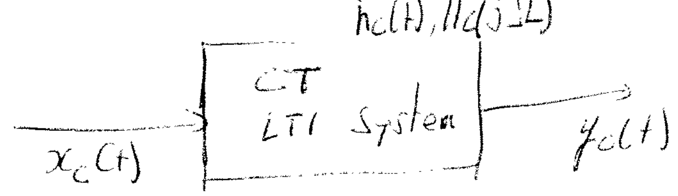
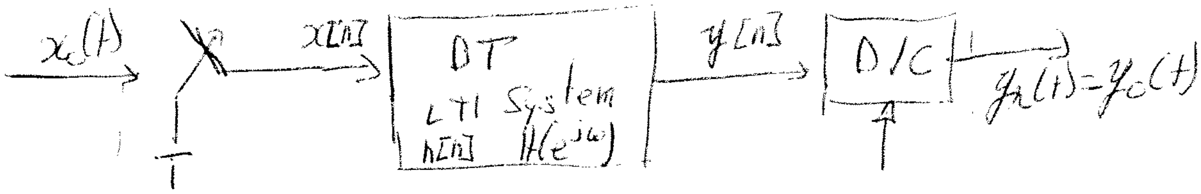
II Sampling and Discrete Time Processing

Most of the time, Discrete Time (DT) Processing is used to process a Continuous Time (CT) signal. This can be illustrated as

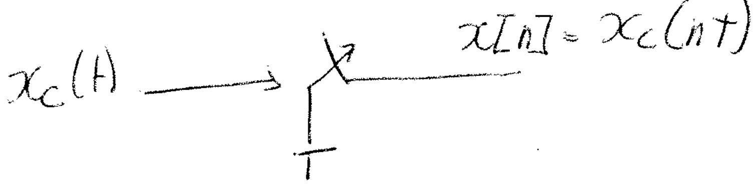


$\Updownarrow$



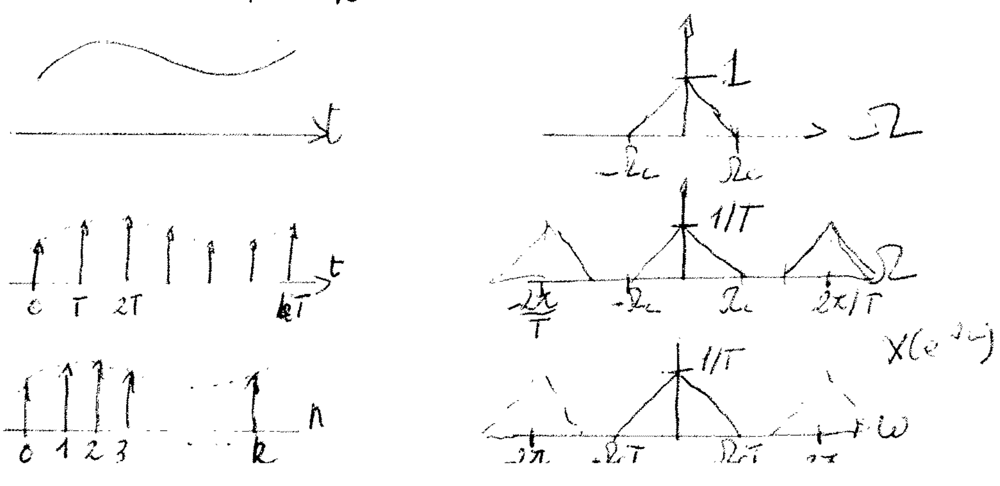
$$H_{eff}(j\Omega) = H_c(j\Omega)$$

II.1 Ideal C/D Converter



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

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



In the Frequency domain, C/D converter is

① Repetition of the spectrum at $\frac{2\pi}{T}$

then ② scaling of the amplitude by $1/T$

then ③ normalization of frequency by $1/T$

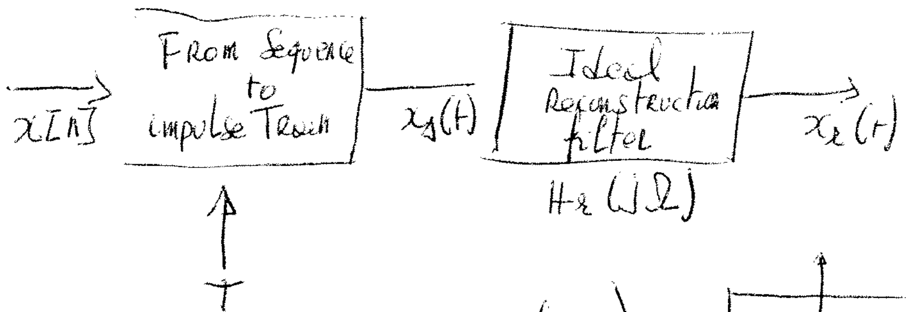
Ω is the frequency of the CT signal

ω is " DT "

$$\Omega = \frac{\omega}{T}$$

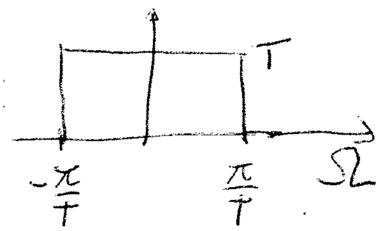
⚠ if $x_c(t)$ not limited to $-\frac{\Omega_s}{2} < \Omega < \frac{\Omega_s}{2}$ ⇒ Aliasing

II.2 Ideal D/C converter

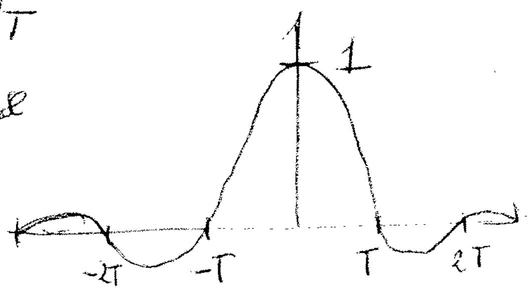


$$x_r(t) = \sum_n x[n] \, \mathrm{sinc}\left(\frac{t - nT}{T}\right)$$

$H_r(j\Omega)$



$$X_r(j\Omega) = \begin{cases} T\,X(e^{j\omega}) & |\Omega| \le \pi/T \\ 0 & \text{otherwise} \end{cases}$$



$$\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$

In the Frequency Domain D/C conversion is

(1) band limiting to the base period

$$-\frac{\Omega_s}{2} \le \omega \le \frac{\Omega_s}{2} = \frac{\pi}{T}$$

(2) normalization of the frequency by T

(3) scaling of the amplitude by T

II.3 DT Processing of CT Signals

in figure of page 11, if $x_c(t)$ is bandlimited

$$\left(X_c(j\Omega) = 0 \text{ for } |\Omega| \le \frac{\pi}{T} \right)$$

the overall system is LTI equivalent to a CT LTI sys

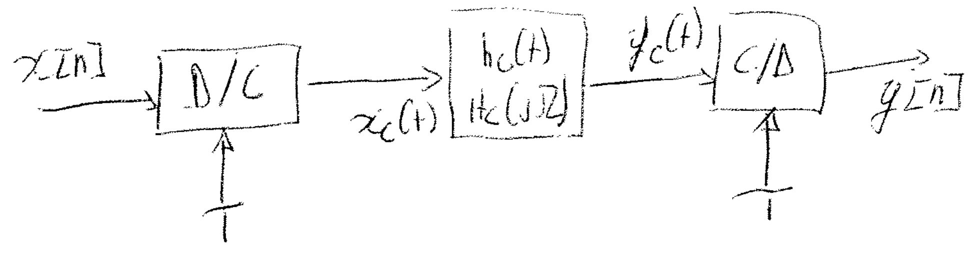
$$H_{eff}(j\Omega) = H(j\Omega) \quad \text{for } |\Omega| \le \frac{\pi}{T}$$

Ex : $H_c(j\omega)$: Low pass filter, cutoff freq $= \pi/4$

$\frac{1}{T}$: 20 kHz

$H_{eff}(j\Omega)$ lowpass filter, cutoff freq = ?

II.4 CT Processing of DT signals Fractional Delay



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

$$y_c(t) = \quad " \quad y[n] \quad "$$

In the frequency domain

$$X_c(j\Omega) = T X(e^{j\Omega T}) \qquad |\Omega| < \pi/T$$

$$Y_c(j\Omega) = H_c(j\Omega) X_c(j\Omega) \qquad |\Omega| < \pi/T$$

$$Y(e^{j\omega}) = \frac{1}{T} Y_c\left(j\frac{\omega}{T}\right) \qquad |\omega| < \pi$$

$\Rightarrow$ We have an overall DT system with

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

Example $\rightarrow$ Non Integer Delay (O. S. B p 164-165)

MA with Noninteger Delay

Fall 15

or equivalently, the overall frequency response of the system in Figure 4.16 will be equal to a given $H(e^{j\omega})$ if the frequency response of the continuous-time system is

$$H_c(j\Omega) = H(e^{j\Omega T}), \qquad |\Omega| < \pi/T. \tag{4.61}$$

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

With this representation of a discrete-time system, we can focus on the equivalent effect of the continuous-time system on the bandlimited continuous-time signal $x_c(t)$. This is illustrated in Examples 4.9 and 4.10.

Example 4.9 Noninteger Delay

Let us consider a discrete-time system with frequency response

$$H(e^{j\omega}) = e^{-j\omega\Delta}, \qquad |\omega| < \pi. \tag{4.62}$$

When Δ is an integer, this system has a straightforward interpretation as a delay of Δ, i.e.,

$$y[n] = x[n - \Delta]. \tag{4.63}$$

When Δ is not an integer, Eq. (4.63) has no formal meaning, because we cannot shift the sequence $x[n]$ by anything but an integer. However, with the use of the system of Figure 4.16, a useful time-domain interpretation can be applied to the system specified by Eq. (4.62). Let $H_c(j\Omega)$ in Figure 4.16 be chosen to be

$$H_c(j\Omega) = H(e^{j\Omega T}) = e^{-j\Omega\Delta T}. \tag{4.64}$$

Then, from Eq. (4.61), the overall discrete-time system in Figure 4.16 will have the frequency response given by Eq. (4.62), whether or not Δ is an integer. To interpret the system of Eq. (4.62), we note that Eq. (4.64) represents a time delay of ΔT seconds. Therefore,

$$y_c(t) = x_c(t - \Delta T). \tag{4.65}$$

Furthermore, $x_c(t)$ is the bandlimited interpolation of $x[n]$, and $y[n]$ is obtained by sampling $y_c(t)$. For example, if $\Delta = \frac{1}{2}$, $y[n]$ would be the values of the bandlimited

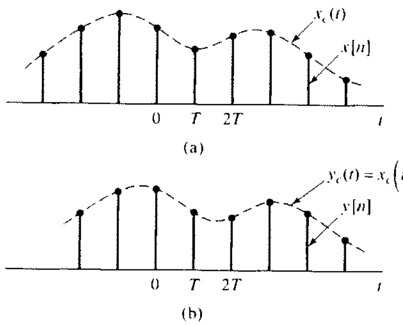


Figure 4.17 Continuous-time processing of the discrete-time sequence in part (a) can produce a new sequence with a "half-sample" delay, as in part (b).

interpolation halfway between the input sequence values. This is illustrated in Figure 4.17. We can also obtain a direct convolution representation for the system defined by Eq. (4.62). From Eqs. (4.65) and (4.57), we obtain

$$\begin{aligned} y[n] &= y_c(nT) = x_c(nT - \Delta T) \\ &= \sum_{k=-\infty}^{\infty} x[k] \left.\frac{\sin[\pi(t - \Delta T - kT)/T]}{\pi(t - \Delta T - kT)/T}\right|_{t=nT} \\ &= \sum_{k=-\infty}^{\infty} x[k] \frac{\sin \pi(n - k - \Delta)}{\pi(n - k - \Delta)}, \end{aligned} \tag{4.66}$$

which is, by definition, the convolution of $x[n]$ with

$$h[n] = \frac{\sin \pi(n - \Delta)}{\pi(n - \Delta)}, \qquad -\infty < n < \infty.$$

When Δ is not an integer, $h[n]$ has infinite extent. However, when $\Delta = n_0$ is an integer, it is easily shown that $h[n] = \delta[n - n_0]$, which is the impulse response of the ideal integer delay system.

The noninteger delay represented by Eq. (4.66) has considerable practical significance, since such a factor often arises in the frequency-domain representation of systems. When this kind of term is found in the frequency response of a causal discrete-time system, it can be interpreted in the light of this example. This interpretation is illustrated in Example 4.10.

Example 4.10 Moving-Average System with Noninteger Delay

In Example 2.20, we considered the general moving-average system and obtained its frequency response. For the case of the causal $(M + 1)$-point moving-average system, $M_1 = 0$ and $M_2 = M$, and the frequency response is

$$H(e^{j\omega}) = \frac{1}{(M+1)} \frac{\sin[\omega(M+1)/2]}{\sin(\omega/2)} e^{-j\omega M/2}, \qquad |\omega| < \pi. \tag{4.67}$$

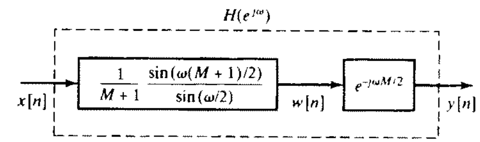
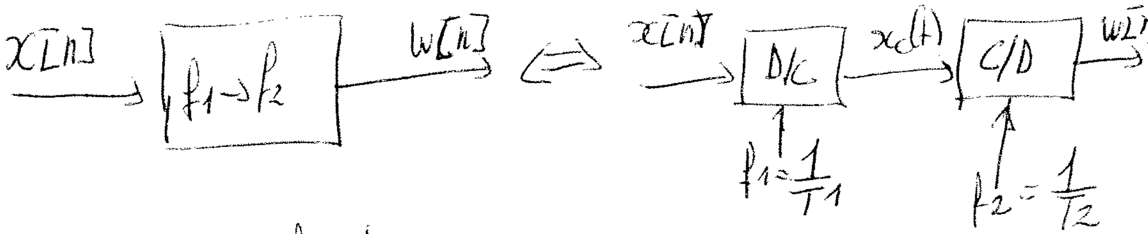


Figure 4.18 The moving-average system represented as a cascade of two systems.

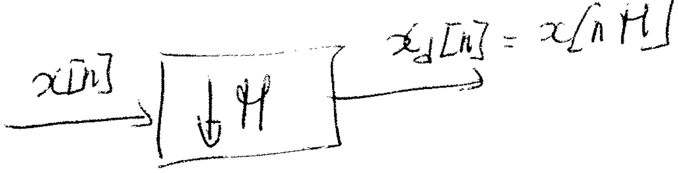
This representation of the frequency response suggests the interpretation of the $(M + 1)$-point moving-average system as the cascade of two systems, as indicated in Figure 4.18. The first system imposes a frequency-domain amplitude weighting. The second system represents the linear-phase term in Eq. (4.67). If M is an even integer (meaning the moving average of an odd number of samples), then the linear-phase

II.5 Sampling Rate Compression by an Integer factor

In general, sampling rate Conversion can be depicted as:



For the Integer factor



$x[n] = x_c(nT)$

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

with

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

$$X_d(e^{j\omega}) = \frac{1}{MT}\sum_k X_c\left(j\left(\frac{\omega}{MT} - \frac{2\pi k}{MT}\right)\right)$$

$$k = i + bM \Rightarrow X_d(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1}\left[\frac{1}{T}\sum_k \left(j\left(\frac{\omega}{MT} - \frac{2\pi k}{T} - \frac{2\pi i}{MT}\right)\right)\right]$$

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

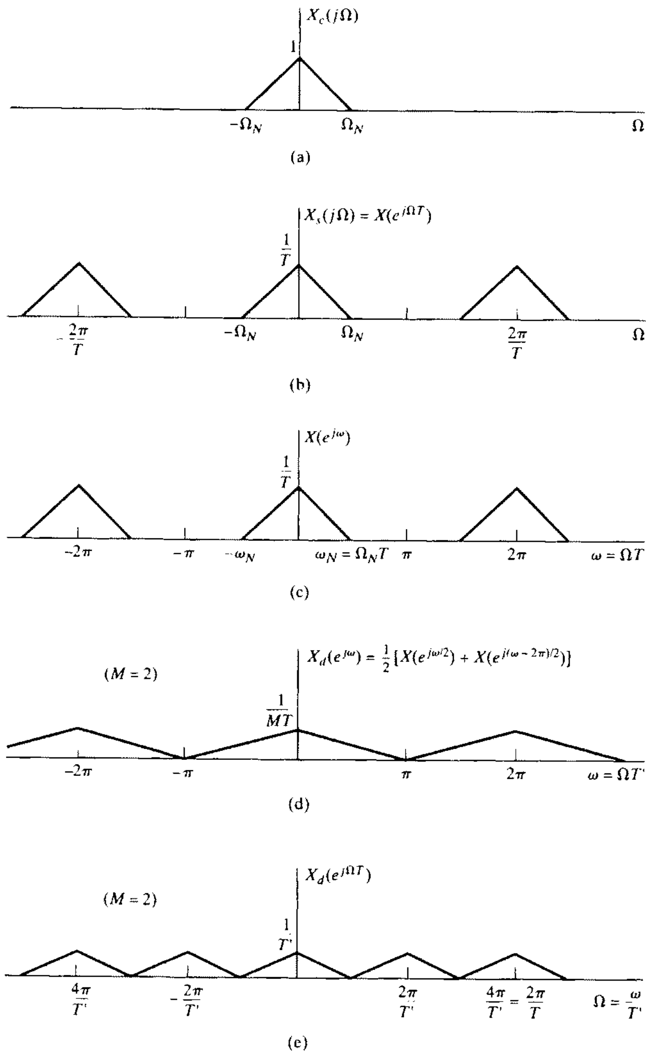


Figure 4.21 Frequency-domain illustration of downsampling.

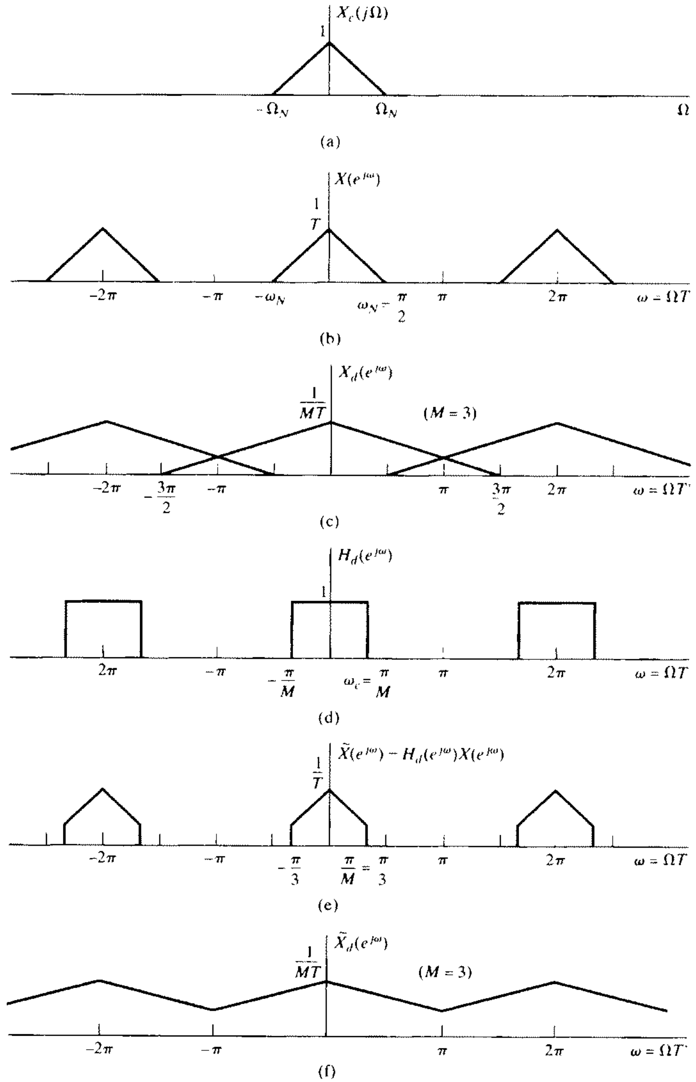
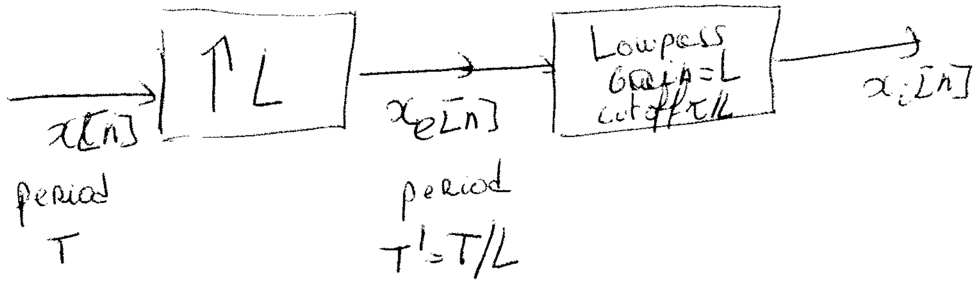


Figure 4.22 (a)–(c) Downsampling with aliasing. (d)–(f) Downsampling with prefiltering to avoid aliasing.

II. 6 Sampling Rate Expansion by an Integer Factor



$$X_e(e^{j\omega}) = \sum_n x_e[n]\, e^{-j\omega n}$$
$$= \sum_k x_e[k]\, e^{-j\omega k L} = X(e^{j\omega L})$$

+ Interpolation: OSB p 174-175

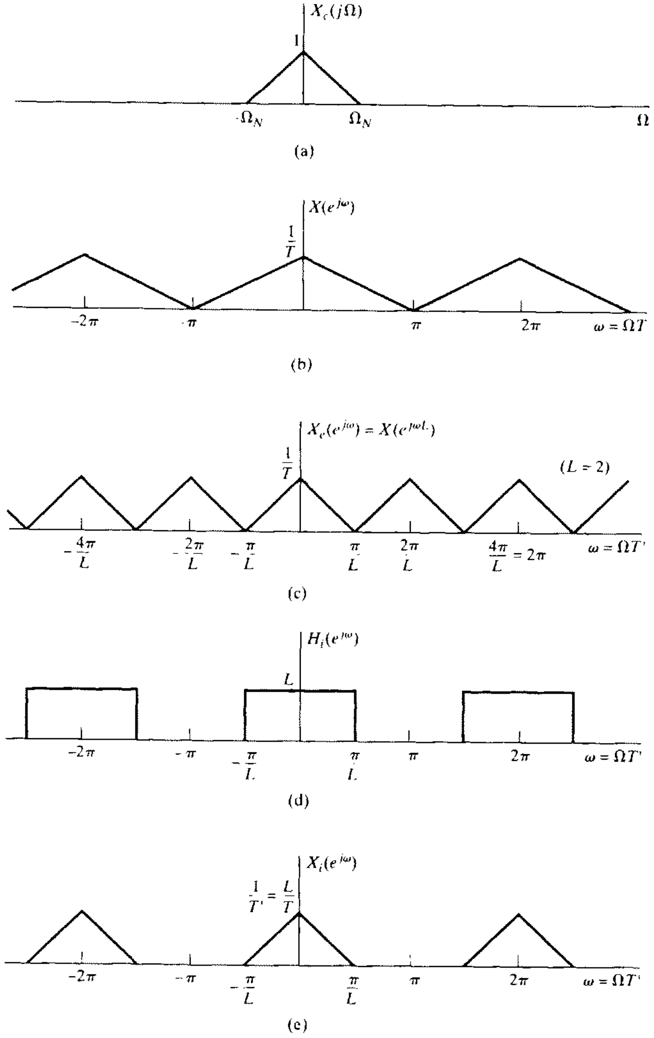


Figure 4.25 Frequency-domain illustration of interpolation.

Filt 19

Rabiner, 1973, and Oetken et al., 1975.) In some cases, very simple interpolation procedures are adequate. Since linear interpolation is often used (even though it is generally not very accurate), it is worthwhile to examine linear interpolation within the general framework that we have just developed.

Linear interpolation can be accomplished by the system of Figure 4.24 if the filter has impulse response

$$h_{\text{lin}}[n] = \begin{cases} 1 - |n|/L, & |n| \le L, \\ 0, & \text{otherwise}, \end{cases} \tag{4.92}$$

as shown in Figure 4.26 for $L = 5$. With this filter, the interpolated output will be

$$x_{\text{lin}}[n] = \sum_{k=-\infty}^{\infty} x_e[k] h_{\text{lin}}[n-k] = \sum_{k=-\infty}^{\infty} x[k] h_{\text{lin}}[n-kL]. \tag{4.93}$$

Figure 4.27(a) depicts $x_e[n]$ and $x_{\text{lin}}[n]$ for the case $L = 5$. From this figure, we see that

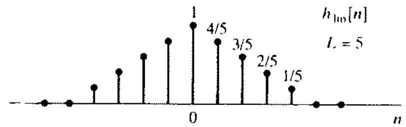


Figure 4.26 Impulse response for linear interpolation.

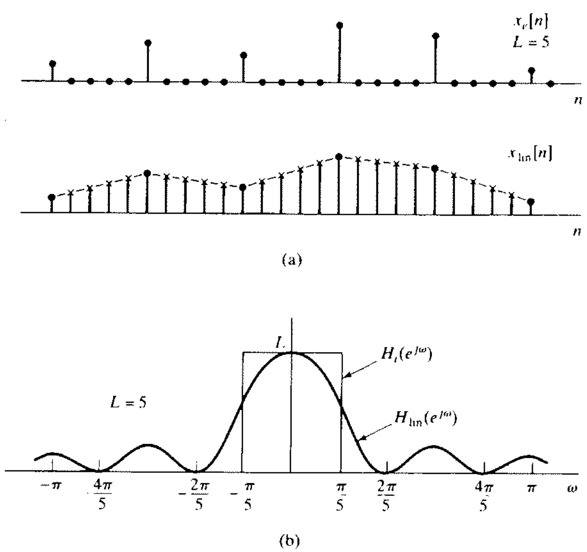


Figure 4.27 (a) Illustration of linear interpolation by filtering. (b) Frequency response of linear interpolator compared with ideal lowpass interpolation filter.

Fült 20

$x_{\text{lin}}[n]$ is identical to the sequence obtained by linear interpolation between the samples. Note that

$$\begin{aligned} h_{\text{lin}}[0] &= 1, \\ h_{\text{lin}}[n] &= 0, \qquad n = \pm L, \pm 2L, \ldots \end{aligned} \tag{4.94}$$

so that

$$x_{\text{lin}}[n] = x[n/L] \quad \text{at} \quad n = 0, \pm L, \pm 2L, \ldots \tag{4.95}$$

The amount of distortion in the intervening samples can be gauged by comparing the frequency response of the linear interpolator with that of the ideal lowpass interpolator for a factor-of-L interpolation. It can be shown (see Problem 4.50) that

$$H_{\text{lin}}(e^{j\omega}) = \frac{1}{L}\left[\frac{\sin(\omega L/2)}{\sin(\omega/2)}\right]^2. \tag{4.96}$$

This function is plotted in Figure 4.27(b) for $L = 5$, together with the ideal lowpass interpolation filter. From the figure we see that if the original signal is sampled at the Nyquist rate, linear interpolation will not be very good, since the output of the filter will contain considerable energy in the band $\pi/L < |\omega| \le \pi$. However, if the original sampling rate is much higher than the Nyquist rate, then the linear interpolator will be more successful in removing the frequency-scaled images of $X_c(j\Omega)$ at multiples of $2\pi/L$. This is because $H_{\text{lin}}(e^{j\omega})$ is small at these normalized frequencies and at higher sampling rates the shifted copies of $X_c(j\Omega)$ are more localized at these frequencies. This is intuitively reasonable, since, if the original sampling rate greatly exceeds the Nyquist rate, the signal will not vary significantly between samples, and thus, linear interpolation should be more accurate for oversampled signals.

4.6.3 Changing the Sampling Rate by a Noninteger Factor

We have shown how to increase or decrease the sampling rate of a sequence by an integer factor. By combining decimation and interpolation, it is possible to change the sampling rate by a noninteger factor. Specifically, consider Figure 4.28(a), which shows an interpolator that decreases the sampling period from T to T/L, followed by a decimator that increases the sampling period by M, producing an output sequence $\tilde{x}_d[n]$ that has an effective sampling period of $T' = TM/L$. By choosing L and M appropriately, we can approach arbitrarily close to any desired ratio of sampling periods. For example, if $L = 100$ and $M = 101$, then $T' = 1.01T$.

If $M > L$, there is a net increase in the sampling period (a decrease in the sampling rate), and if $M < L$, the opposite is true. Since the interpolation and decimation filters in Figure 4.28(a) are in cascade, they can be combined as shown in Figure 4.28(b) into one lowpass filter with gain L and cutoff equal to the minimum of π/L and π/M. If $M > L$, then π/M is the dominant cutoff frequency, and there is a net reduction in sampling rate. As pointed out in Section 4.6.1, if $x[n]$ was obtained by sampling at the Nyquist rate, the sequence $\tilde{x}_d[n]$ will be a lowpass-filtered version of the original underlying bandlimited signal if we are to avoid aliasing. On the other hand, if $M < L$, then π/L is the dominant cutoff frequency, and there will be no need to further limit the bandwidth of the signal below the original Nyquist frequency.

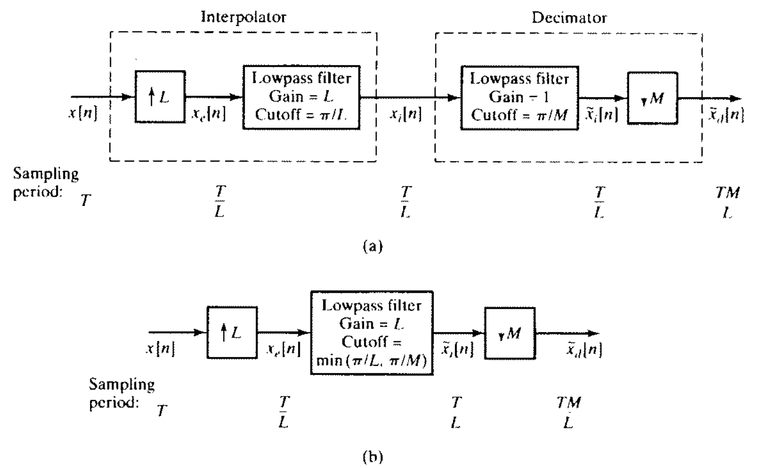


Figure 4.28 (a) System for changing the sampling rate by a noninteger factor. (b) Simplified system in which the decimation and interpolation filters are combined.

Example 4.11 Sampling Rate Conversion by a Noninteger Rational Factor

Figure 4.29 illustrates sampling rate conversion by a rational factor. Suppose that a bandlimited signal with $X_c(j\Omega)$ as given in Figure 4.29(a) is sampled at the Nyquist rate; i.e., $2\pi/T = 2\Omega_N$. The resulting discrete-time Fourier transform

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

is plotted in Figure 4.29(b). If we wish to change the sampling period to $T' = (3/2)T$, we must first interpolate by a factor $L = 2$ and then decimate by a factor of $M = 3$. Since this implies a net decrease in sampling rate, and the original signal was sampled at the Nyquist rate, we must incorporate additional lowpass filtering in order to avoid aliasing.

Figure 4.29(c) shows the discrete-time Fourier transform of the output of the $L = 2$ upsampler. If we were interested only in interpolating by a factor of 2, we could choose the lowpass filter to have a cutoff frequency of $\omega_c = \pi/2$ and a gain of $L = 2$. However, since the output of the filter will be decimated by $M = 3$, we must use a cutoff frequency of $\omega_c = \pi/3$, but the gain of the filter should still be 2 as in Figure 4.29(d). The Fourier transform $\tilde{X}_i(e^{j\omega})$ of the output of the lowpass filter is shown in Figure 4.29(e). The shaded regions indicate the part of the signal spectrum that is removed due to the lower cutoff frequency for the interpolation filter. Finally, Figure 4.29(f) shows the discrete-time Fourier transform of the output of the downsampler by $M = 3$. Note that the shaded regions show the aliasing that would have occurred if the cutoff frequency of the interpolation lowpass filter had been $\pi/2$ instead of $\pi/3$.

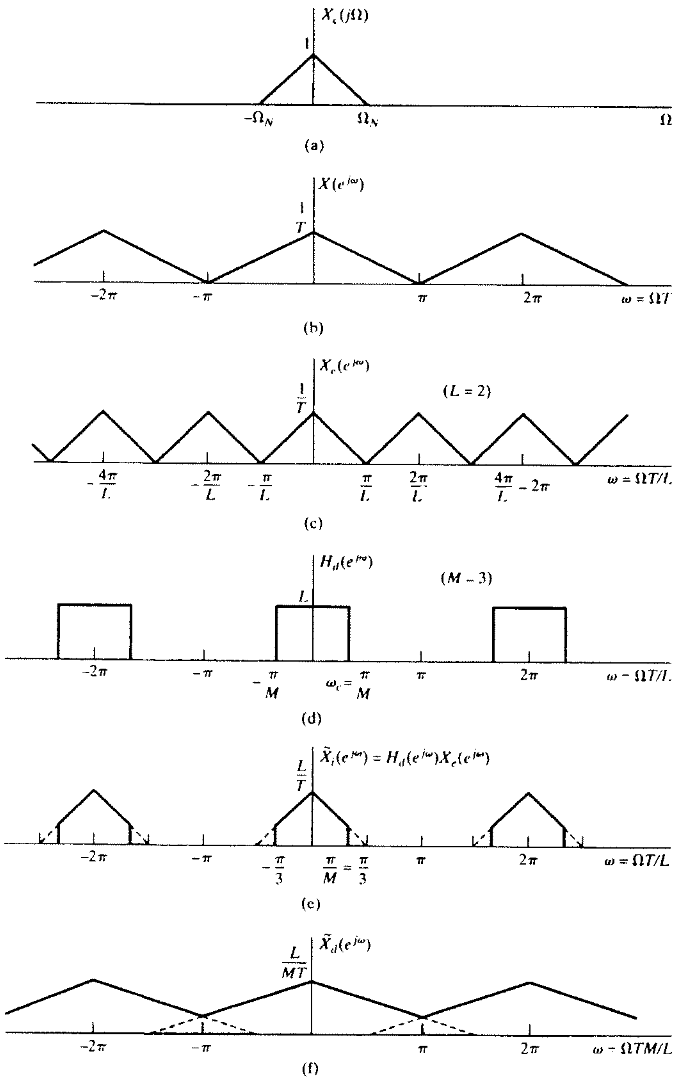


Figure 4.29 Illustration of changing the sampling rate by a noninteger factor.

Fi.lt 21

4.7 MULTIRATE SIGNAL PROCESSING

As we have seen, it is possible to change the sampling rate of a discrete-time signal by a combination of interpolation and decimation. For example, if we want a new sampling period of $T' = 1.01T$, we can first interpolate by $L = 100$ using a lowpass filter that cuts off at $\omega_c = \pi/101$ and then decimate by $M = 101$. These large intermediate changes in sampling rate would require large amounts of computation for each output sample if we implement the filtering in a straightforward manner at the high intermediate sampling rate that is required. Fortunately, it is possible to greatly reduce the amount of computation required by taking advantage of some basic techniques in the area of *multirate signal processing*. Multirate techniques refer in general to utilizing upsampling, downsampling, compressors, and expanders in a variety of ways to increase the efficiency of signal-processing systems. Besides their use in sampling rate conversion, they are exceedingly useful in A/D and D/A systems that exploit oversampling and noise shaping. Another important class of signal-processing algorithms that relies increasingly on multirate techniques is filter banks for the analysis and/or processing of signals.

Because of their widespread applicability, there is a large body of results on multirate signal processing. In this section, we will focus on two basic results and show how a combination of these results can greatly improve the efficiency of sampling rate conversion. The first result is concerned with the interchange of filtering and downsampling or upsampling operations. The second is the polyphase decomposition.

4.7.1 Interchange of Filtering and Downsampling/Upsampling

We will first derive two identities that aid in manipulating and understanding the operation of multirate systems. It is straightforward to show that the two systems in Figure 4.30 are equivalent. To see the equivalence, note that in Figure 4.30(b),

$$X_b(e^{j\omega}) = H(e^{j\omega M})X(e^{j\omega}), \tag{4.97}$$

and from Eq. (4.78),

$$Y(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X_b(e^{j(\omega/M - 2\pi i/M)}). \tag{4.98}$$

Substituting Eq. (4.97) into Eq. (4.98) gives

$$Y(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X(e^{j(\omega/M - 2\pi i/M)})H(e^{j(\omega - 2\pi i)}). \tag{4.99}$$

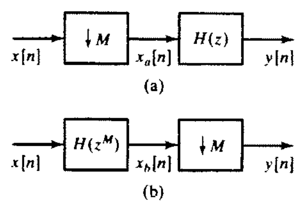
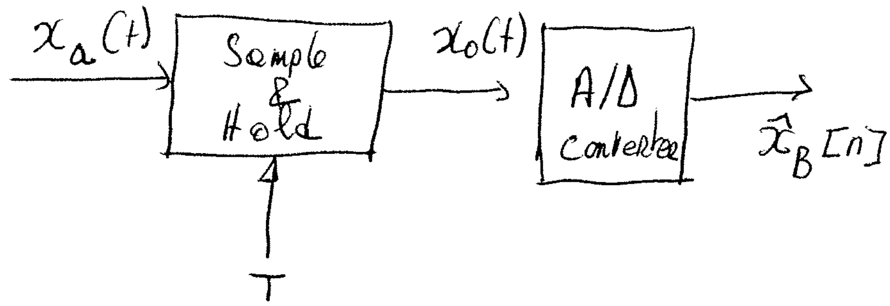


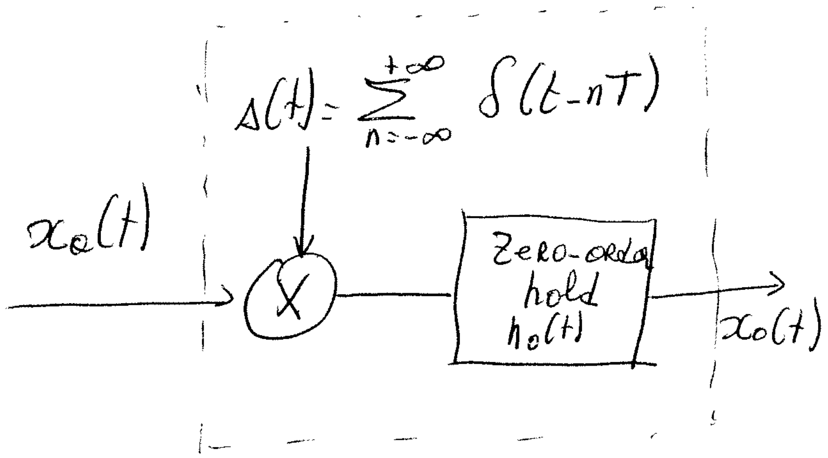
Figure 4.30 Two equivalent systems based on downsampling identities.

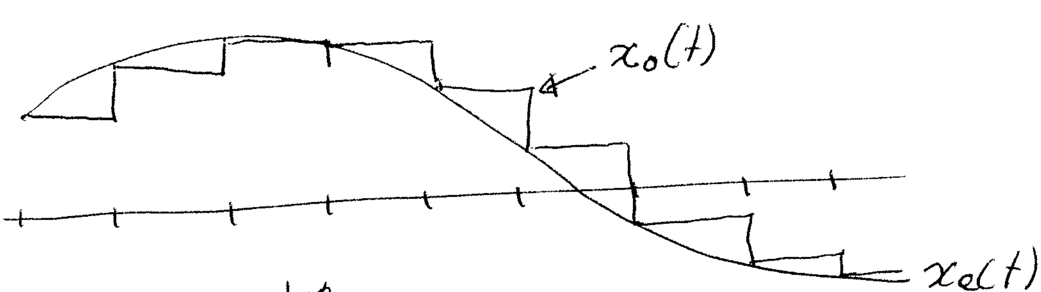
III Quantization and Oversampled Noise Shaping

Analog-to-digital conversion "in the real world"



Ideal sample & Hold





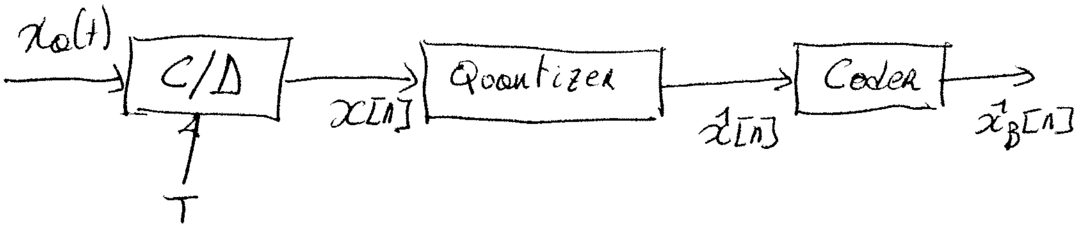
$$x_0(t) = \sum_{n=-\infty}^{+\infty} x[n]\, h_0(t-nT)$$

$$= h_0(t) \otimes \sum_{n=-\infty}^{+\infty} x_a(nT)\, \delta(t-nT)$$

$$h_0(t) = \begin{cases} 1 & 0 < t < T \\ 0 & \text{ailleurs} \end{cases}$$

Filt 23

An analysable representation of the A/D



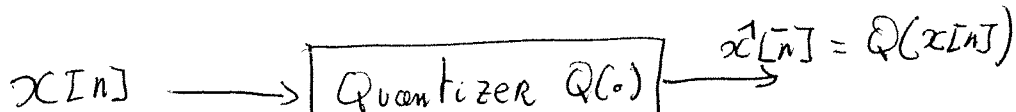
For example, with uniform quantization, if $-X_M < x[n] < X_M$ and quantization on B bits

$\Rightarrow$ 2^B levels

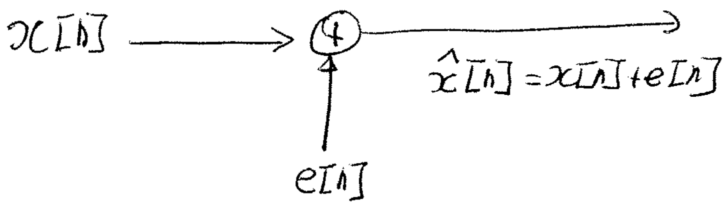
$\rightarrow$ Spacing between adjacent levels : $\Delta = X_M 2^{-B}$

Problem of quantization

+ Error $\rightarrow$ quantization noise
+ Non linear operation



Additive noise model



Statistics of quantization errors

$$-\Delta/2 < e[n] < \Delta/2$$

$\Rightarrow$ for many signals $e[n]$ uniformly distributed

Model of $e[n]$

+ $e[n]$ is a sample sequence of a stationary random process
+ $e[n]$ is uncorrelated with $x[n]$
+ $e[n]$ is a white-noise process
+ $e[n]$ is uniformly distributed over $[-\Delta/2, \Delta/2]$

$$\Rightarrow E\{e[n]\} = 0$$

$$\sigma_e^2 = \frac{\Delta^2}{12}$$

$$R_e[l] = \frac{\Delta^2}{12}\delta[m]$$

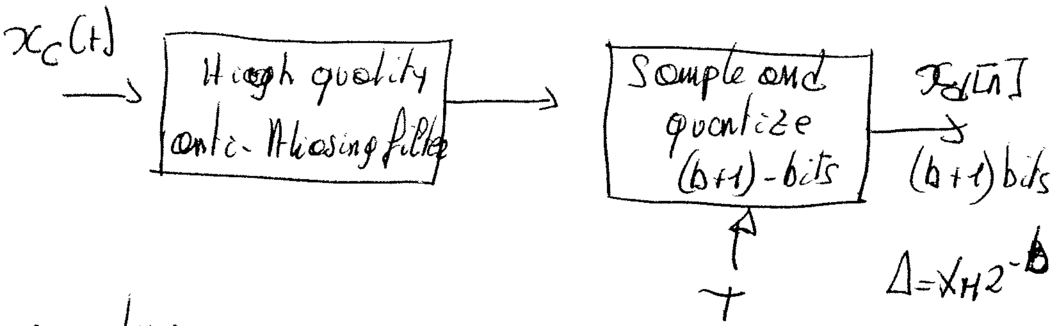
$$\Rightarrow S_e(f) = \frac{\Delta^2}{12}$$

$$\Rightarrow \text{total power} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{\Delta^2}{12}\,d\omega = R_e[0] = \frac{\Delta^2}{12}$$

CD player example

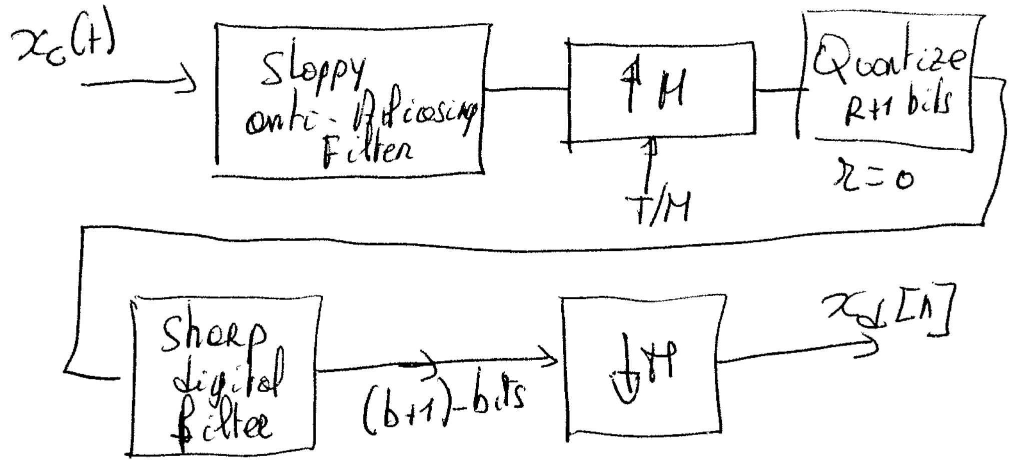
Use of oversampling to mitigate quantization effects

the classical way

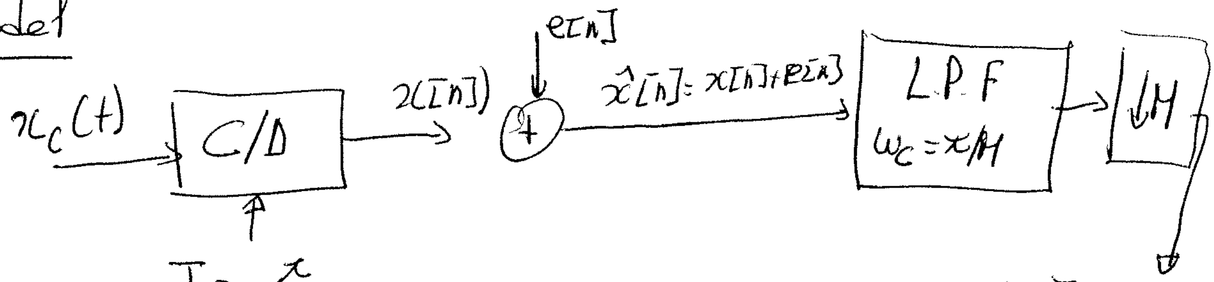


$\Delta = X_M 2^{-b}$

Using oversampling



Noise Model



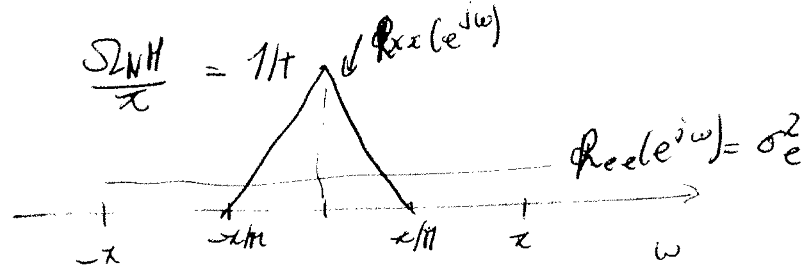
$T = \frac{\pi}{\Omega_N M}$

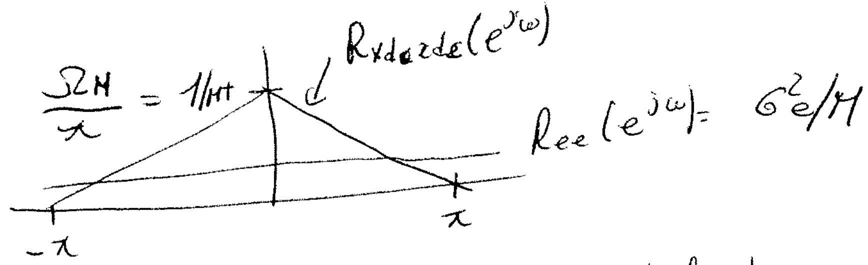
$x_d[n] = x_{d_c}[n] + x_{d_e}[n]$

Noise power $\int_{-\frac{1}{2}}^{1/2} S_e(f)\,df = \frac{\Delta^2}{12}$

after the LPF $\xrightarrow{\text{Power}} R(0) = \frac{\Delta^2}{12 M}$

Filt 26





when M is doubled, noise power is halved

since $\Delta = X_M 2^{-B}$

$$\frac{\Delta^2}{12} = \frac{X_M^2 4^{-B}}{12}$$

doubling M $\Rightarrow \frac{1}{2}\frac{\Delta^2}{12} = \frac{X_M^2 4^{-(B+1/2)}}{12}$

$\rightarrow$ equivalent to add half a bit of precision

SNR calculation

Noise power: $\sigma_e^2 = \frac{2^{-2B} X_M^2}{12}$

Signal power: σ_x^2

$$\Rightarrow SNR_{dB} = 10 \log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right) = 10 \log_{10} \frac{12 \cdot 2^{2B} \sigma_x^2}{X_m^2}$$

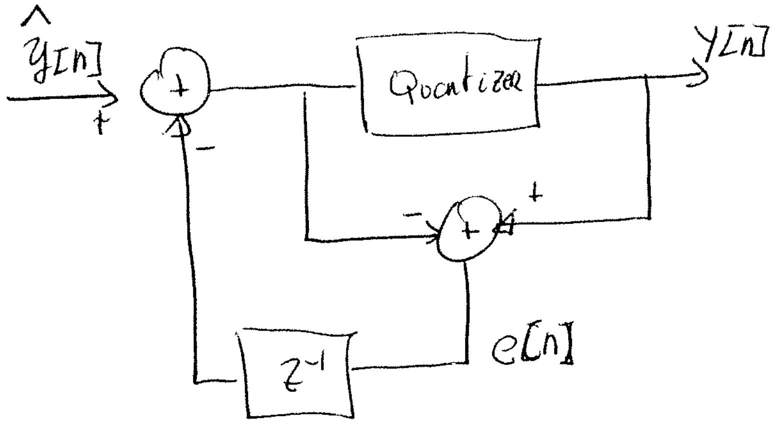
$$= 6.02 B + 10.8 - 20 \log_{10}\left(\frac{X_m}{\sigma_x}\right)$$

$\Rightarrow$ adding 1 Bit enhances the SNR by 6dB

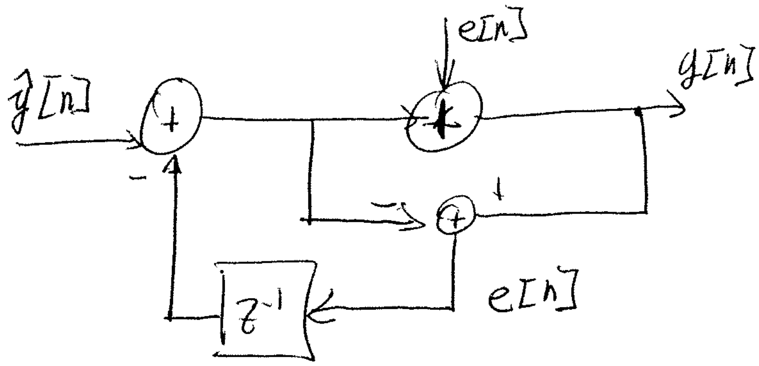
Felt 27

Noise Shaping

D/A converter



Noise Model



Question: how slow does $e[n]$ need to be changing so that it is significantly reduced?

Suppose, without loss of generality

$$\hat{y}[n] = 0$$

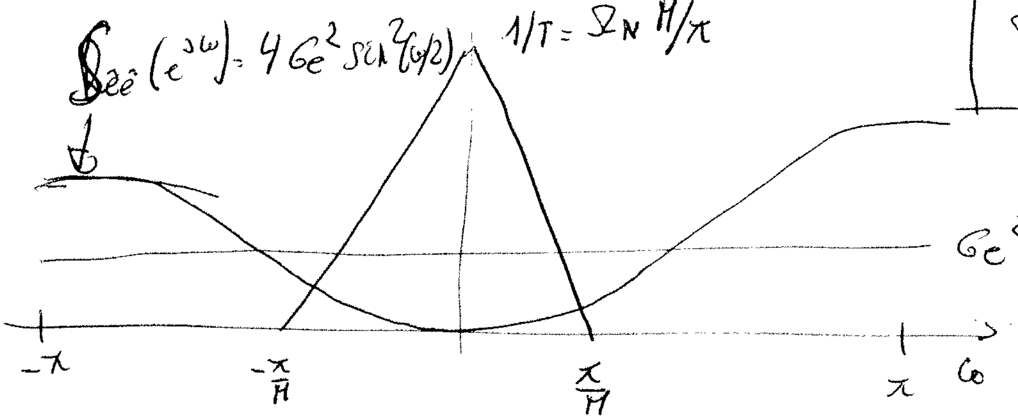
$$\Rightarrow y[n] = e[n] - e[n-1]$$

$$\Rightarrow \frac{Y(e^{j\omega})}{E(e^{j\omega})} = 1 - e^{-j\omega}$$

$$\left|\frac{Y(e^{j\omega})}{E(e^{j\omega})}\right|^2 = (1-e^{-j\omega})(1-e^{j\omega}) = 2 - 2\cos\omega = 4\sin^2\omega/2$$

$$\Rightarrow S_{\hat{e}\hat{e}}(f) = \frac{\Delta^2}{12} \cdot 4 \cdot \sin^2(\omega/2)$$

Filt 28



after downsampling