Chapter 9

Multirate Digital Signal Processing

9.1 Introduction

Multirate systems have gained popularity since the early 1980s and they are commonly used for audio and video processing, communications systems, and transform analysis to name but a few. In most applications multirate systems are used to improve the performance, or for increased computational efficiency. The two basic operations in a multirate system are decreasing (*decimation*) and increasing (*interpolation*) the sampling-rate of a signal. Multirate systems are sometimes used for *sampling-rate conversion*, which involves both decimation and interpolation.

9.2 Decimation

Decimation can be regarded as the discrete-time counterpart of sampling. Whereas in sampling we start with a continuous-time signal $x(t)$ and convert it into a sequence of samples $x[n]$, in decimation we start with a discrete-time signal $x[n]$ and convert it into another discrete-time signal $y[n]$, which consists of *sub-samples* of $x[n]$. Thus, the formal definition of M-fold decimation, or down-sampling, is defined by Equation 9.1. In decimation, the sampling rate is reduced from F_s to F_s/M by discarding $M - 1$ samples for every M samples in the original sequence.

$$y[n] = v[nM] = \sum_{k=-\infty}^{\infty} h[k]x[nM - k] \tag{9.1}$$

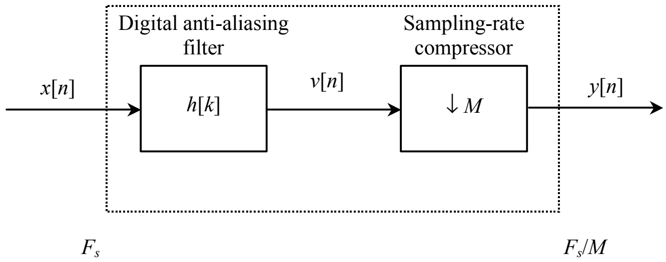

Figure 9.1: Block diagram notation of decimation, by a factor of M.

The block diagram notation of the decimation process is depicted in Figure 9.1. An anti-aliasing digital filter precedes the down-sampler to prevent aliasing from occurring, due to the lower sampling rate. The subject of aliasing in decimated signals is covered in more detail in Section 9.4. In Figure 9.2 below, it illustrates the concept of 3-fold decimation i.e. $M = 3$. Here, the samples of $x[n]$ corresponding to $n = \ldots, -2, 1, 4, \ldots$ and $n = \ldots, -1, 2, 5, \ldots$ are lost in the decimation process. In general, the samples of $x[n]$ corresponding to $n \neq kM$, where k is an integer, are discarded in M-fold decimation. In Figure 9.2 (b), it shows samples of the decimated signal $y[n]$ spaced three times wider than the samples of $x[n]$. This is not a coincidence. In real time, the decimated signal appears at a slower rate than that of the original signal by a factor of M. If the sampling frequency of $x[n]$ is F_s, then that of $y[n]$ is F_s/M.

9.3 Interpolation

Interpolation is the exact opposite of decimation. It is an information preserving operation, in that all samples of $x[n]$ are present in the expanded signal $y[n]$. The mathematical definition of L-fold interpolation is defined by Equation 9.2 and the block diagram notation is depicted in Figure 9.3. Interpolation works by inserting (L–1) zero-valued samples for each input sample. The sampling rate therefore increases from F_s to LF_s. With reference to Figure 9.3, the expansion process is followed by a unique digital low-pass filter called an *anti-imaging filter*. Although the expansion process does not cause aliasing in the interpolated signal, it does however yield undesirable replicas in the signal's frequency spectrum. We shall see how this special filter, in Section 9.4, is necessary to remove these replicas from the frequency spectrum.

$$y[n] = L\sum_{k=-\infty}^{\infty} h[k]w[n-k] \tag{9.2}$$

Where, $$w[n] = \begin{cases} x[n/L] & \text{, if } n/\text{L is an integer} \\ 0 & \text{, if } n/\text{L is non - integer} \end{cases}$$

In Figure 9.4 below, it depicts 3-fold interpolation of the signal $x[n]$ i.e. L = 3. The insertion of zeros effectively attenuates the signal by L, so the output of the anti-imaging filter must be multiplied by L, to maintain the same signal magnitude.

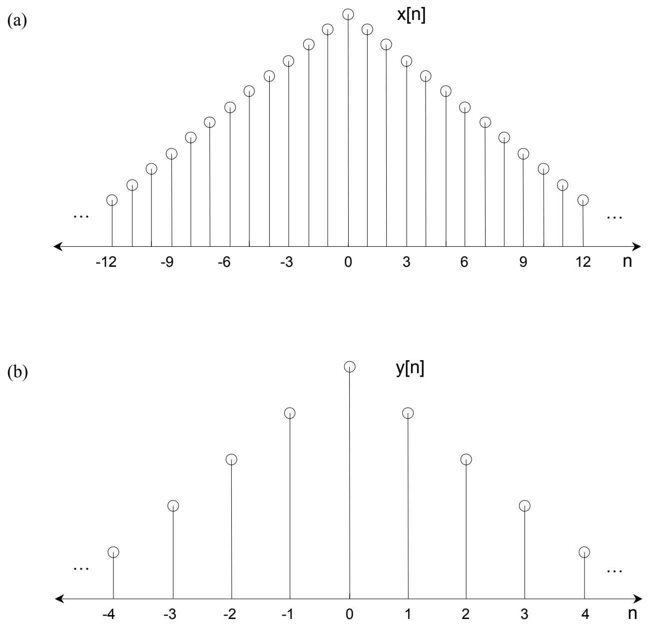

Figure 9.2: Decimation of a discrete-time signal by a factor of 3.

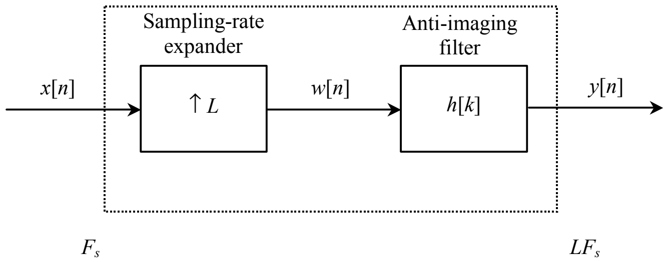

Figure 9.3: Block diagram notation of interpolation, by a factor of L.

9.4 Frequency Transforms of Decimated and Expanded Sequences

The analysis of decimation and expansion is better understood by assessing their respective frequency spectrums using the Fourier transform.

9.4.1 Decimation

The implications of aliasing caused by decimation are very similar to those in the case of sampling a continuous-time signal, in Section 1.3. In general, if the Fourier transform of a signal, $X(\theta)$, occupies the entire bandwidth from [$-\pi$, π], then the Fourier transform of the decimated signal, $X_{(\downarrow M)}(\theta)$, will be aliased. This is due to the superposition of the M shifted and frequency-scaled transforms. This is illustrated in Figure 9.5 below, which shows the aliasing phenomenon for M = 3.

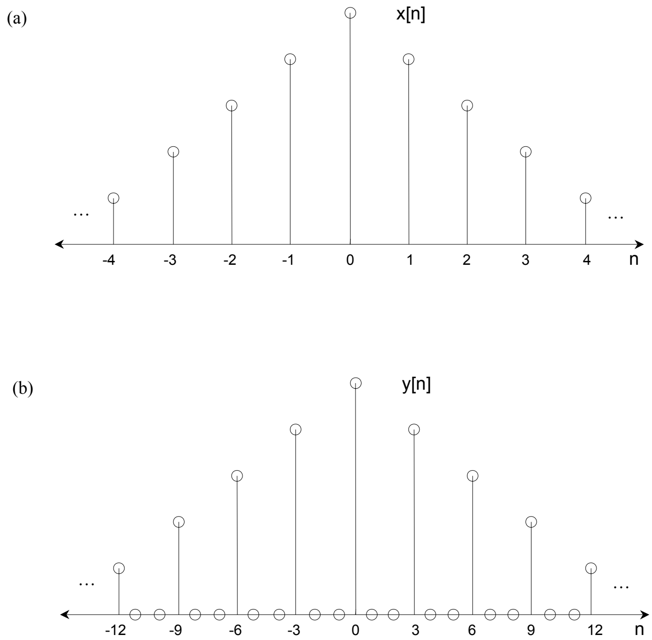

Figure 9.4: Interpolation of a discrete-time signal by a factor of 3.

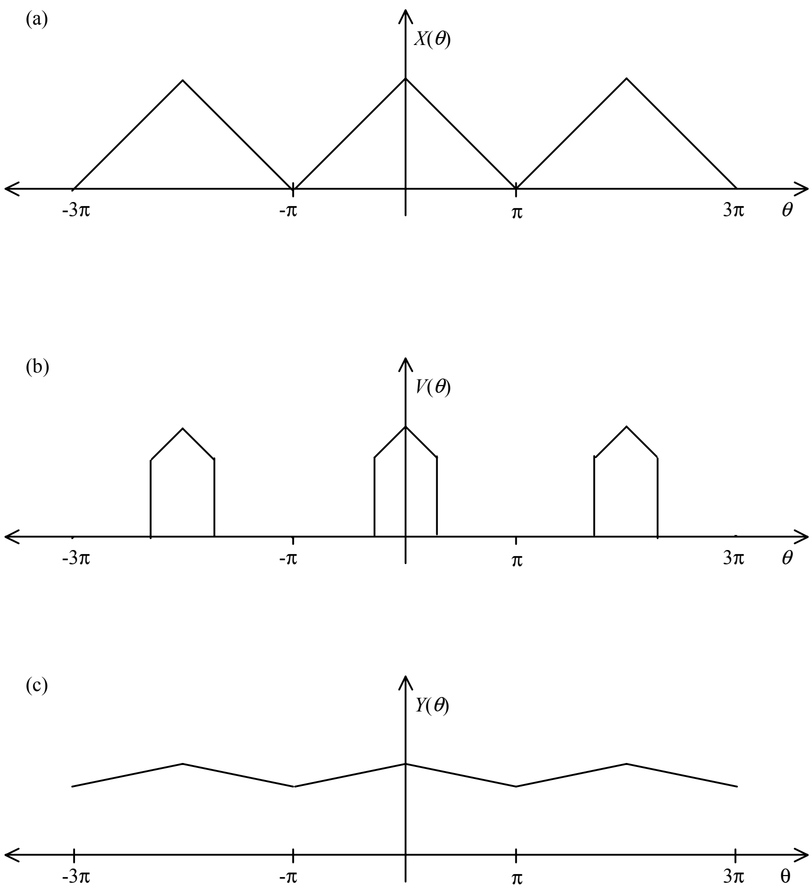

Figure 9.5: Aliasing caused by decimation; (a) Fourier transform of the original signal; (b) After decimation filtering; (c) Fourier transform of the decimated signal.

In Figure 9.5 (a) it shows the Fourier transform of the original signal. Part (b) shows the signal after lowpass filtering. In Figure 9.5 (c), it depicts the expanded spectrum after decimation.

9.4.2 Expansion

The effect of expansion on a signal in the frequency domain is illustrated in Figure 9.6 below. Part (a) shows the Fourier transform of the original signal; part (b) illustrates the Fourier transform of the signal with zeros added $W(\theta)$; and part (c) shows the Fourier transform of the signal after the interpolation filter. It is clearly visible that the shape of the Fourier transform is compressed by a factor L in the frequency axis and is also repeated L times in the range of $[-\pi, \pi]$. Despite the compression of the signal in the frequency axis, the shape of the Fourier transform is still preserved, confirming that expansion does not lead to aliasing. These replicas are removed by a digital low-pass filter called an *anti-imaging* filter, as indicated in Figure 9.3.

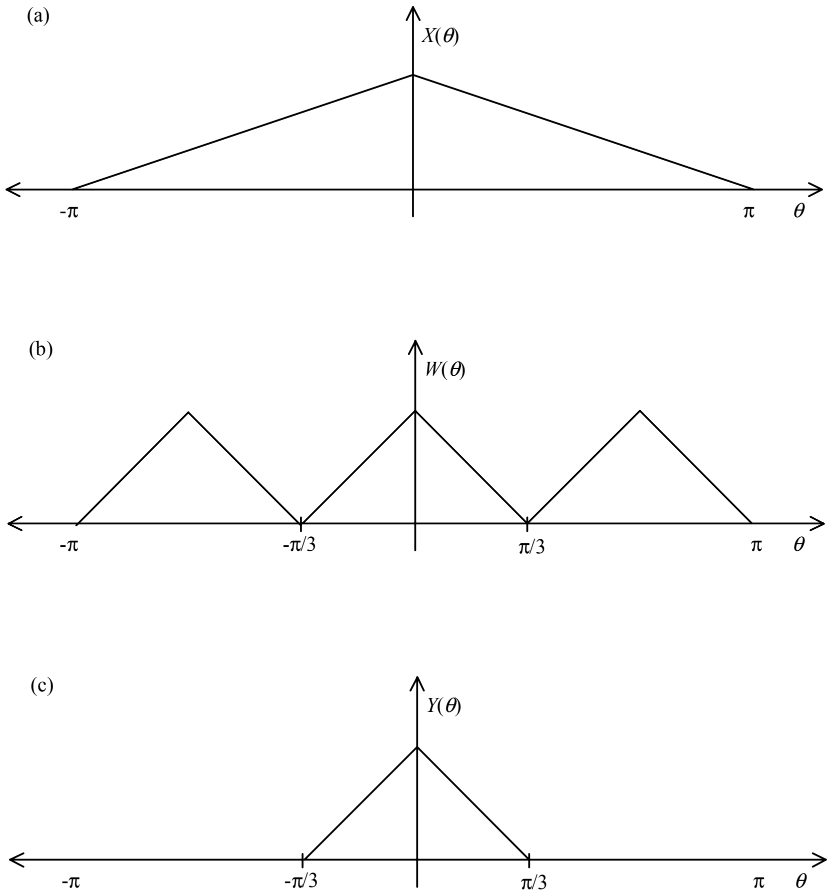

Figure 9.6: Expansion in the frequency domain of the original signal (a) and the expanded signal (b).

9.5 Sampling-rate Conversion

A common use of multirate signal processing is for sampling-rate conversion. Suppose a digital signal $x[n]$ is sampled at an interval T_1, and we wish to obtain a signal $y[n]$ sampled at an interval T_2. Then the techniques of decimation and interpolation enable this operation, providing the ratio T_1/T_2 is a rational number i.e. L/M.

Sampling-rate conversion can be accomplished by L-fold expansion, followed by low-pass filtering and then M-fold decimation, as depicted in Figure 9.7. It is important to emphasis that the interpolation should be performed first and decimation second, to preserve the desired spectral characteristics of $x[n]$. Furthermore by cascading the two in this manner, both of the filters can be combined into one single low-pass filter.

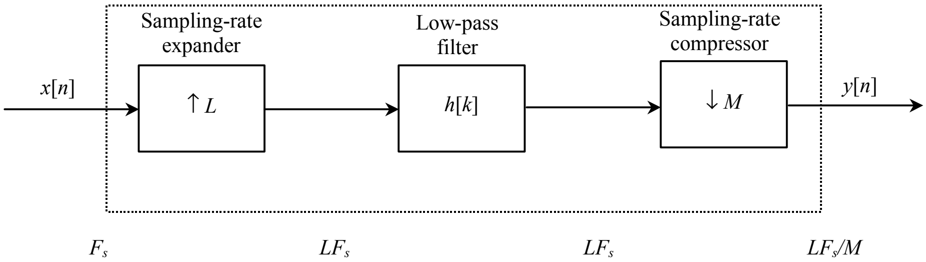

Figure 9.7: Sampling-rate conversion by expansion, filtering, and decimation.

An example of sampling-rate conversion would take place when data from a CD is transferred onto a DAT. Here the sampling-rate is increased from 44.1 kHz to 48 kHz. To enable this process the non-integer factor has to be approximated by a rational number:

$$\frac{L}{M}=\frac{48}{44.1}=\frac{160}{147}=1.08844$$

Hence, the sampling-rate conversion is achieved by interpolating by L i.e. from 44.1 kHz to [44.1x160] = 7056 kHz. Then decimating by M i.e. from 7056 kHz to [7056/147] = 48 kHz.

9.6 Multistage Approach

When the sampling-rate changes are large, it is often better to perform the operation in multiple stages, where $M_i(L_i)$, an integer, is the factor for the stage i.

$$M = M_1 M_2 \ldots M_I \quad \text{or} \quad L = L_1 L_2 \ldots L_I$$

An example of the multistage approach for decimation is shown in Figure 9.8. The multistage approach allows a significant relaxation of the anti-alias and anti-imaging filters, with a consequent reduction in the filter complexity. The optimum number of stages is one that leads to the least computational effort in terms of either the multiplications per second (MPS), or the total storage requirement (TSR).

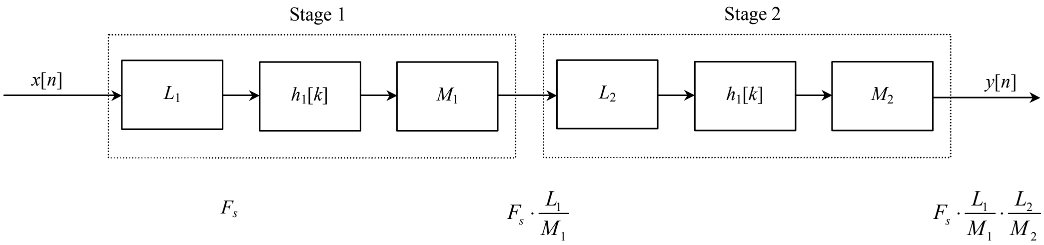

Figure 9.8: Multistage approach for the decimation process.

9.7 Polyphase Filters

Potential computational savings can be made within the process of decimation, interpolation, and sampling-rate conversion. *Polyphase filters* is the name given to certain realisations of multirate filtering operations, which facilitate computational savings in both hardware and software.

As an example, the combined low-pass filter in the sampling-rate converter, as illustrated in Figure 9.7, can be re-drawn as a realisation structure (Chapter 3). In principle, the simplest realisation of the low-pass filter is the direct-form FIR structure, as depicted in Figure 9.9. However, this type of structure is very inefficient owing to the interpolation process, which introduces (L–1) zeros between consecutive points in the signal. If L is large, then the majority of the signal components fed into the FIR filter are zero. As a result, most of the multiplications and additions are zero i.e. many pointless calculations. Furthermore, the decimation process itself implies that only one out of every M output samples is required at the output of the sampling-rate converter. Consequently, only one out of every M possible values at the output of the filter needs to be computed. This type of structure therefore, leads to much inefficiency during the process of sampling-rate conversion.

A more efficient realisation structure of the sampling-rate converter uses polyphase filters, as illustrated in

Figure 9.10. It takes into account that after the interpolation process the signal consists of (L–1) zero coefficients, and the decimation process implies that only one out of every M samples is required at the output of the converter. To make the scheme more efficient, the low-pass filter in Figure 9.9 is replaced by a *bank of filters* arranged in *parallel*, as illustrated in the efficient realisation. The sampling-rate conversion process is undertaken by the multiplexer at the output by selecting every MT/L samples. In this example, the efficient realisation is illustrated for a signal which is interpolated by L = 3 and decimated by M = 2 samples.

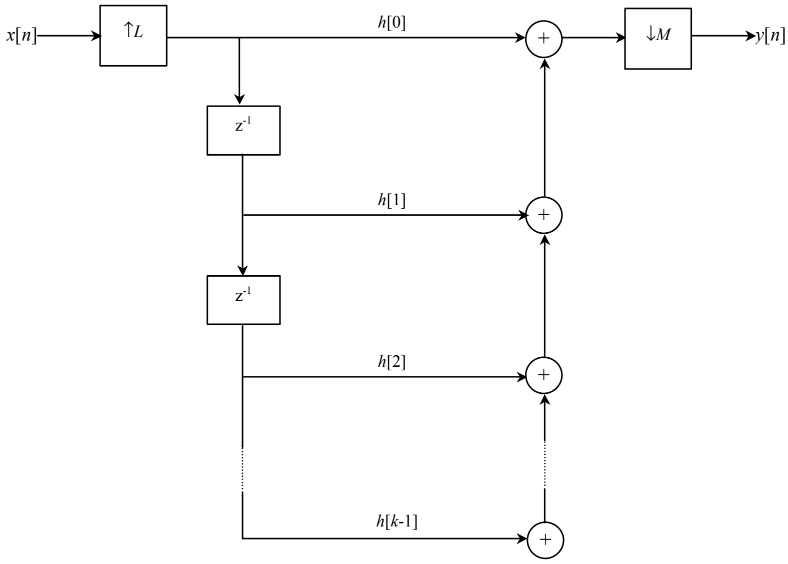

Figure 9.9: Realisation structure of sampling-rate conversion.

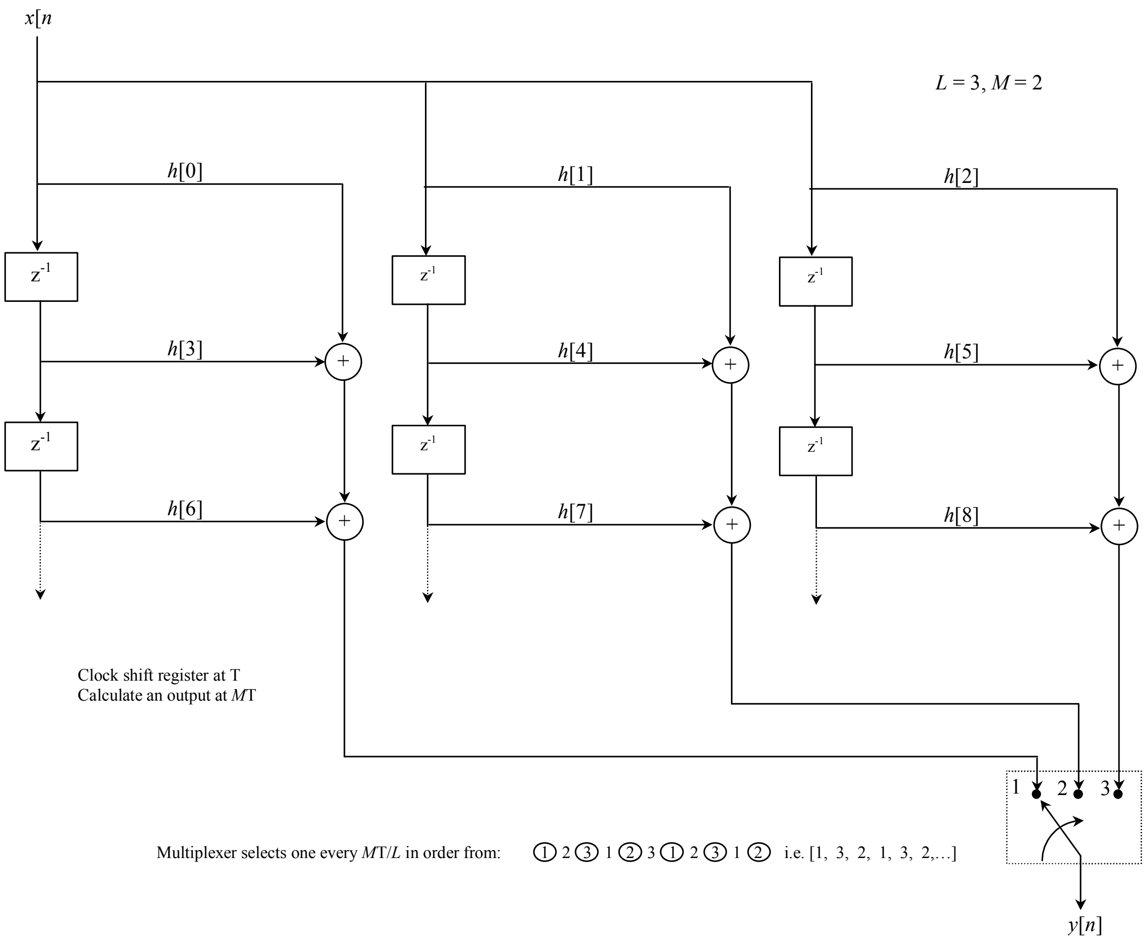

Figure 9.10: Efficient realisation structure for sampling-rate conversion.

9.8 Applications of Multirate DSP

Multirate systems are used in a CD player when the music signal is converted from digital into analogue (DAC). Digital data (16-bit words) are read from the disk at a sampling rate of 44.1 kHz. If this data were converted directly into an analogue signal, image frequency bands centred on multiples of the sampling-rate would occur, causing amplifier overload, and distortion in the music signal. To protect against this, a common technique called *oversampling* is often implemented nowadays in all CD players and in most digital processing systems of music signals. Figure 9.11 below illustrates a basic block diagram of a CD player and how oversampling is utilised. It is customary to oversample (or expand) the digital signal by a factor of x8, followed by an interpolation filter to remove the image frequencies. The sampling rate of the resulting signal is now increased up to 352.8 kHz. The digital signal is then converted into an analogue waveform by passing it through a 14-bit DAC. Then the output from this device is passed through an analogue low-pass filter before it is sent to the speakers.

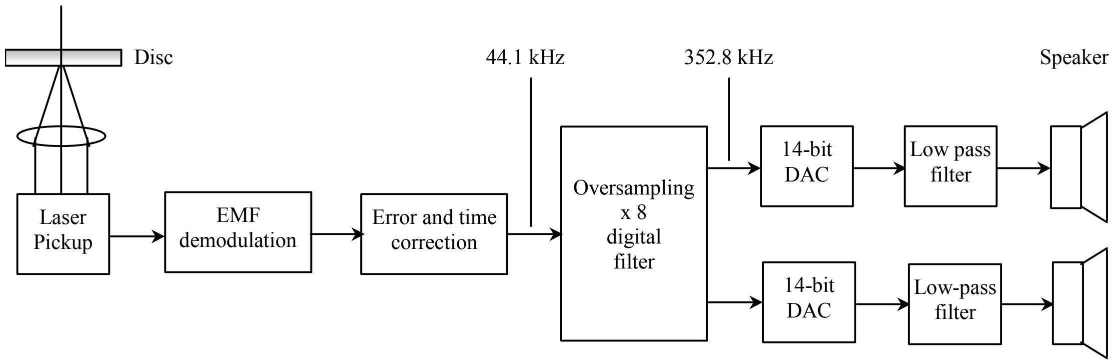

Figure 9.11: Digital to analogue conversion for a CD player using x8 oversampling.

Figure 9.12 illustrates the procedure of converting a digital waveform into an analogue signal in a CD player using x8 oversampling. As an example, Figure (a) illustrates a 20 kHz sinusoidal signal sampled at 44.1 kHz, denoted by $x[n]$. The six samples of the signal represent the waveform over two periods. If the signal $x[n]$ was converted directly into an analogue waveform, it would be very hard to exactly reconstruct the 20 kHz signal from this diagram. Now, Figure (b) shows $x[n]$ with an x8 interpolation, denoted by $y[n]$. Figure (c) shows the analogue signal $y(t)$, reconstructed from the digital signal $y[n]$ by passing it through a DAC. Finally, Figure (d) shows the waveform of $z(t)$, which is obtained by passing the signal y(t) through an analogue low-pass filter.

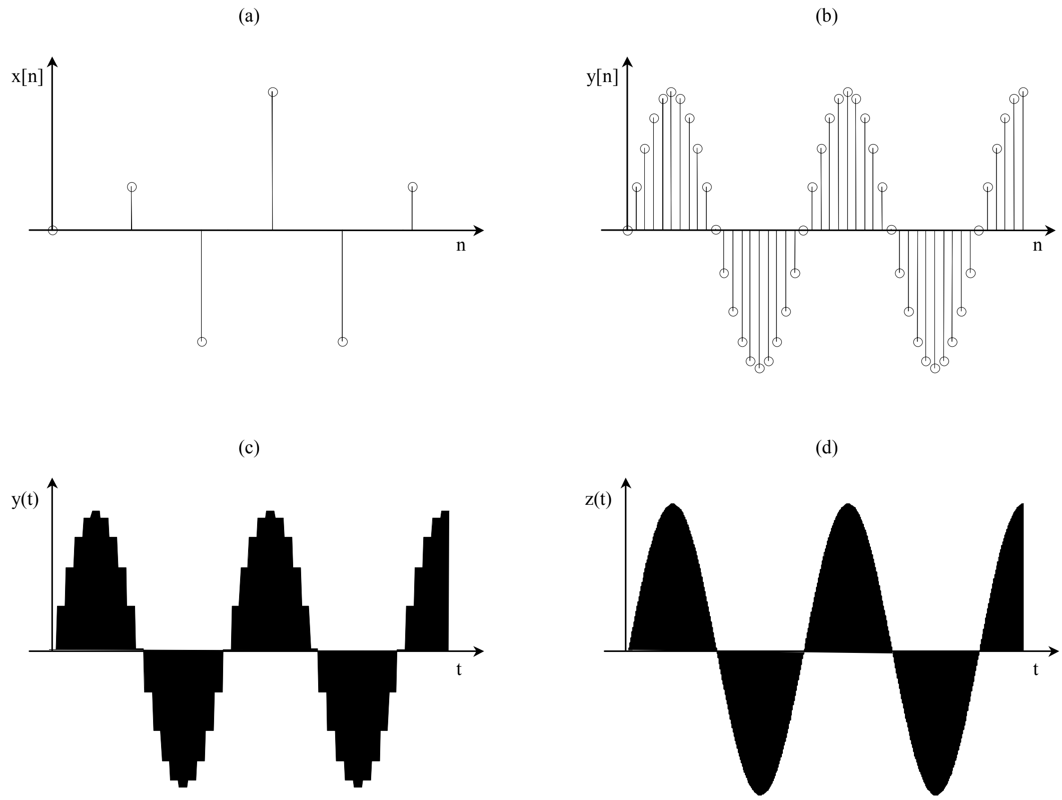

Figure 9.12: Illustration of oversampling in CD music signal reconstruction.

The effect of oversampling also has some other desirable features. Firstly, it causes the image frequencies to be much higher and therefore easier to filter out. The anti-alias filter specification can therefore be very much relaxed i.e. the cut-off frequency of the filter for the previous example increases from [44.1 / 2] = 22.05 kHz to [44.1x8 / 2] = 176.4 kHz after the interpolation.

One other attractive feature about oversampling is the effect of reducing the *noise power spectral density*, by spreading the noise power over a larger bandwidth. This is illustrated in Figure 9.13 and mathematical defined below by Equation 9.3.

$$Noise\ \ power\ \ spectral\ \ density = \frac{Total\ \ power}{Bandwidth} \tag{9.3}$$

For both sequences, the *total noise power* (shaded area in Figure 9.13) remains the same. However, as the bandwidth is increased by a factor of x8 because of the interpolation process, it causes the level of the noise power spectral density to decrease by a factor of x8, over the whole range of the bandwidth.

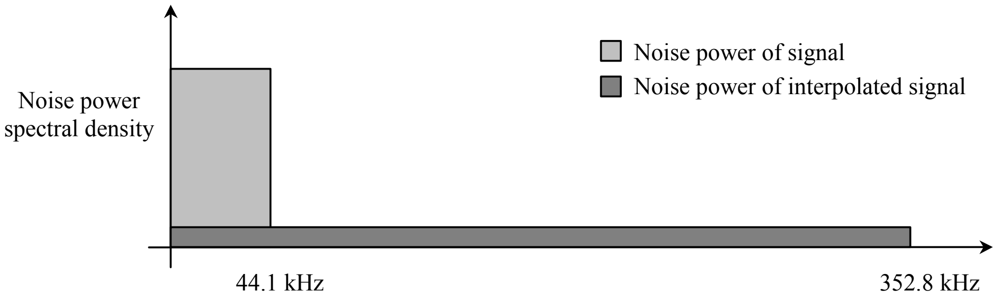

Figure 9.13: Illustration of noise power spectral density reduction due to oversampling.

As a consequence of the reduction in the noise power spectral density, it means that the level of tolerable noise can be increased by a factor of 8. In terms of the quantisation noise power, q^2, it means that it can now be 8 times greater (or the quantisation step size, q, can be increased by $\sqrt{8}$). This ultimately means that a reduction in the number of bits for the DAC is possible. In general, the reduction in the number of bits for the DAC process is given by Equation 9.4 below.

$$DAC\ \ bit\ \ reduction = \frac{1}{2}\log_2(oversample\ \ factor) \tag{9.4}$$

For the previous example, the DAC bit reduction owing to the x8 oversample factor is $1/2\log_2(8) = 1.5$ bits.

There are in fact more sophisticated oversampled ADCs and DACs that use various feedback paths within the system to move most of the quantisation noise into a high frequency out-of-band region. Substantially larger savings in the number of bits can then be made, even to one bit only, but these techniques are beyond the topic of this course.