Sampling of continuous-time signals

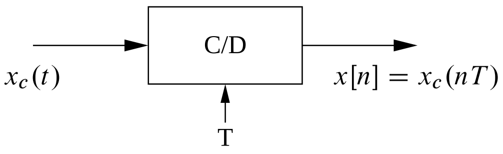
See Oppenheim and Schafer, Second Edition pages 140–239, or First Edition pages 80–148.

1 Periodic sampling

Discrete-time signal $x[n]$ often arises from periodic sampling of continuous-time signal $x_c(t)$:

$$x[n] = x_c(nT), \qquad -\infty < n < \infty.$$

This system is called an ideal continuous-to-discrete-time (C/D) converter or sampler,



and is described by the following:

- Sampling period: T seconds.
- Sampling frequency: $f_s = 1/T$ samples per second, or $\Omega_s = 2\pi/T$ radians per second.

In practice, sampling is usually approximately implemented using analog-to-digital (A/D) converter.

The sampling process is not generally invertible: one cannot always reconstruct $x_c(t)$ unambiguously from $x[n]$. However, ambiguity can be removed by *restricting input signals to sampler*.

2 Frequency-domain representation of sampling

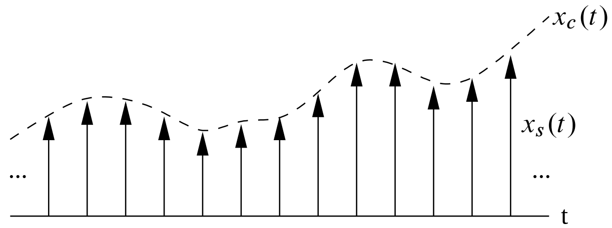
What is the frequency-domain relation between input and output of C/D converter?

Consider converting $x_c(t)$ to $x_s(t)$, by modulating it with the periodic impulse train

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT),$$

which has frequency representation

$$S(j\Omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s) :$$



Through the sifting property of the impulse function,

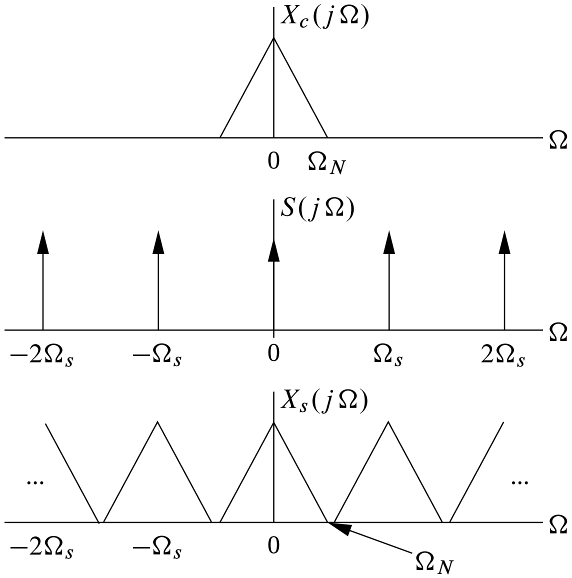
$$x_s(t) = x_c(t)s(t) = x_c(t) \sum_{n=-\infty}^{\infty} \delta(t - nT)$$

$$= \sum_{n=-\infty}^{\infty} x_c(nT)\delta(t - nT).$$

The Fourier transform $X_s(j\Omega)$ of $x_s(t) = x_c(t)s(t)$ is the continuous-time convolution of Fourier transforms $X_c(j\Omega)$ and $S(j\Omega)$, so

$$X_s(j\Omega) = \frac{1}{2\pi} X_c(j\Omega) * S(j\Omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_c(j(\Omega - k\Omega_s)).$$

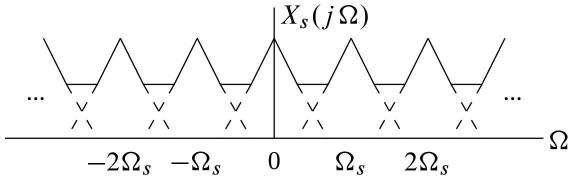
Therefore, the Fourier transform of $x_s(t)$ consists of copies of $X_c(j\Omega)$, shifted by integer multiples of sampling frequency Ω_s, and then superimposed:



If $x_c(t)$ is bandlimited, with highest nonzero frequency at Ω_N, then the replicas do not overlap when

$$\Omega_s > 2\Omega_N.$$

Then we can recover $x_c(t)$ from $x_s(t)$ using an ideal lowpass filter. *Otherwise*, $X_c(j\Omega)$ cannot be recovered using lowpass filtering — **aliasing** results:



The frequency Ω_N is referred to as the **Nyquist frequency**, and the frequency

$2\Omega_N$ that must be exceeded in the sampling is the **Nyquist rate**.

The objective now is to express the Fourier transform $X(e^{j\omega})$ of $x[n]$ in terms of $X_c(j\Omega)$ and $X_s(j\Omega)$. Taking the Fourier transform of the relationship

$$x_s(t) = \sum_{n=-\infty}^{\infty} x_c(nT)\delta(t - nT)$$

yields the following:

$$X_s(j\Omega) = \sum_{n=-\infty}^{\infty} x_c(nT)e^{-j\Omega Tn}.$$

Now, since $x[n] = x_c(nT)$ and

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n},$$

it follows that

$$X_s(j\Omega) = X(e^{j\omega})\big|_{\omega=\Omega T} = X(e^{j\Omega T}).$$

Consequently,

$$X(e^{j\Omega T}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c(j(\Omega - k\Omega_s)),$$

and

$$X(e^{j\omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right).$$

Thus $X(e^{j\omega})$ is just a frequency-scaled version of $X_s(j\Omega)$, with the scaling specified by $\omega = \Omega T$. Alternatively, the effect of sampling may be thought of as a *normalisation* of the frequency axis, so that the frequency $\Omega = \Omega_s$ of $X_s(j\Omega)$ is normalised to $\omega = 2\pi$ for $X(e^{j\omega})$.

3 Reconstruction of bandlimited signal from samples

If samples of a bandlimited continuous-time signal are taken frequently enough, then they are sufficient to represent the signal exactly. The continuous-time signal can then be recovered from the samples. This task is ideally performed by a discrete-to-continuous-time (D/C) converter. The form and behaviour of such a converter is discussed in this section.

Given sequence of samples $x[n]$, we can form impulse train

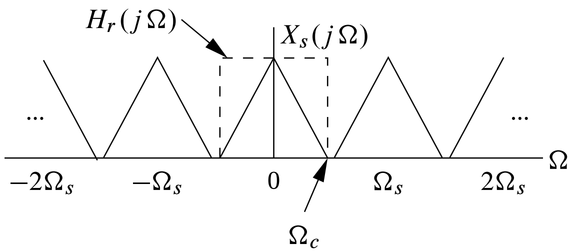
$$x_s(t) = \sum_{n=-\infty}^{\infty} x[n]\delta(t - nT).$$

The nth sample corresponds to the impulse at time $t = nT$.

If appropriate sampling conditions are met, namely the signal is bandlimited and the Fourier transform replicas do not overlap, then $x(t)$ can be reconstructed from $x_s(t)$ by ideal continuous-time lowpass filtering:

$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n]h_r(t - nT).$$

Here $h_r(t)$ is impulse response of an ideal LPF with cutoff frequency at Ω_c:



A convenient choice for the cutoff frequency is $\Omega_c = \Omega_s/2 = \pi/T$,

corresponding to the ideal reconstruction filter

$$H_r(j\Omega) = \begin{cases} T & |\Omega| \leq \pi/T \\ 0 & |\Omega| > \pi/T \end{cases}$$

and reconstructed signal

$$\begin{aligned} X_r(j\Omega) &= H_r(j\Omega)X(e^{j\Omega T}) \\ &= \begin{cases} TX(e^{j\Omega T}) & |\Omega| \leq \pi/T \\ 0 & |\Omega| > \pi/T. \end{cases} \end{aligned}$$

In the time domain the ideal reconstruction filter has impulse response

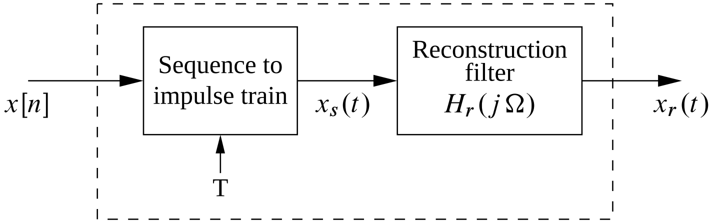
$$h_r(t) = \frac{\sin(\pi t/T)}{\pi t/T},$$

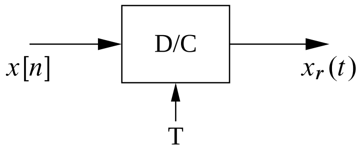
so the reconstructed signal is

$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n]\frac{\sin[\pi(t - nT)/T]}{\pi(t - nT)/T}.$$

From the previous frequency-domain argument, if $x[n] = x_c(nT)$ with $X_c(j\Omega) = 0$ for $|\Omega| \geq \pi/T$, then $x_r(t) = x_c(t)$. Note that the filter $h_r(t)$ is not realisable since it has infinite duration.

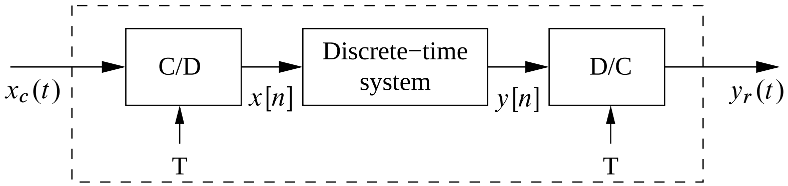
An ideal discrete-to-continuous (D/C) reconstruction system therefore has the form





4 Discrete-time processing of continuous-time signals

Discrete-time systems are often used to process continuous-time signals. This can be accomplished by a system of the form:



$H_{\text{eff}}(j\Omega) = H_c(j\Omega)$

For now it is assumed that the C/D and D/C converters have the same sampling rate.

The C/D converter produces the discrete-time signal

$$x[n] = x_c(nT),$$

with Fourier transform

$$X(e^{j\omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right).$$

The D/C converter creates a continuous-time output of the form

$$y_r(t) = \sum_{n=-\infty}^{\infty} y[n]\frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T}.$$

The continuous-time Fourier transform of $y_r(t)$, namely $Y_r(j\Omega)$, and the discrete-time Fourier transform of $y[n]$, namely $Y(e^{j\Omega})$, are related by

$$\begin{aligned} Y_r(j\Omega) &= H_r(j\Omega)Y(e^{j\Omega T}) \\ &= \begin{cases} TY(e^{j\Omega T}) & |\Omega| < \pi/T \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

If the discrete-time system is LTI, then

$$Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega}),$$

where $H(e^{j\omega})$ is the frequency response of the system. Therefore

$$\begin{aligned} Y_r(j\Omega) &= H_r(j\Omega)H(e^{j\Omega T})X(e^{j\Omega T}) \\ &= H_r(j\Omega)H(e^{j\Omega T})\frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\Omega - \frac{2\pi k}{T}\right)\right). \end{aligned}$$

If $X_c(j\Omega) = 0$ for $|\Omega| \geq \pi/T$, then the ideal LPF $H_r(j\Omega)$ selects only the term for $k = 0$ in the sum, and scales the result:

$$Y_r(j\Omega) = \begin{cases} H(e^{j\Omega T})X_c(j\Omega) & |\Omega| < \pi/T \\ 0 & |\Omega| \geq \pi/T. \end{cases}$$

Thus if $X_c(j\Omega)$ is bandlimited and sampled above the Nyquist rate, then the output is related to the input by

$$Y_r(j\Omega) = H_{\text{eff}}(j\Omega)X_c(j\Omega),$$

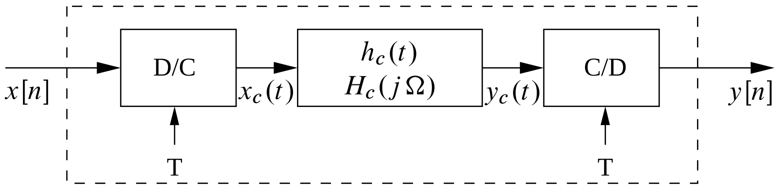
where

$$H_{\text{eff}}(j\Omega) = \begin{cases} H(e^{j\Omega T}) & |\Omega| < \pi/T \\ 0 & |\Omega| \geq \pi/T \end{cases}$$

is the effective frequency response of the system.

5 Continuous-time processing of discrete-time signals

It is conceptually useful to consider continuous-time processing of discrete-time signals. A system to perform this task is:



$h[n]$, $H(e^{j\omega})$

Since the D/C converter includes an ideal LPF, $X_c(j\Omega)$ and therefore also $Y_c(j\Omega)$ will be zero for $|\Omega| \geq \pi/T$. Thus the C/D converter samples $y_c(t)$ without aliasing and we have

$$x_c(t) = \sum_{n=-\infty}^{\infty} x[n] \frac{\sin[\pi(t - nT)/T]}{\pi(t - nT)/T}$$

and

$$y_c(t) = \sum_{n=-\infty}^{\infty} y[n] \frac{\sin[\pi(t - nT)/T]}{\pi(t - nT)/T},$$

where $x[n] = x_c(nT)$ and $y[n] = y_c(nT)$. In the frequency domain,

$$X_c(j\Omega) = TX(e^{j\Omega T}), \qquad |\Omega| < \pi/T,$$
$$Y_c(j\Omega) = H_c(j\Omega)X_c(j\Omega), \qquad |\Omega| < \pi/T,$$
$$Y(e^{j\omega}) = \frac{1}{T} Y_c\left(j\frac{\omega}{T}\right), \qquad |\omega| < \pi.$$

The overall system therefore behaves like a discrete-time system with frequency response

$$H(e^{j\omega}) = H_c\left(j\frac{\omega}{T}\right) \qquad |\omega| < \pi.$$

Equivalently, the overall frequency response of the system will be equal to a given $H(e^{j\omega})$ if the frequency of the continuous-time system is

$$H_c(j\Omega) = H(e^{j\Omega T}), \qquad |\Omega| < \pi/T.$$

Since $X_c(j\Omega) = 0$ for $|\Omega| \geq \pi/T$, $H_c(j\Omega)$ may be chosen arbitrarily above π/T.

6 Changing sampling rate using discrete-time processing

Given the sequence

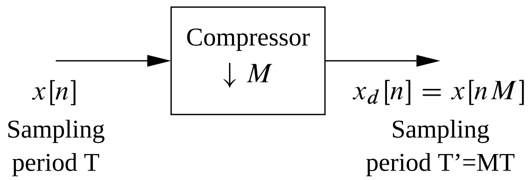
$$x[n] = x_c(nT)$$

obtained by sampling (with period T) the signal $x_c(t)$, we often want to change the sampling rate (to period T'):

$$x'[n] = x_c(nT').$$

One approach is to reconstruct $x_c(t)$ from $x[n]$, and then resample with new period T'. However, we want to do this using only discrete-time operations.

6.1 Sampling rate reduction by integer factor



The sampling rate **compressor** implements the following function:

$$x_d[n] = x[nM] = x_c(nMT).$$

Here $x_d[n]$ is exactly the sequence that would be obtained by sampling $x_c(t)$ with period $T' = MT$.

If $X_c(j\Omega) = 0$ for $|\Omega| \geq \Omega_N$, then $x_d[n]$ is an exact (unaliased) representation of $x_c(t)$ if $\pi/(MT) \geq \Omega_N$.

In the frequency domain we have

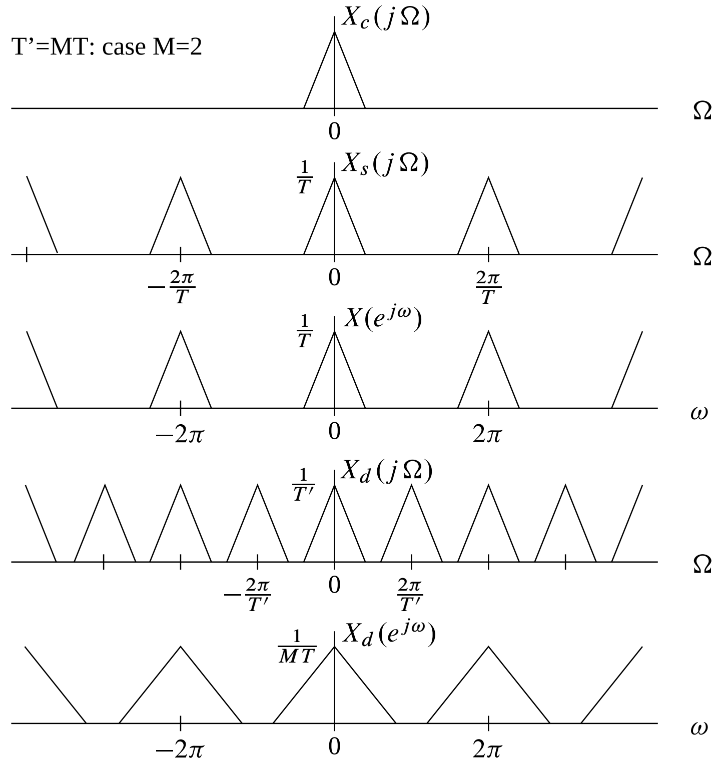
$$X(e^{j\omega}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right)$$

and

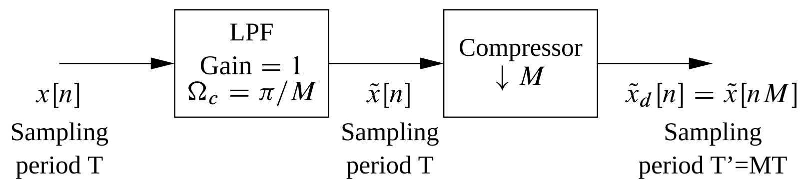
$$X_d(e^{j\omega}) = \frac{1}{T'} \sum_{r=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T'} - \frac{2\pi r}{T'}\right)\right).$$

Since $T' = MT$, and noting that with $r = i + kM$ we can write the summation over r as a summation over $-\infty < k < \infty$ and $0 \leq i \leq M-1$, we obtain

$$\begin{aligned} X_d(e^{j\omega}) &= \frac{1}{M} \sum_{i=0}^{M-1} \left[\frac{1}{T} \sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{MT} - \frac{2\pi k}{T} - \frac{2\pi i}{MT}\right)\right)\right]. \\ &= \frac{1}{M} \sum_{i=0}^{M-1} X(e^{j(\omega/M - 2\pi i/M)}). \end{aligned}$$



Applying a compressor to a signal can result in aliasing. This can be avoided (at the cost of some information) by prefiltering with a lowpass filter, and then compressing the sampling rate:



This is referred to as **downsampling (or decimation) by a factor** M.

6.2 Increasing sampling rate by integer factor

With underlying continuous-time signal $x_c(t)$, we want to obtain samples

$$x_i[n] = x_c(nT')$$

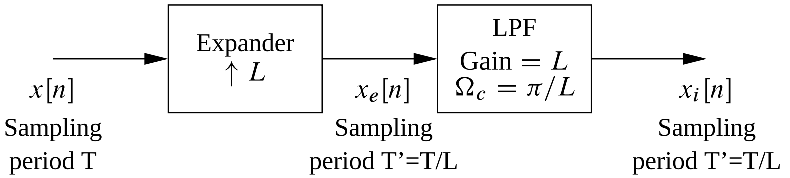
from

$$x[n] = x_c(nT),$$

where $T' = T/L$. Therefore

$$x_i[n] = x[n/L] = x_c(nT/L), \qquad n = 0, \pm L, \pm 2L, \ldots.$$

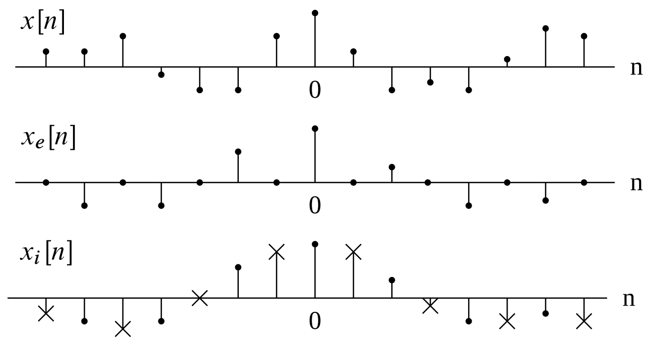
This is referred to as **upsampling (or interpolating) by a factor** L, and is performed by **expanding** the sampling rate, and then lowpass filtering:



The expanded signal is

$$x_e[n] = \begin{cases} x[n/L], & n = 0, \pm L, \pm 2L, \ldots, \\ 0, & \text{otherwise}, \end{cases}$$
$$= \sum_{k=-\infty}^{\infty} x[k]\delta[n - kL].$$

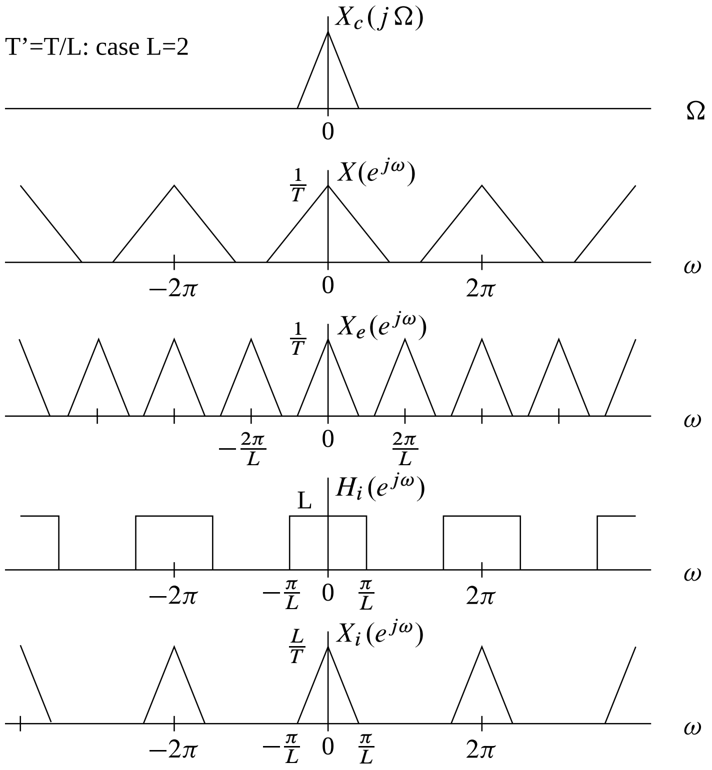
An example of upsampling in the discrete-time domain is shown below:



The Fourier transform of the expanded signal is

$$X_e(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \left(\sum_{k=-\infty}^{\infty} x[k]\delta[n - kL] \right) e^{-j\omega n}$$
$$= \sum_{k=-\infty}^{\infty} x[k] e^{-j\omega L k} = X(e^{j\omega L}).$$

Final upsampling is obtained by lowpass filtering the expanded signal.



We can obtain an interpolation formula for $x_i[n]$ in terms of $x[n]$: since the LPF has impulse response

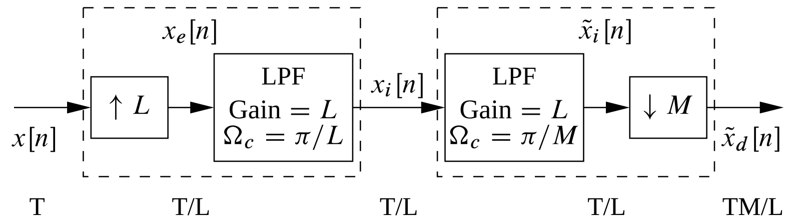
$$h_i[n] = \frac{\sin(\pi n/L)}{\pi n/L},$$

we have

$$x_i[n] = \sum_{k=-\infty}^{\infty} x[k] \frac{\sin[\pi(n-kL)/L]}{\pi(n-kL)/L}.$$

6.3 Changing the sampling rate by a noninteger factor

By cascading upsampling (by factor L) and downsampling (by factor M), the sampling rate can be changed by a noninteger factor.



This forms the basis of **multirate signal processing**, where highly efficient structures are developed for implementing complicated signal processing operations. The discrete wavelet transform (DWT) can be developed in this framework.