample by $M = 5$ to 16 kHz. Note that 80 is the least common multiple of the upsampling and downsampling ratios L and M.

Figure 1.12 depicts two implementations of a system that changes the effective sampling rate by the fraction L/M. The upper panel shows the direct implementation of such a system, consisting of the cascade of upsampling by L followed by decimation by a factor of M. The lowpass antialiasing filter that is part of most practical decimation systems is included explicitly in the figure. In addition to the change in sampling rate, the amplitude of the input will be scaled by the factor of L/M by the processing. As discussed in the section the ideal antialiasing filter prior to decimation has a gain of 1 and a cutoff frequency of π/M. The lowpass filter that performs the interpolation in the upsampling module has a gain of L and a cutoff frequency of π/L. The initial block in the upper panel simply inserts $L-1$ zeros between each successive sample of the input. The final block contracts its input in time by a factor of M, preserving only those samples for which the input sample index is an integer multiple of M.

The lower panel of Fig. 1.12 depicts a more efficient implementation that replaces the cascade of the two ideal lowpass filters. It will have a gain of L and a cutoff frequency that is the minimum of π/L and π/M.

In summary, we have reviewed in this chapter the mathematics that are used to describe the processes of ideal conversion of signals from continuous time to discrete time and vice versa, known as ideal C/D and D/C conversion. We also discussed the basic mathematics that describe discrete-time change in sampling rate, including interpolation (or upsampling), decimation (or downsampling), and change of sampling rate by a rational fraction. We noted that decimation is similar to C/D conversion in a number of respects and that interpolation is similar to D/C conversion.

In the following chapter we will discuss techniques used to make the upsampling and downsampling operations much more computationally efficient.