Sampling of Continuous-Time Signals

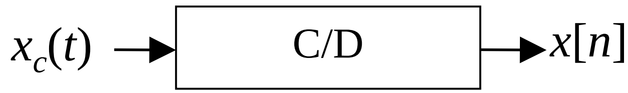
Carrson C. Fung
Dept. of Electronics Engineering
National Chiao Tung University

Outline

- Continuous-to-discrete (C/D)
- Discrete-to-continuous (D/C) – perfect reconstruction
- Frequency-domain analysis of sampling process
- Sampling rate conversion

Periodic Sampling

Ideal continuous-to-discrete-time (C/D) converter



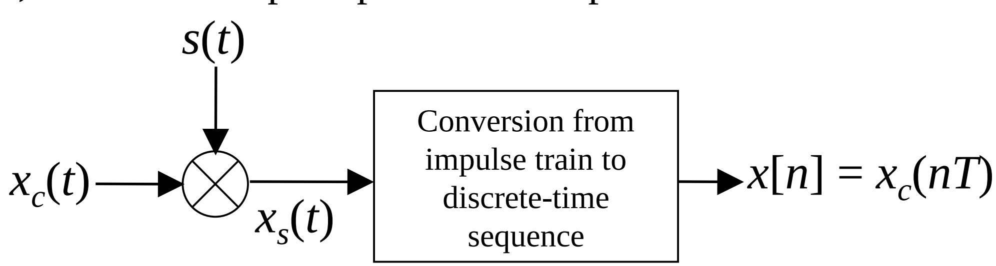
Continuous-time signal: $x_c(t)$

Discrete-time signal: $x[n] = x_c(nT)$, $-\infty < n < \infty$, T: sampling period

In theory, we break the C/D operation in two steps:

1. Ideal sampling using "analog delta function (Dirac delta function)"
   - Can be modeled by equations
2. Conversion from impulse train to discrete-time sequence
   - Only a concept, no mathematical model

In reality, the electronic analog-to-digital (A/D) circuits can approximate the ideal C/D operation. This circuitry is one piece; it cannot be split up into two steps



Ideal Sampling – Time Domain

$$x_c(t) \longrightarrow \boxed{\text{Sampling}} \longrightarrow x_s(t)$$

Ideal sampling signal: impulse train (continuous-time signal)

$$s(t) = \sum_n \delta(t - nT), \quad T\text{: sampling period}$$

Continuous - time signal: $x_c(t)$

Sampled (continuous - time) signal: $x_s(t)$

$$\begin{aligned} x_s(t) &= x_c(t)\,s(t) = x_c(t)\sum_n \delta(t - nT) \\ &= \sum_n x_c(t)\,\delta(t - nT) \\ &= \sum_n x_c(nT)\,\delta(t - nT) \end{aligned}$$

Ideal Sampling – Frequency Domain

Note that $$s(t) \leftrightarrow S(j\Omega) = \frac{2\pi}{T}\sum_k \delta(\Omega - k\Omega_s), \quad \text{where } \Omega_s = \frac{2\pi}{T}$$

Step 1: Ideal sampling (all in analog domain)

$$X_s(j\Omega) = \frac{1}{2\pi} X_c(j\Omega) * S(j\Omega) = \frac{1}{T} X_c(j\Omega) * \sum_k \delta(\Omega - k\Omega_s)$$

$$= \frac{1}{T}\sum_k X_c(j\Omega) * \delta(\Omega - k\Omega_s)$$

$$\overset{\overbrace{}^{\text{sifting property}}}{=} \frac{1}{T}\sum_k X_c\left(j\left(\Omega - k\Omega_s\right)\right) \qquad (*)$$

Remark: Ω: analog frequency (radians/sec)

ω: discrete (normalized) frequency (radians/sample)

$$\Omega = \frac{\omega}{T}; \quad -\pi < \omega \le \pi, \quad -\frac{\pi}{T} < \Omega < \frac{\pi}{T}$$

Step 1 (contd)

The sampled signal spectrum is the sum of shifted copies of the original.

Remark: In analog domain $x(t)y(t)$

$\Leftrightarrow \frac{1}{2\pi} X(j\Omega) * Y(j\Omega)$, $X_s(j\Omega)$ can also be expressed as:

$$X_s(j\Omega) = \int_t x_s(t) e^{-j\Omega t} dt = \int_t \sum_n x_c(nT)\delta(t - nT) e^{-j\Omega t} dt$$

$$= \sum_n x_c(nT) \int_t \delta(t - nT) e^{-j\Omega t} dt$$

$$= \sum_n x_c(nT) e^{-j\Omega nT} \qquad (**)$$

We also express $X(e^{j\omega})$ as:

$$X(e^{j\omega}) = \sum_n x[n] e^{-j\omega n} = \sum_n x_c(nT) e^{-jwn} \qquad (***)$$

Step 2: Analog to Sequence (Analog to Discrete-Time)

Comparing (**) and (***), we see that $X\left(e^{j\omega}\right)$ is equivalent to $X_s\left(e^{j\omega}\right)$ if $\omega = \Omega T$, so that

$$X_s\left(j\Omega\right) = X\left(e^{j\omega}\right)\Big|_{\omega=\Omega T} = X\left(e^{j\Omega T}\right)$$

Finally, from (*) we have

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

No mathematical model. The spectrum of $x_s(t)$, $X_s(j\Omega)$ has the same spectrum as $x[n]$ and $X\left(e^{j\Omega T}\right)$, respectively.

$X\left(e^{j\omega}\right)$ is a frequency-scaled version of $X(j\Omega)$

$$X\left(j\Omega\right) = X\left(e^{j\omega}\right)\Big|_{\omega=\Omega T}.$$

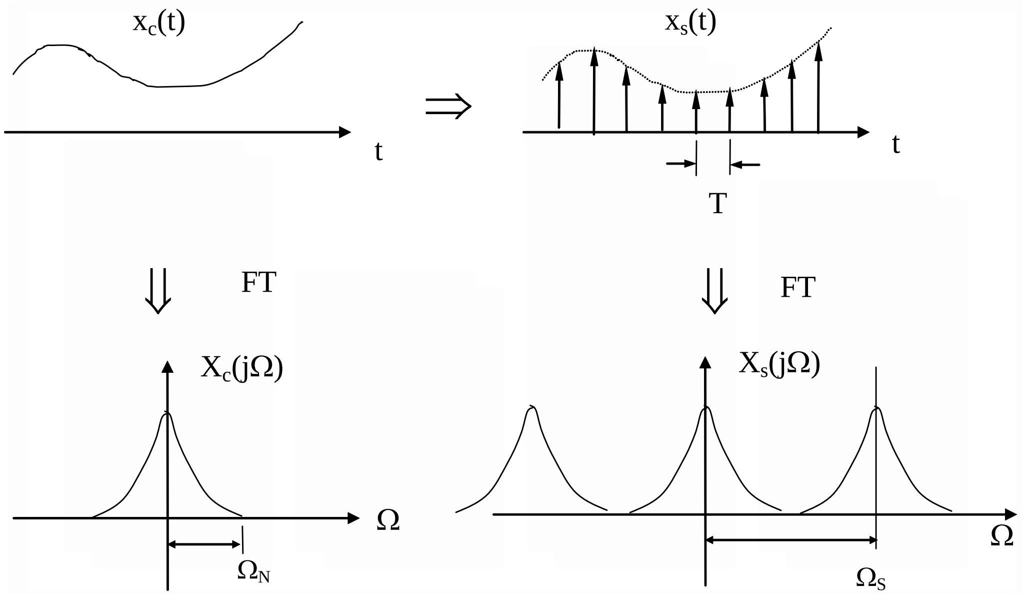
Since $X\left(e^{j\Omega T}\right) = \frac{1}{T}\sum_k X_c\left(j\left(\Omega - k\Omega_s\right)\right)$, thus

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

Remark: In time domain, $x_s(t)$ and $x[n]$ are two very different signals but have similar spectra in frequency domain.

Aliasing

- Two cases
  - No aliasing: $\Omega_s > 2\,\Omega_N$
  - Aliasing: $\Omega_s < 2\,\Omega_N$, where Ω_N is the highest nonzero frequency component of $X_c(j\Omega)$.
    - After sampling, the replicas of overlap (in frequency domain). That is, the higher frequency components of overlap with the lower frequency components of .

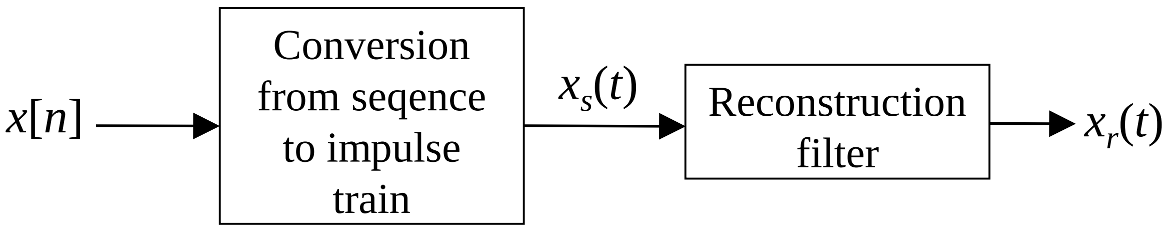


Nyquist Sampling Theorem

- Let $x(t)$ be a *bandlimited* signal with $X_c(j\Omega)=0$ for $|\Omega| \geq \Omega_N$. (i.e., no components at frequencies greater than Ω_N) Then $x_c(t)$ is uniquely determined by its samples $x[n]=x_c(nT)$, for $n=0, \pm 1, \pm 2, \ldots$, if $\Omega_s = 2\pi/T \geq 2\Omega_N$. (Nyquist, Shannon)
  - **Nyquist frequency** = Ω_N, the bandwidth of signal
  - **Nyquist rate** = $2\Omega_N$, the minimum sampling rate without distortion. (In some books, Nyquist frequency = Nyquist rate.)
    - Undersampling: $\Omega_s < 2\Omega_N$
    - Oversampling: $\Omega_s > 2\Omega_N$

Reconstruction of Bandlimited Signals

- Perfect reconstruction
  - Recovers the original continuous-time signal without distortion, e.g. ideal lowpass (bandpass) filter



- Based on frequency-domain analysis, if we can "clip" one copy of the original spectrum, $X_c(j\Omega)$, without distortion, we can achieve perfect reconstruction. For example, ideal lowpass filter, $h_r(t)$, can be used as a reconstruction filter
- Note that $x_s(t)$ is an analog signal

Signal Reconstruction Derivation

$$x_c(t) \rightarrow \text{sampling} \rightarrow x_s(t) = \sum_n x(nT)\delta(t-nT) \rightarrow \text{sequence conversion} \rightarrow x[n]$$

$$x[n] \rightarrow \text{impulse conversion} \rightarrow x_s(t) = \sum_n x[n]\delta(t-nT) \rightarrow \text{reconstruction} \rightarrow x_r(t)$$

$$x_r(t) = x_s(t) * h_r(t) = \int_\lambda \left\{ \sum_{n=-\infty}^{\infty} x[n]\delta(\lambda - nT) h_r(t-\lambda) \right\} d\lambda$$

$$= \sum_n \left\{ x[n] \int_\lambda \delta(\lambda - nT) h_r(t-\lambda) d\lambda \right\} = \sum_n x[n] h_r(t-nT)$$

Taking the Fourier transform of $x_r(t)$, we have

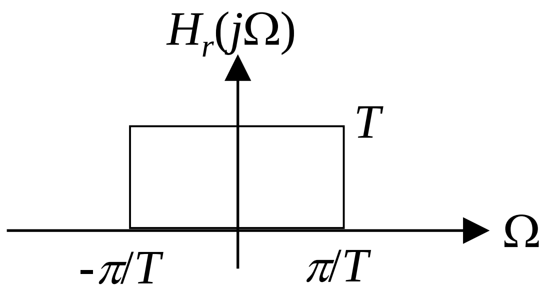
$$X_r(j\Omega) = \sum_n x[n] H_r(j\Omega) e^{-j\Omega T n} = H_r(j\Omega) \left\{ \sum_n x[n] e^{-j\Omega T n} \right\}$$

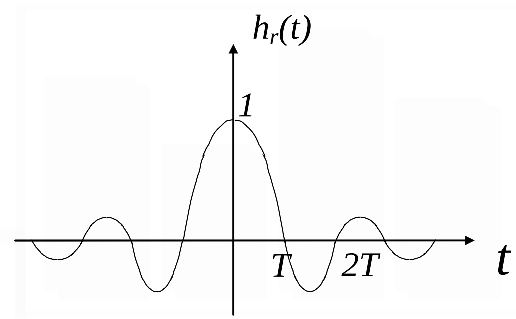
$$= H_r(j\Omega) X\left(e^{j\omega}\right)\Big|_{\omega=\Omega T} = H_r(j\Omega) X\left(e^{j\Omega T}\right) = H_r(j\Omega) X(j\Omega)$$

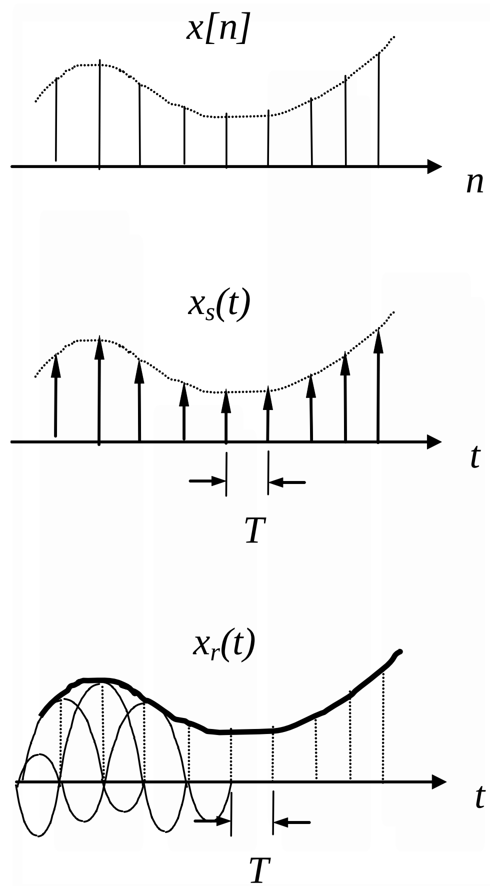
Ideal Lowpass Reconstruction Filter

Given: $H_r(j\Omega)=\begin{cases} T & -\pi/T<\Omega\le\pi/T \\ 0 & otherwise \end{cases} \Leftrightarrow h_r(t)=\dfrac{\sin(\pi t/T)}{\pi t/T}$

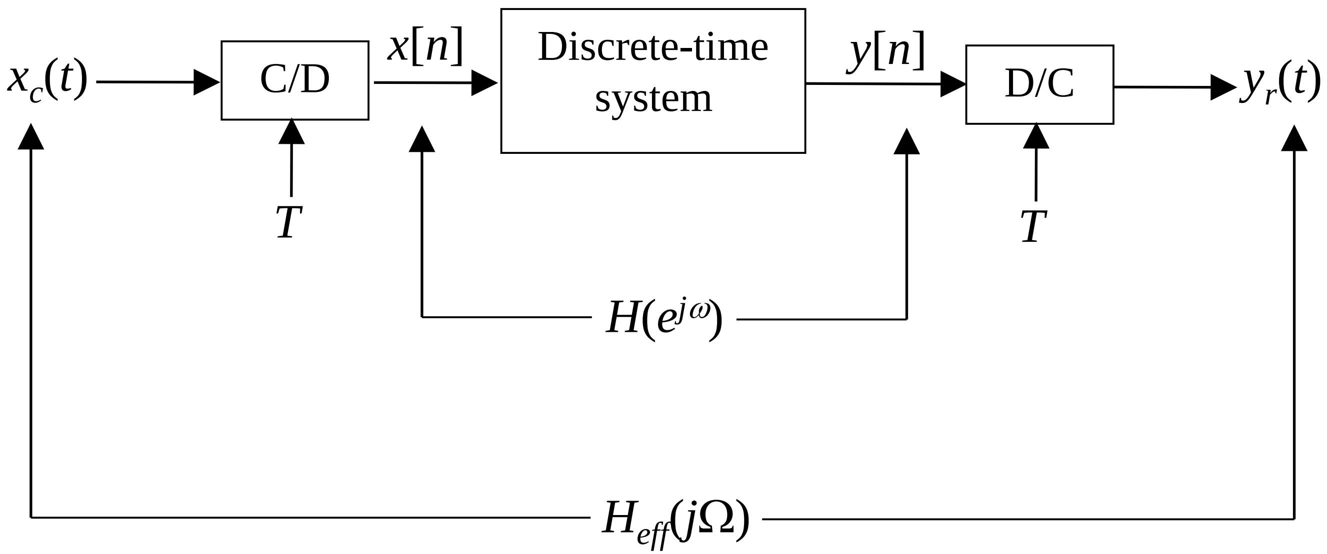
Then: $x_r(t)=\sum_n x[n]\dfrac{\sin\left[\dfrac{\pi(t-nT)}{T}\right]}{\dfrac{\pi(t-nT)}{T}}$


$H_r(j\Omega)$
T
Ω
$-\pi/T$
π/T


$h_r(t)$
1
T
$2T$
t


$x[n]$
n
$x_s(t)$
t
T
$x_r(t)$
t
T

Discrete-Time Processing of Continuous-Time Signals (DTPCTS)



DTPCTS

If this is an LTI system

(1) $x[n] \to y[n]$: $Y\left(e^{j\omega}\right) = H\left(e^{j\omega}\right) X\left(e^{j\omega}\right)$

(2) $x_c(t) \to x[n]$: $X\left(e^{j\omega}\right) = \frac{1}{T}\sum_k X_c\left(j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right)$

(3) $y[n] \to y_r(t)$: $Y_r(j\Omega) = H_r(j\Omega) Y\left(e^{j\Omega T}\right)$

(4) $x_c(t) \to \cdots \to y_r(t)$

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

If $H_r(j\Omega)$ is an ideal lowpass reconstruction filter, then

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

If other words, if $x_c(t)$ is *bandlimited* and is *ideally sampled* at a rate above the *Nyquist rate*, and the reconstruction filter is the *ideal lowpass* filter, then the *equivalent analog filter* has the same spectrum shape of the discrete-time filter.

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

Equivalency between $H(e^{j\omega})$ and $H_{eff}(j\Omega)$

- In order to have the above equivalent relation between $H(e^{j\omega})$ and $H_{eff}(j\Omega)$, we need
  - The system to be LTI
  - The input to be bandlimited
  - The input to be sampled without aliasing and the ideal impulse train to be used in sampling
  - The ideal reconstruction filter to be used to produce the analog output
- In practice, the above conditions are only approximately valid at best. However, there are methods in designing the sampling and the reconstruction processes to make the approximation better.

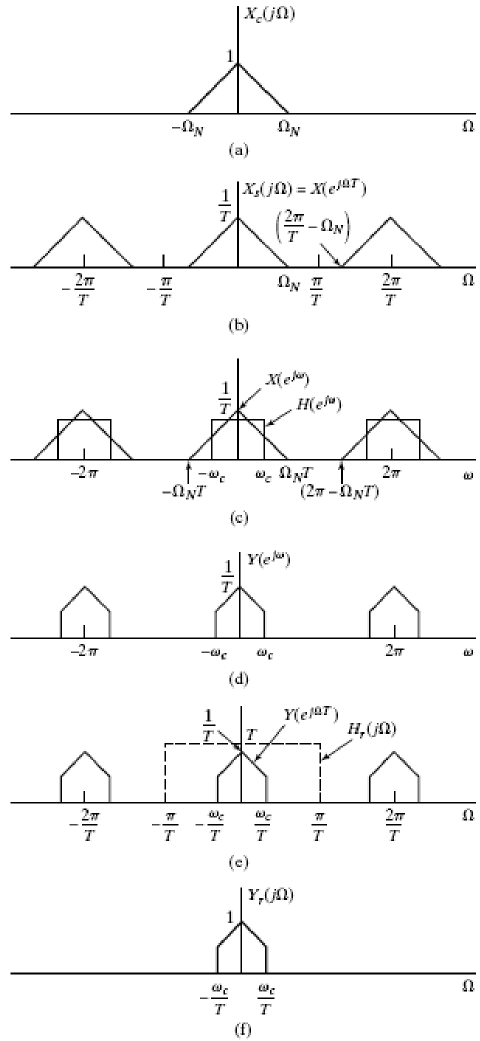


Figure 4.13 (a) Fourier transform of a bandlimited input signal. (b) Fourier transform of sampled input plotted as a function of continuous-time frequency Ω. (c) Fourier transform $X(e^{j\omega})$ of sequence of samples and frequency response $H(e^{j\omega})$ of discrete-time system plotted vs. ω. (d) Fourier transform of output of discrete-time system. (e) Fourier transform of output of discrete-time system and frequency response of ideal reconstruction filter plotted vs. Ω. (f) Fourier transform of output.

Design of Discrete-Time Filter $H(e^{j\omega})$

• One way to design the discrete-time filter is by first obtaining the impulse response of the analog filter $h_c(t)$, then simply sample that in the time domain to obtain $h[n]$. However, from the expressions above, $H_{eff}(j\Omega) = H_c(j\Omega)$, for $|\Omega| \leq \pi/T$, and 0 otherwise. The design problem becomes

Desire: $H_{eff}(j\Omega) = H_c(j\Omega)$ (because we want the entire DTPCTS system to approximate $H_c(j\Omega)$) or specifically

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

and with further requirement that T be chosen such that

$$H_c(j\Omega) = 0, \qquad |\Omega| \geq \frac{\pi}{T}. \qquad (\dagger)$$

Design of Discrete-Time Filter $H(e^{j\omega})$

But, what we actually have:

In the time domain, from sampling, we know that

$$h[n] = h_c(nT) \qquad (\dagger\dagger)$$

So in the frequency domain, we have

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

Then from (†), we have

$$H\left(e^{j\omega}\right) = \frac{1}{T} H_c\left(\frac{j\omega}{T}\right), \qquad |\omega| \le \pi \qquad (\dagger\dagger\dagger)$$

However, we want

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

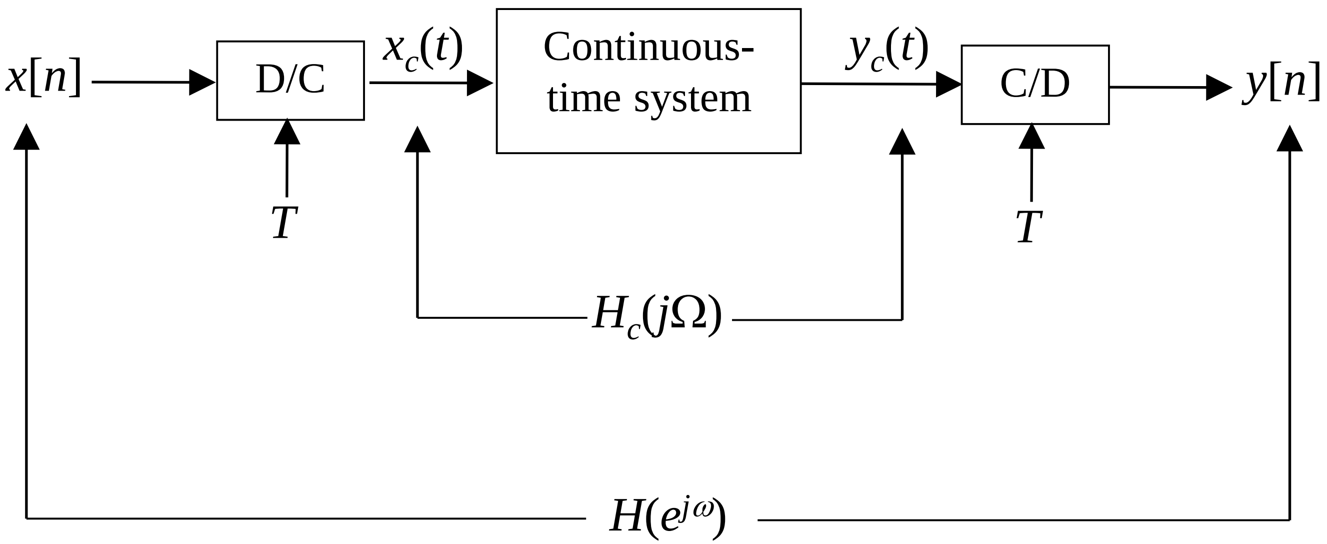
Then modifying $(\dagger\dagger)$ and $(\dagger\dagger\dagger)$ to account for the scale factor T

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

$$h[n] = Th_c(nT)$$

Therefore, the impulse response of the discrete-time system is a scaled, sampled version of $h_c(t)$.

Continuous-Time Processing of Discrete-Time Signals (CTPDTS)



System Equations for CTPDTS

$$X_c(j\Omega) = TX(e^{j\Omega T}), \qquad \text{for } |\Omega| < \frac{\pi}{T}$$

$$Y_c(j\Omega) = H_c(j\Omega)X_c(j\Omega), \qquad \text{for } |\Omega| < \frac{\pi}{T}$$

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

$$\Rightarrow \quad Y\left(e^{j\omega}\right) = \frac{1}{T}H_c\left(j\frac{\omega}{T}\right)X_c\left(j\frac{\omega}{T}\right) = H_c\left(j\frac{\omega}{T}\right)X\left(e^{j\omega}\right), \qquad \text{for } |\omega| < \pi$$

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

$$\left(\text{since } \frac{1}{T}X_c\left(\frac{j\omega}{T}\right) = X\left(e^{j\omega}\right)\right)$$

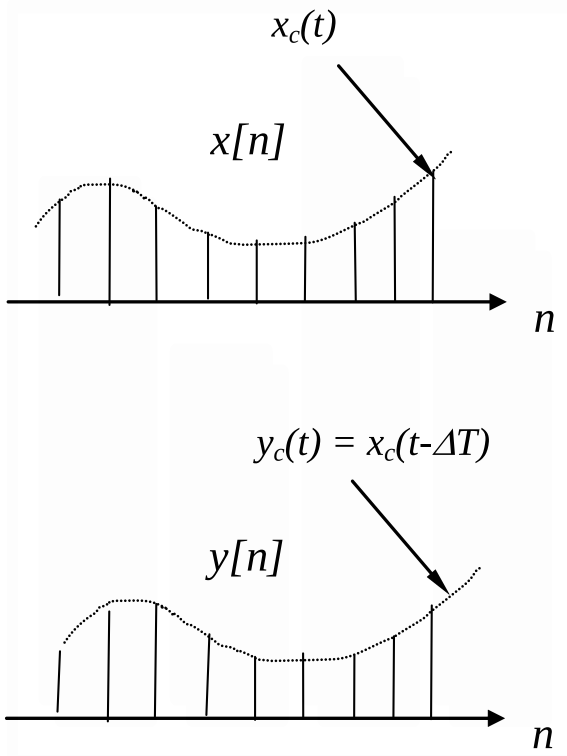
or equivalently $\quad H(e^{j\Omega T}) = H_c(j\Omega), \qquad \text{for } |\Omega| < \frac{\pi}{T}$

Example: Non-Integer Delay

- Let $y[n] = x[n-\Delta]$, where $\Delta \in \mathbb{R}$.
  No formal meaning in the time domain, but we can interpret this in the frequency domain if we let
  $$H(e^{j\omega}) = e^{-j\omega\Delta}, \text{ for } |\omega| < \pi,$$
  or equivalently,
  $$H(e^{j\Omega T}) = e^{-j\Omega\Delta T} = H_c(j\Omega), \text{ for } |\Omega| < \pi/T.$$
  Therefore, $y_c(t) = x_c(t-\Delta T)$. With this, then if $\Delta=1/2$, then we can interpret $y[n]$ as a bandlimited interpolation halfway between the input sequence values since $y[n]$ is just a sampled version of $y_c(t)$.
  Since $y[n]$ and $x[n]$ are sampled version of $y_c(t)$ and $x_c(t)$, respectively, therefore,
  $$\begin{aligned} y[n] &= y_c(nT) = x_c(nT - \Delta T) \\ &= \sum_k x[k] \left. \frac{\sin[\pi(t-\Delta T - kT)/T]}{\pi(t-\Delta T-kT)/T} \right|_{t=nT} \\ &= \sum_k x[k] \frac{\sin[\pi(n-k-\Delta)]}{\pi(n-k-\Delta)}. \end{aligned}$$

Example: Non-Integer Delay

So $$h[n] = \frac{\sin \pi(n-\Delta)}{\pi(n-\Delta)}, \quad \text{for } -\infty < n < \infty,\ n \in \mathbb{Z}.$$



Changing of Sampling Rate Using Discrete-Time Processing

Idea:

$$x_c(t)\ \begin{cases} \rightarrow T \rightarrow & x[n] = x_c(nT) \\ \rightarrow T' \rightarrow & x'[n] = x_c(nT') \end{cases}$$

Original sampling period: T

New sampling period: T', $\qquad T \neq T'$

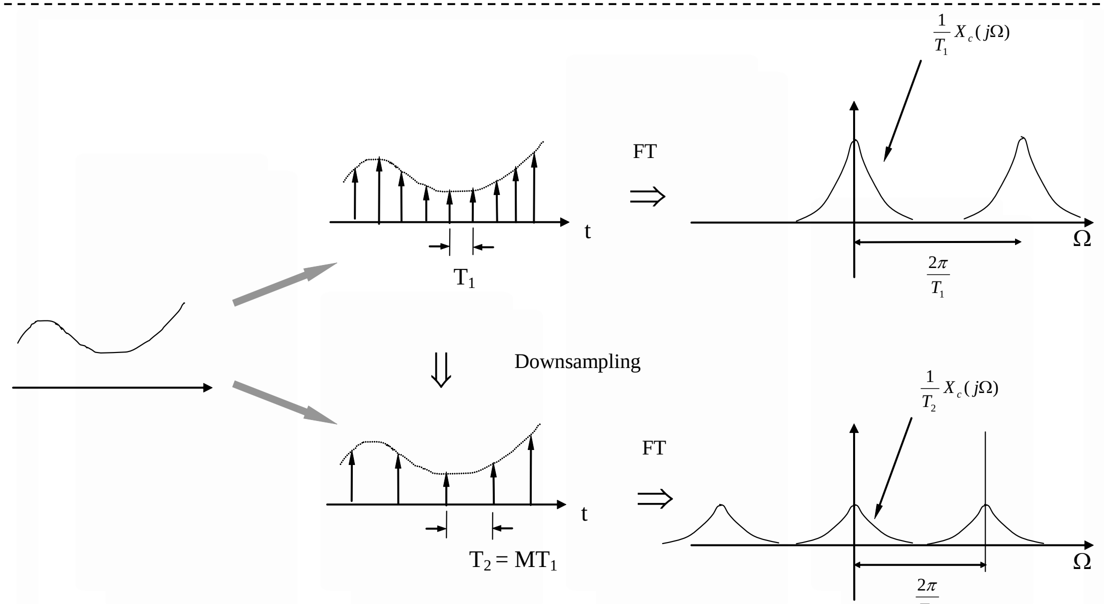
Sampling Rate Reduction By An Integer Factor

Sampling rate compressor:

$$T' = MT, \quad \text{where } M \text{ is an integer}$$

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

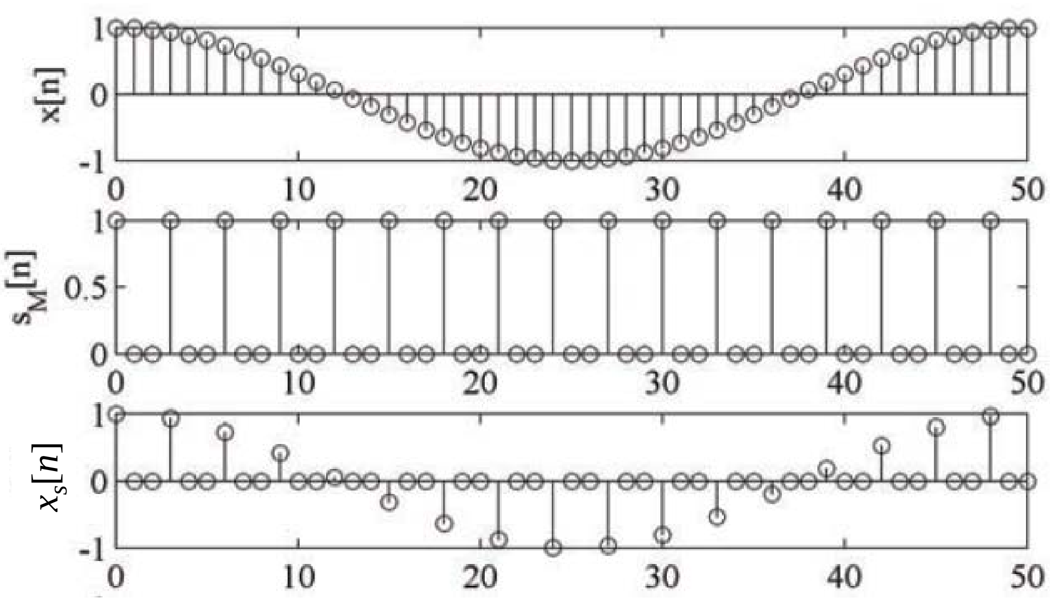
$$x[n] \longrightarrow \boxed{\downarrow M} \longrightarrow x_d[n] = x[nM] = x_c(nMT)$$



Aliasing: If the original signal BW is not small enough to meet the Nyquist rate requirement, prefiltering is needed.

Frequency-Domain Representation

Since $x_d[n] = x[nM]$. Define a sampling sequence $s_M[n] = \begin{cases} 1, & n = 0, \pm M, \pm 2M, \ldots, \\ 0, & \text{otherwise} \end{cases}$, so that $x_s[n] = x[n]s_M[n] = \begin{cases} x[n], & n = 0, \pm M, \pm 2M, \ldots, \\ 0, & \text{otherwise} \end{cases}$

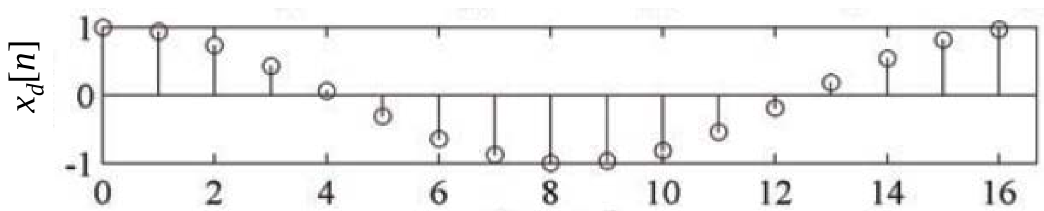


Frequency-Domain Representation

- $\Rightarrow x_d[n] = x_s[nM] = x[nM]$

  So the downsampling operation involves zeroing out samples in $x[n]$ AND compressing the resulting sequence $x_s[nM]$ into $x_d[n]$
- We like to express $X_d(z)$ to $X(z)$ by using the intermediate signal $X_s(z)$
- So $X_d(z) = \sum_n x[nM] z^{-n}$. We want to express right-hand side in terms of $X(z)$. Can use the relation $x_s[nM] = x[nM]$ to do that.

$$X_d(z) = \sum_n x[nM] z^{-n} = \sum_n x_s[nM] z^{-n}$$



Frequency-Domain Representation

- Since zero values of $x_s[n]$ are going to be discarded:

$$X_d(z) = \sum_n x_s[nM] z^{-n} = \sum_k x_s[k] z^{-k/M}$$
$$= X_s\left(z^{1/M}\right)$$

E.g. let $M = 3$: $\begin{cases} x_s[0]z^0 = x_s[0], & k=0 \\ x_s[1]z^{-1/3} = 0 \cdot z^{-1/3} = 0, & k=1 \\ x_s[2]z^{-2/3} = 0 \cdot z^{-2/3} = 0, & k=2 \\ x_s[3]z^{-3/3} = x_s[3]z^{-1}, & k=3 \end{cases}$. Can now see that the exponent in z corresponds to only correct nonzero sample of $x_s[n]$ which are also the same samples in $x[n]$

- Since $x_s[n] = x[n] s_M[n]$, we can express $X_s(z)$ in terms of $X(z)$ as $X_s(z) = \sum_n s_M[n] x[n] z^{-n}$.

Frequency-Domain Representation

$$X_s(z) = \sum_n s_M[n]x[n]z^{-n}$$

$$X_d(z) = \sum_n x_s[nM]z^{-n} = \sum_k x_s[k]z^{-k/M} = X_s\left(z^{1/M}\right)$$

- Note that $s_M[m] = \sum_p \delta[m - pM]$. Define the DFT of $s_M[m]$ as

$$S_M[k] = \sum_{m=0}^{M-1}\left(\sum_p \delta[m-pM]\right)e^{-j\frac{2\pi mk}{M}} = \sum_p \sum_{m=0}^{M-1} \delta[m-pM]e^{-j\frac{2\pi mk}{M}} = \sum_{m=0}^{M-1}\delta[m]e^{-j\frac{2\pi mk}{M}} = 1$$

Then the IDFT is $s_M[n] = \frac{1}{M}\sum_{k=0}^{M-1} 1\cdot e^{j\frac{2\pi nk}{M}} = \frac{1}{M}\sum_{k=0}^{M-1} e^{j\frac{2\pi nk}{M}} = \frac{1}{M}\sum_{k=0}^{M-1} W_M^{-kn}$

- Then $X_s(z) = \frac{1}{M}\sum_n \sum_{k=0}^{M-1} x[n]W_M^{-kn}z^{-n} = \frac{1}{M}\sum_{k=0}^{M-1}\sum_n x[n]\left(zW_M^k\right)^{-n} = \frac{1}{M}\sum_{k=0}^{M-1} X\left(zW_M^k\right)$
- $\Rightarrow X_d(z) = X_s\left(z^{1/M}\right) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(z^{1/M}W_M^k\right)$

Frequency-Domain Representation

$$X_d(z) = X_s\left(z^{1/M}\right) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(z^{1/M} W_M^k\right)$$

This implies $X_d\left(e^{j\omega}\right) = X_s\left(e^{j\omega/M}\right)$

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

- $x_d[n]$ is identical to the sequence that would be obtained from $x_c(t)$ by using the sampling period $T' = MT$.
- If $X_c(j\Omega)=0$, for $|\Omega| \geq \Omega_N$, then $x_d[n]$ is an exact representation of $x_c(t)$ if $\frac{\pi}{T'} = \frac{\pi}{MT} \geq \Omega_N$

  I.e. sampling rate is reduced by a factor of M without aliasing if the original rate was at least M times the Nyquist rate of if the bandwidth of the sequence is first reduced by a factor of M by DT filtering
- $X_d\left(e^{j\omega}\right)$ is equal to M copies of $X\left(e^{j\omega}\right)$ scaled by factor M and shifted by 2π. That is, first we stretched the x-axis by a factor of M, then to produce the copies, we modulate the stretched signal by an integer multiple of 2π.

Frequency-Domain Representation

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

Recall $X\left(e^{j\omega}\right) = \frac{1}{T}\sum_{k} X_c\left(j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right)$. Since $x_d[n] = x[nM] = x_c(nMT)$,

$\Rightarrow \omega = \frac{\Omega}{T'} = \frac{\Omega}{MT}$, then $X_d\left(e^{j\omega}\right) = \frac{1}{MT}\sum_{k} X_c\left(j\left(\frac{\omega}{MT} - \frac{2\pi k}{MT}\right)\right)$

The downsampled spectrum = sum of shifted replica of the original (M=2)

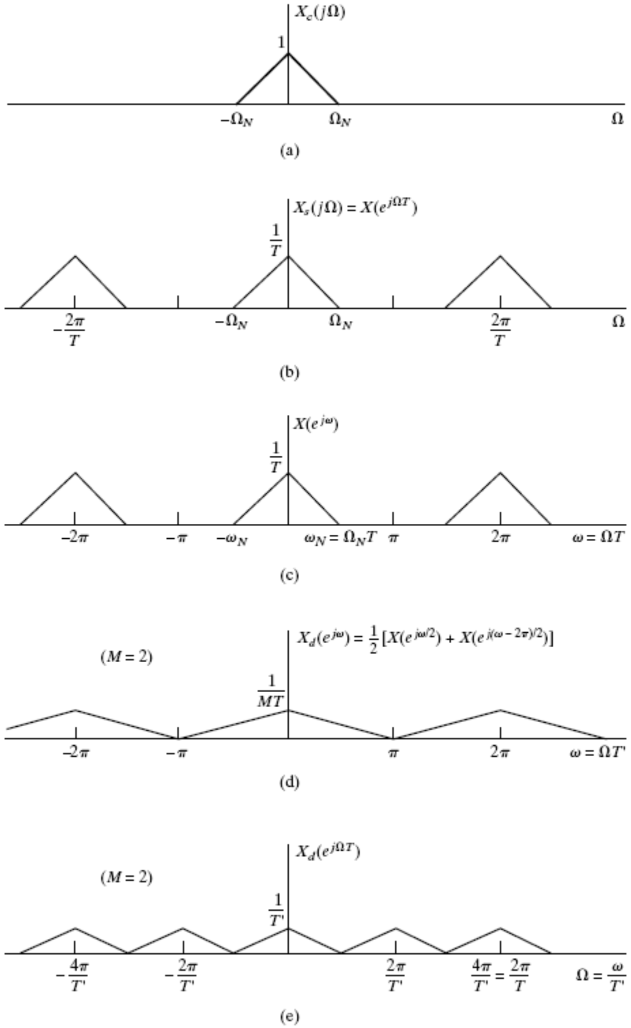


Figure 4.21 Frequency-domain illustration of downsampling.

Nyquist theorem: $\Omega_s = \dfrac{2\pi}{T} \geq 2\Omega_N$

If $\Omega_s = \dfrac{2\pi}{T} = 4\Omega_N \quad (M = 2)$

$\Rightarrow \omega_N = \Omega_N T = \dfrac{\pi}{2}$

$\therefore$ Bandlimit $x_c(t)$ by $\dfrac{\pi}{2T}$ to avoid aliasing

or $x[n]$ by $\dfrac{\pi}{2}$

$\Rightarrow$ In general, bandlimit $x[n]$ by a LPF with cutoff frequency $\omega_c = \dfrac{\pi}{M}$ and amplitude = 1

Graphical illustration for M = 3 (no aliasing)

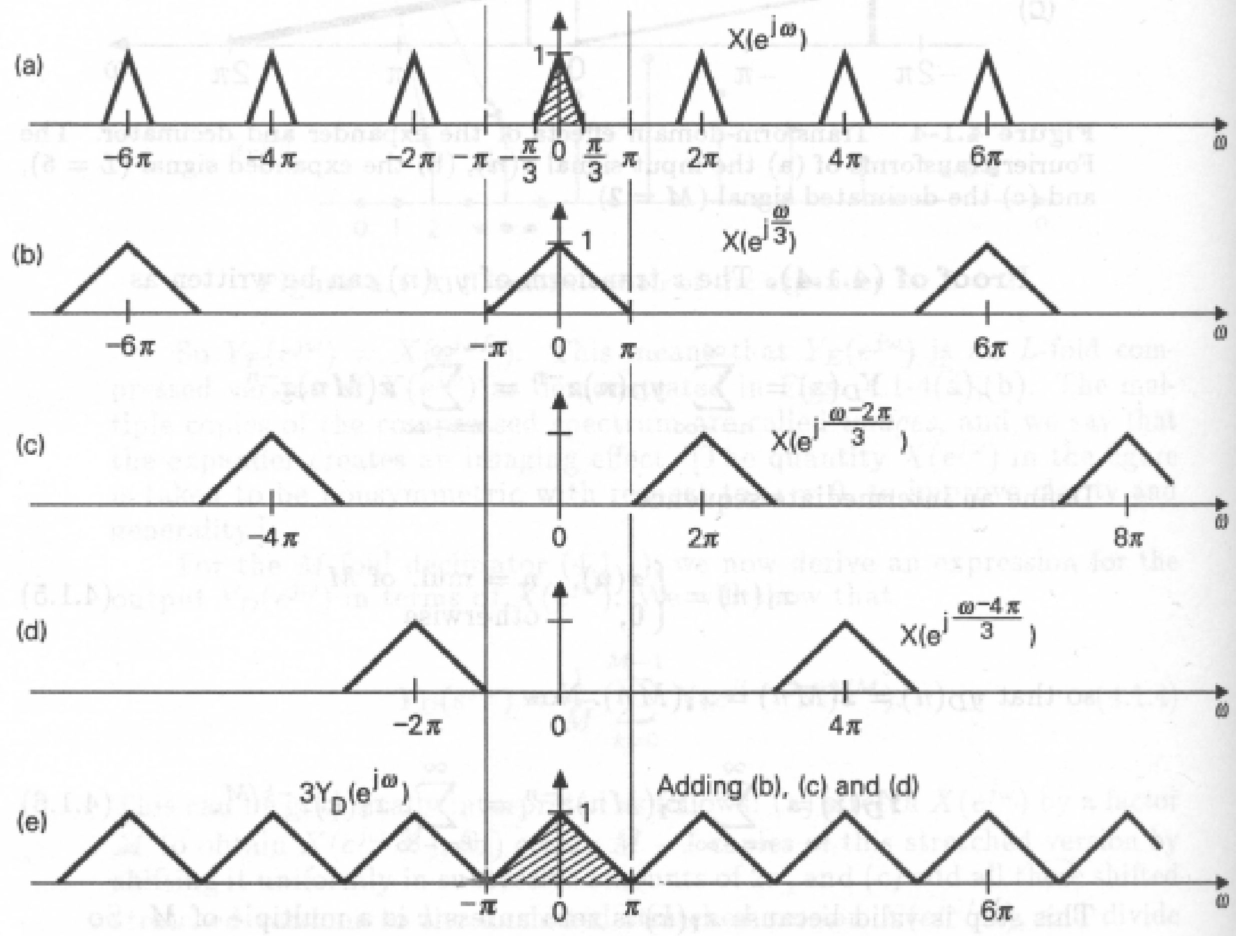


Figure 4.1-5 Demonstrating the frequency-domain effect of decimation with $M = 3$.

Downsampling with aliasing

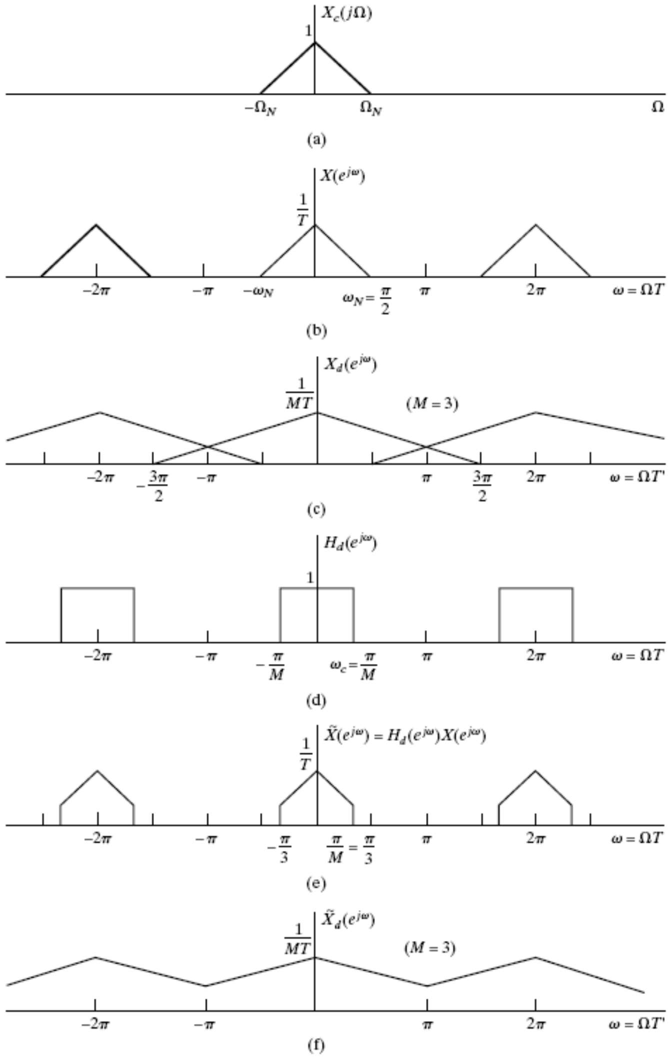
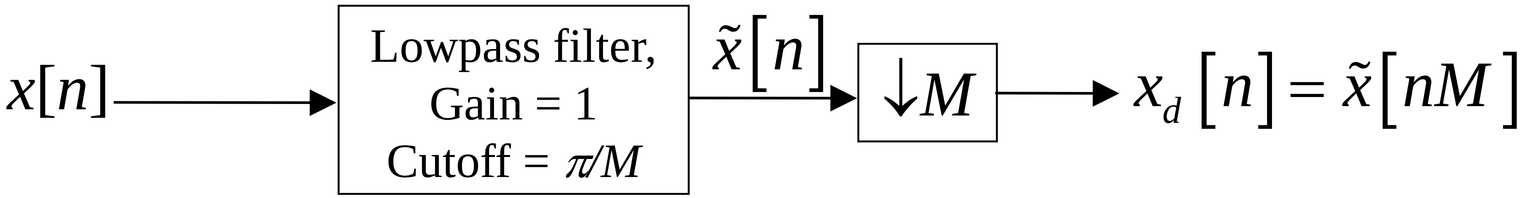


Figure 4.22 (a)–(c) Downsampling with aliasing. (d)–(f) Downsampling with prefiltering to avoid aliasing.

How Do We Avoid Aliasing?

To avoid aliasing => $\omega_N M < \pi$



General System for Sampling Rate Reduction by M

$$\tilde{x}[n] = \sum_k x[k]h[n-k]$$

$$\Rightarrow x_d[n] = \tilde{x}[Mn] = \sum_k x[k]h[Mn-k]$$

Convolution matrix

$$\mathbf{H}_d = \begin{bmatrix} h[0] & 0 & 0 & 0 & 0 & 0 \\ h[M] & h[M-1] & h[M-2] & 0 & 0 & 0 \\ h[2M] & h[2M-1] & h[2M-2] & h[2M-3] & h[2M-4] & 0 \\ 0 & 0 & h[2M] & h[2M-1] & h[2M-2] & h[2M-3] \\ 0 & 0 & 0 & 0 & h[2M] & h[2M-1] \\ 0 & 0 & 0 & 0 & 0 & h[2M] \end{bmatrix}$$

E.g. $M = 2, h[n] = [1 \ \ 2 \ \ 3 \ \ 4 \ \ 5]^H, L_x = 10$

$$\mathbf{H}_d = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 4 & 3 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 4 & 3 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 4 & 3 & 2 & 1 & 0 \end{bmatrix}$$

Is Decimation Time-Invariant?

$$x_d[n] = \tilde{x}[nM] = \sum_k x[k]h[nM-k]$$

Suppose we delay $x[n]$ by $n_0 \Rightarrow \tilde{x}[n] = \sum_k x[k-n_0]h[n-k]$,

let $p = k - n_0 \Rightarrow k = p + n_0$, $\tilde{x}[n] = \sum_p x[p]h[n-n_0-p]$

$$\Rightarrow x_d[n] = \sum_k x[p]h[nM - n_0 - k]$$

Suppose we delay after: $\Rightarrow x_d[n-n_0] = \sum_k x[k]h[nM - n_0M - k]$

$\therefore$ Unless n_0 on top equals integer multiple of M (n_0M at the bottom), else decimation is time-varying (it is block-invariant)

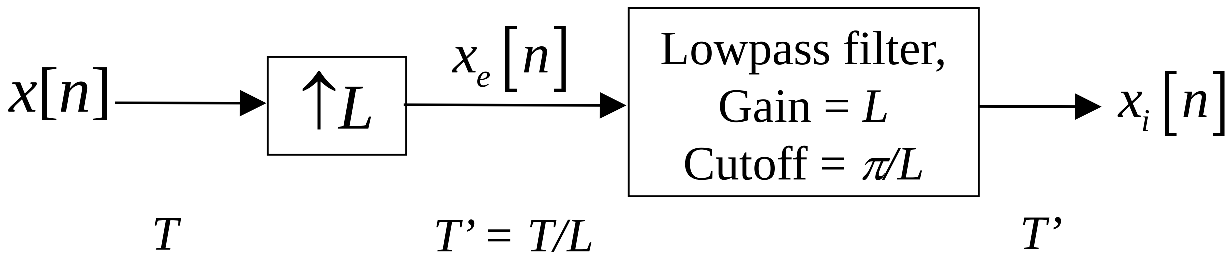
From this, it is clear that the rate reduction system above is time-varying. To see this, input the signal $x[n-n_0]$ into the above rate reduction system and you will see that the result does not match $x_d[n-n_0]$, unless n_0 is a multiple of M.

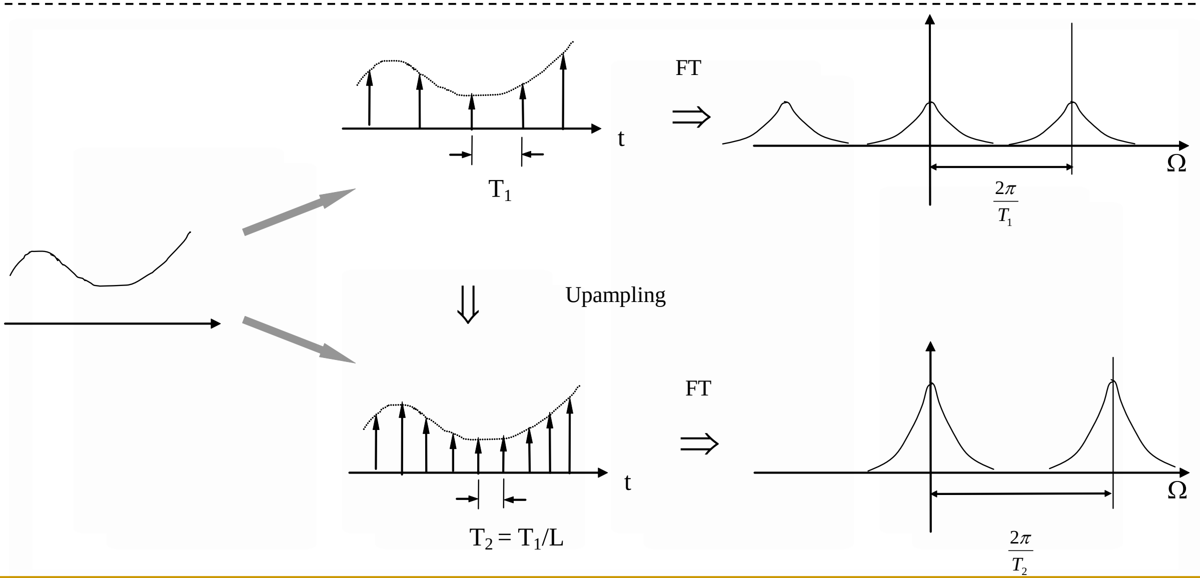
Sampling Rate Enlargement By An Integer Factor

Sampling rate expander:

$$T' = T / L, \quad \text{where } L \text{ is an integer}$$

$$x_e[n] = x\left[\frac{n}{L}\right] = x_c\left(n\frac{T}{L}\right)$$





(1) shape is compressed; (2) replicas are removed

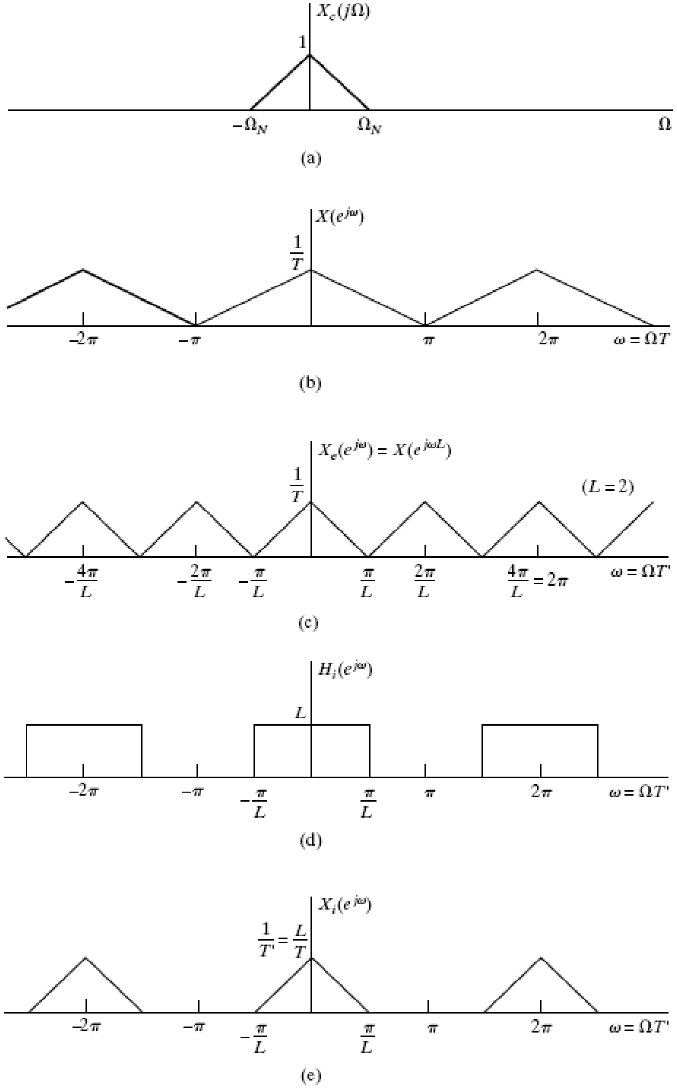


Figure 4.25 Frequency-domain illustration of interpolation.

Mathematical Representation

1) Increase samples (Time-domain)

$$x_e[n] = \begin{cases} x\left[\dfrac{n}{L}\right], & n = 0, \pm L, \pm 2L, \cdots \\ 0, & otherwise \end{cases}$$

$$= \sum_k x[k]\delta[n-kL]$$

(Frequency-domain)

$$X_e(e^{j\omega}) = \sum_n \left(\sum_k x[k]\delta[n-kL] \right) e^{-j\omega n}$$

$$= \sum_k x[k] \underbrace{\left(\sum_n \delta[n-kL] e^{-j\omega n} \right)}_{=e^{-j\omega kL}} = X\left(e^{j\omega L}\right)$$

Note that $\sum_n \delta[n-kL]e^{-j\omega n} = e^{-j\omega Lk}$

Remark: Essentially, the horizontal frequency axis is compressed.
The shape of the spectrum is not changed.

Remark: At this point, we only insert zeros into the original signal.
In time domain, this signal doesn't look like the original.

Mathematical Representation

1) Let assume ideal lowpass filtering (frequency-domain)

$$H_i\left(e^{j\omega}\right)=\begin{cases}1, & -\pi/L<\omega\le\pi/L,\\ 0, & otherwise\end{cases} \Leftrightarrow h_i[n]=\frac{\sin(\pi n/L)}{(\pi n/L)}$$ (this is an interpolator)

2) Many ways to develop relationship between $x_i[n]$ and $x[n]$

1st way: $x_i[n]=x_e[n]*h[n]$ and $x_e[n]=x\left[\frac{n}{L}\right]$, for $n=0,\pm1,\pm2,\ldots$

$$x_i[n]=\sum_k x_e[k]h_i[n-k]=\sum_k x\left[\frac{k}{L}\right]h_i[n-k]$$

$$=\sum_p x[p]h_i[n-pL]$$

2nd way:

$$\sum_p x_e[n-p]h_i[p]=\sum_p\left(\sum_k x[k]\delta[n-p-kL]\right)h_i[p]$$

$$=\sum_k x[k]\left(\sum_p \delta[n-p-kL]h_i[p]\right)=\sum_k x[k]\left(\sum_p \delta\left[p-(n-kL)\right]h_i[p]\right)$$

$$=\sum_k x[k]h_i[n-kL]=\sum_{k=-\infty}^{\infty} x[k]\frac{\sin\left[\pi(n-kL)/L\right]}{\pi(n-kL)/L}$$

Is Interpolator Time-Invariant?

$$x_i[n] = \sum_p x[p] h_i[n - pL]$$

Delay first then transform: $x_i[n] = \sum_p x[p - n_0] h_i[n - pL]$, let $k = p - n_0 \Rightarrow p = k + n_0$

$$\Rightarrow x_i[n] = \sum_k x[k] h_i[n - n_0 L - kL]$$

Transform first then delay: $x_i[n] = \sum_p x[p] h_i[n - pL]$

$$\Rightarrow x_i[n - n_0] = \sum_p x[p] h_i[n - n_0 - pL]$$

$\therefore$ Similar to the decimator, the interpolator is not time-invariant nnless n_0 is an integer multiple of L ($n_0 L$ at the top). It is block-invariant.

Matrix Representation

Upsample following by filtering by $h[n]$

$$\mathbf{H}_u = \begin{bmatrix} h[0] & 0 & 0 & 0 \\ h[1] & 0 & 0 & 0 \\ h[2] & h[2-L] & 0 & 0 \\ h[3] & h[3-L] & 0 & 0 \\ h[4] & h[4-L] & h[4-2L] & 0 \\ h[5] & h[5-L] & h[5-2L] & 0 \\ h[6] & h[6-L] & h[6-2L] & h[6-3L] \\ h[7] & h[7-L] & h[7-2L] & h[7-3L] \end{bmatrix}$$

E.g. $L = 2,\ h[n] = [1 \ \cdots \ 8]^H,\ L_x = 5$

$$\mathbf{H}_u = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \\ 4 & 2 & 0 & 0 & 0 \\ 5 & 3 & 1 & 0 & 0 \\ 6 & 4 & 2 & 0 & 0 \\ 7 & 5 & 3 & 1 & 0 \\ 8 & 6 & 4 & 2 & 0 \\ 0 & 7 & 5 & 3 & 1 \\ 0 & 8 & 6 & 4 & 2 \\ 0 & 0 & 7 & 5 & 3 \\ 0 & 0 & 8 & 6 & 4 \\ 0 & 0 & 0 & 7 & 5 \\ 0 & 0 & 0 & 8 & 6 \\ 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Linear Interpolation

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

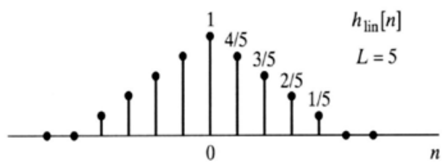


Figure 4.26 Impulse response for linear interpolation.

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

$$x_{lin}[n] = \sum_{k=-\infty}^{\infty} x[k]h_{lin}[n-kL]$$

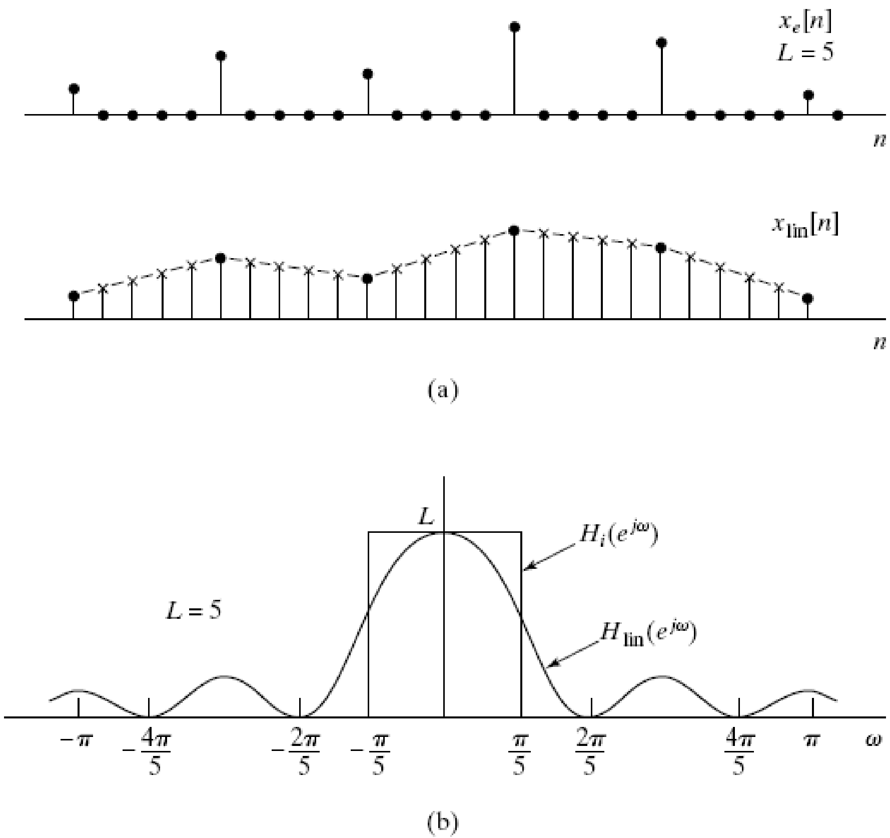


Figure 4.27 (a) Illustration of linear interpolation by filtering. (b) Frequency response of linear interpolator compared with ideal lowpass interpolation filter.

Changing Sampling Rate By a Rational Factor

Idea: Sampling period $\quad T \xrightarrow{\text{interpolation}} \dfrac{T}{L} \xrightarrow{\text{decimation}} \dfrac{M}{L}T$

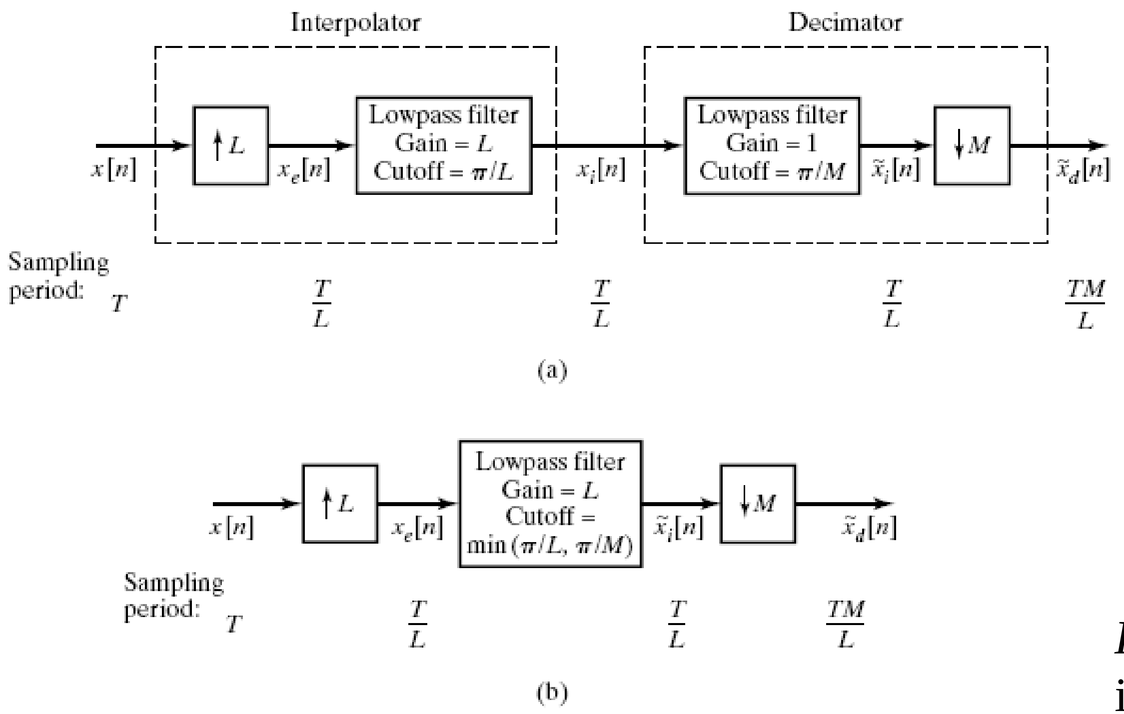


Figure 4.28 (a) System for changing the sampling rate by a noninteger factor. (b) Simplified system in which the decimation and interpolation filters are combined.

Remark: In general, if the factor is not rational, go back to the continuous signals.

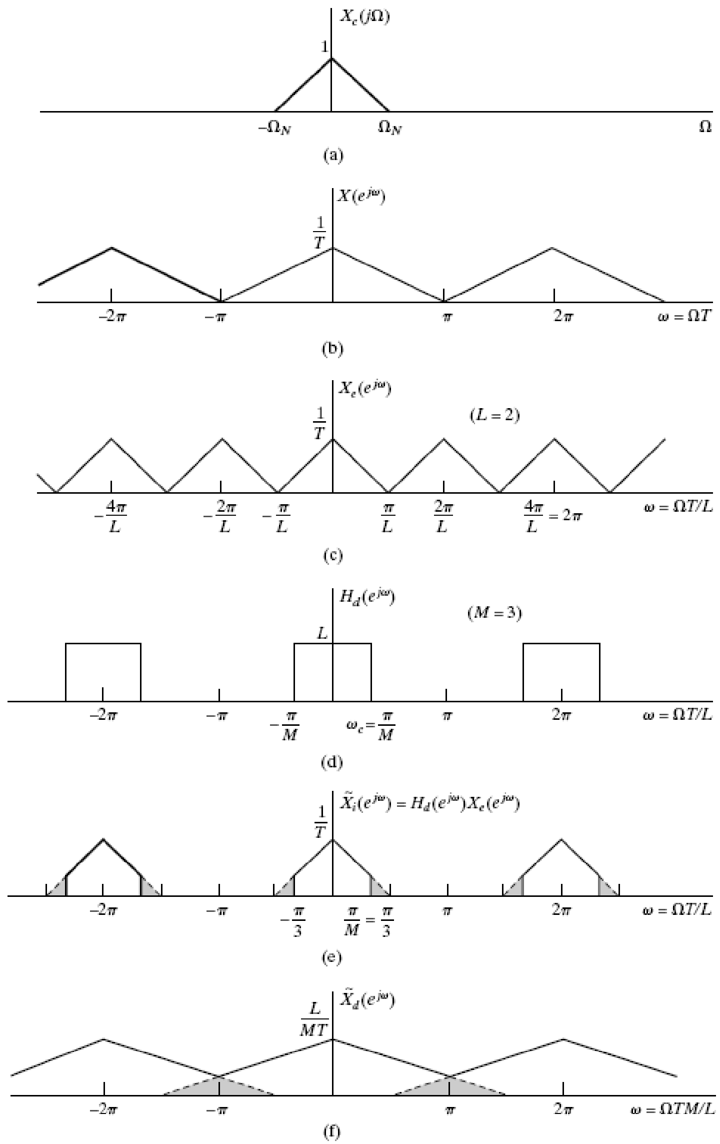


Figure 4.29 Illustration of changing the sampling rate by a noninteger factor.

Summary

Sampling

Time-domain	Frequency-domain
Prefiltering	Limit bandwidth $\Omega_s > 2\Omega_N$
Analog sampling (impulse train)	Duplicate and shift Ω
Analog to discrete $\delta(t) \rightarrow \delta[n]$	$\Omega \rightarrow \omega$

Reconstruction

Time-domain	Frequency-domain
Analog to discrete $\delta[n] \rightarrow \delta(t)$	$\omega \rightarrow \Omega$
Interpolation	Remove extra copies (Ω)

Decimation

Time-domain	Frequency-domain
Prefiltering	Limit bandwidth
Drop samples (rearrange index)	Expand (by a factor M) and duplicate (insert (M-1) copies)

Interpolation

Time-domain	Frequency-domain
Insert zeros	Shrink (by a factor of L)
Interpolation	Remove extra copies in a 2π period

Digital Processing of Analog Signals

Ideal C/D converter → (approximation) analog-to-digital (A/D) converter

Ideal D/C converter → (approximation) digital-to-analog (D/A) converter

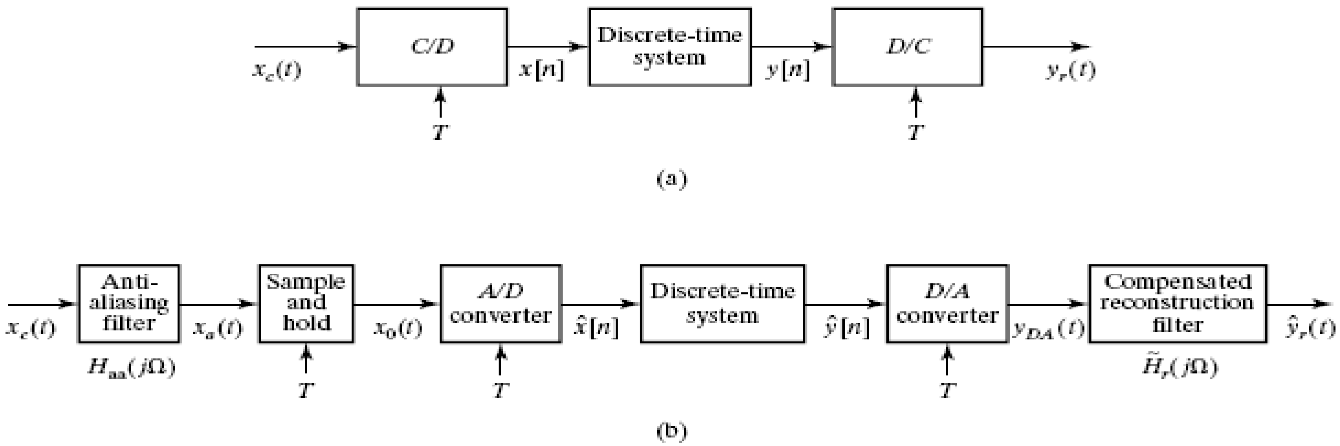


Figure 4.41 (a) Discrete-time filtering of continuous-time signals. (b) Digital processing of analog signals.

Prefiltering

Ideal antialiasing filter: Ideal lowpass filter (difficult to implement sharp cutoff analog filters)

➔Solution: simple prefilter and oversampling followed by sharp antialiasing filters in discrete-time domain

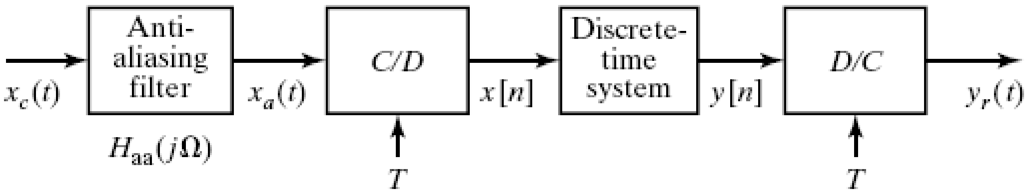


Figure 4.42 Use of prefiltering to avoid aliasing.

Remark:

- Sharp cutoff analog filters are expensive and difficult to implement.
- Passband of sharp cutoff analog filter is often non-linear phase because IIR is required.

Discrete-Time Solution To Avoid Aliasing

- To reduce cost, an antialiasing filter is designed to have a gradual cutoff at $\Omega_c = M\Omega_N$ (instead of π/T) so that this can be easily implemented using analog circuitry
- $x_a(t)$ is sampled at T such that $\left(\frac{2\pi}{T} - \Omega_c\right) > \Omega_N$. In fact, we want to implement the C/D block at sampling rate $\Omega_S >> 2\Omega_N$, e.g. at $\Omega_S = 2M\Omega_N$ so that

$$T = \frac{2\pi}{\Omega_s} = \frac{2\pi}{2M\Omega_N} = \frac{1}{M}\left(\frac{\pi}{\Omega_N}\right)$$

This makes sure that only the "noise" portion of the signal (or unwanted high frequency component) is corrupted, but not the actual signal

- This then is followed by a rate reduction by a factor of M that includes a sharp antialiasing filter at $\omega = \pi/M$. This is done because we can easily implement digital filters with sharp cutoff (see Ch. 7).
- Now we can downsample by M to obtain $x_d[n]$. T and T' are chosen such that $T' = MT$ and $\pi/T = \Omega_N$. This makes it possible for to be filtered with a cutoff frequency at $\omega = \pi/M$
- Note that the "noise" is aliased but won't affect the signal band $|\omega| < \omega_N = \Omega_N T$

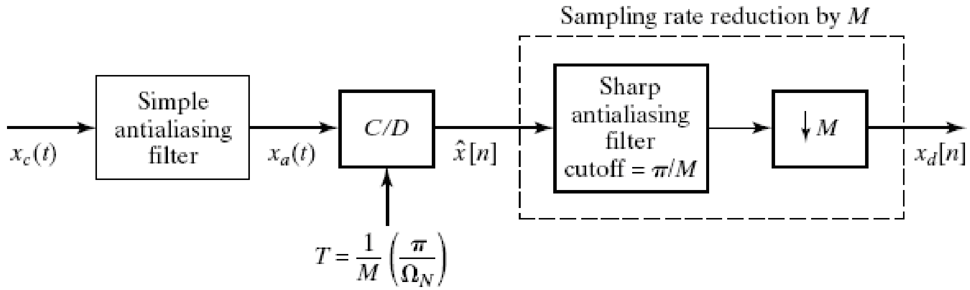
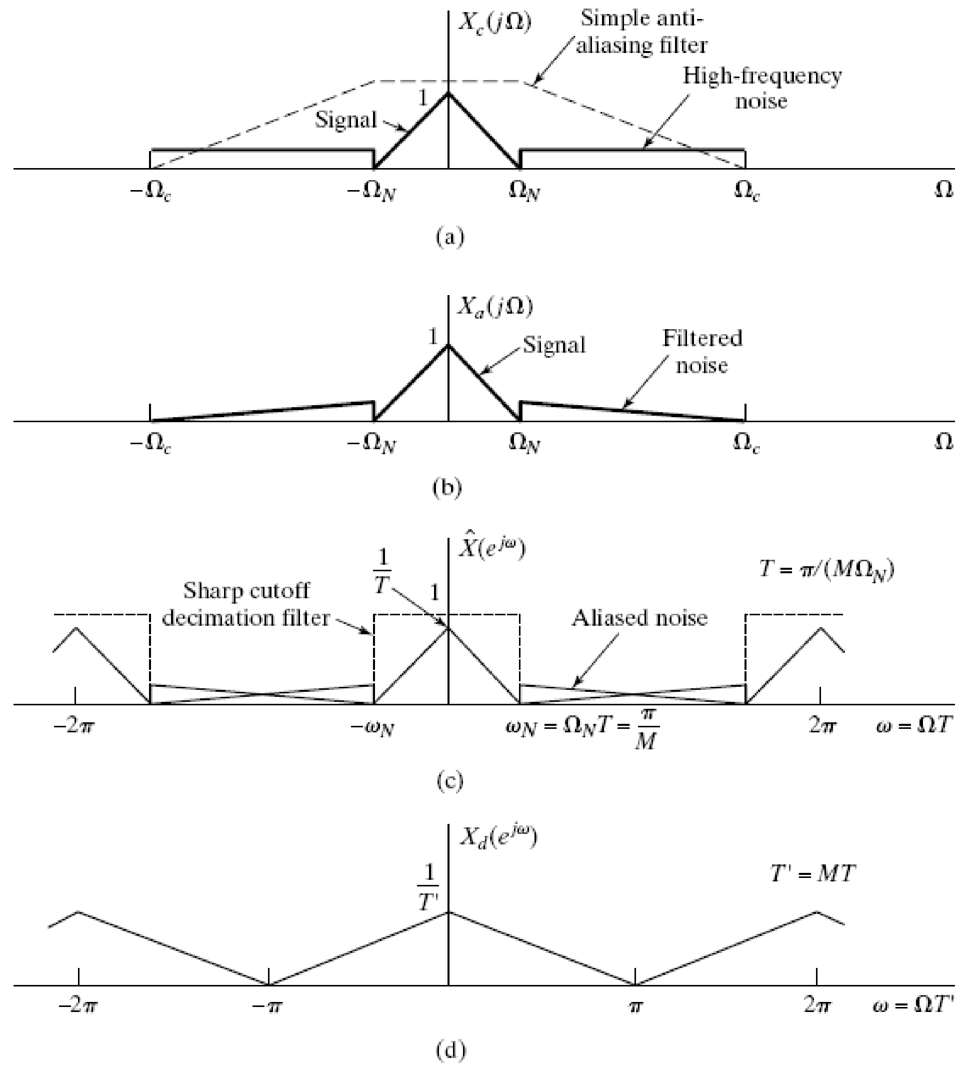


Figure 4.43 Using oversampled A/D conversion to simplify a continuous-time antialiasing filter.

Oversample by M:
$T = 2\pi/\Omega_N$ vs.
$T = 2\pi/2M\Omega_N$

Oversample by M:
$T=2\pi/\Omega_N$ *vs.*
$T=2\pi/2M\Omega_N$



When $T=2\pi/\Omega_N$:
aliasing

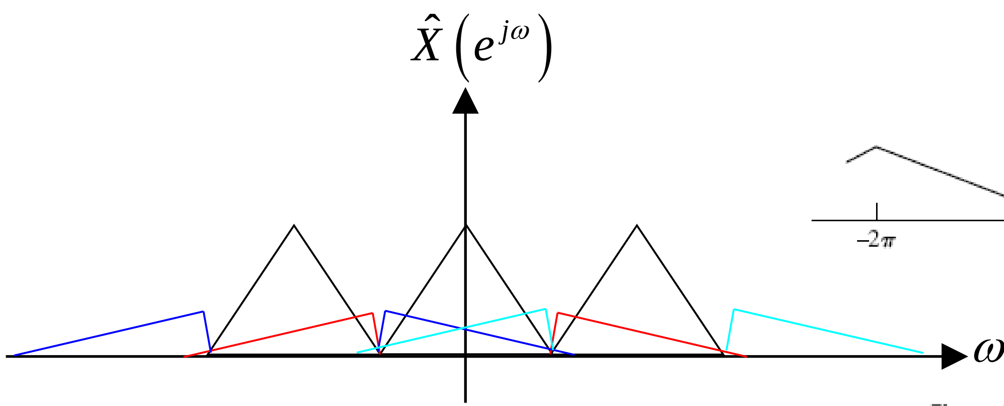


Figure 4.44 Use of oversampling followed by decimation in C/D conversion.

A/D Conversion

Digital: discrete in *time* and *discrete* in *amplitude*

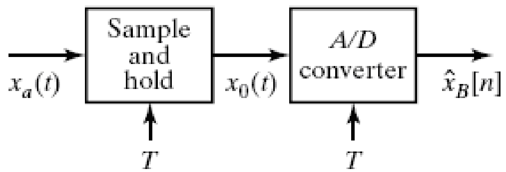


Figure 4.45 Physical configuration for analog-to-digital conversion.

Ideal sample-and-hold: Sample the (input) analog signal and hold its value for T seconds. This is used because the A/D process is not instantaneous.

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

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

$$x_0(t) = \sum_{n=-\infty}^{\infty} x_a(nT)h_0(t-nT)$$

$$= \left\{ \sum_{n=-\infty}^{\infty} x_a(nT)\delta(t-nT) \right\} * h_0(t)$$

Thus, the sample and hold can be regarded as an impulse train modulation followed by filtering with $h_0(t)$. This is shown in Fig. 4.46a below. Thus the frequency response relationship between $x_0(t)$ and $x_a(t)$ is similar to the one between $x[n]$ and $x_c(\mathrm{t})$ in the previous section on sampling.

Sample and hold

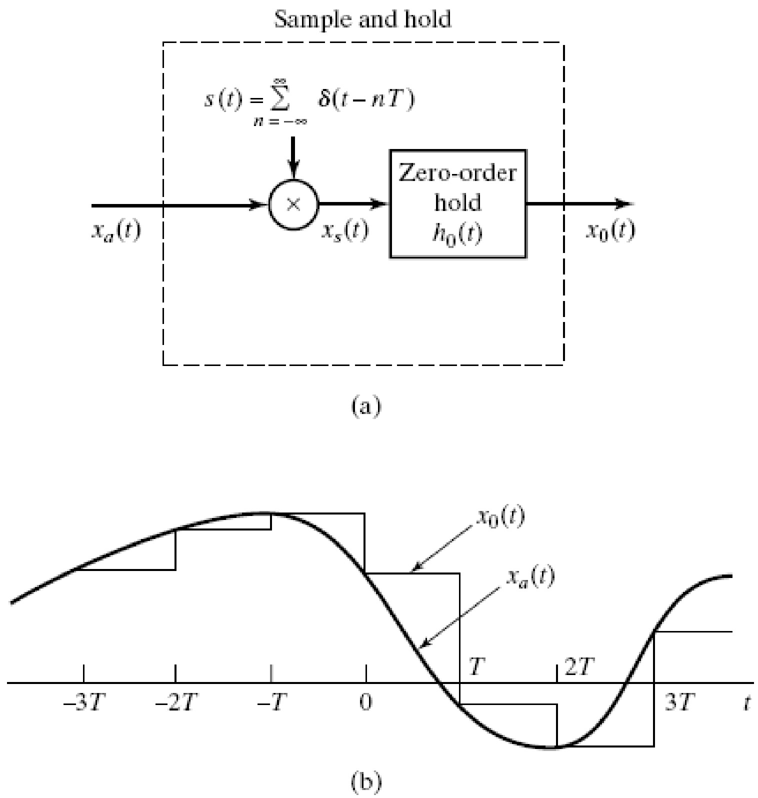
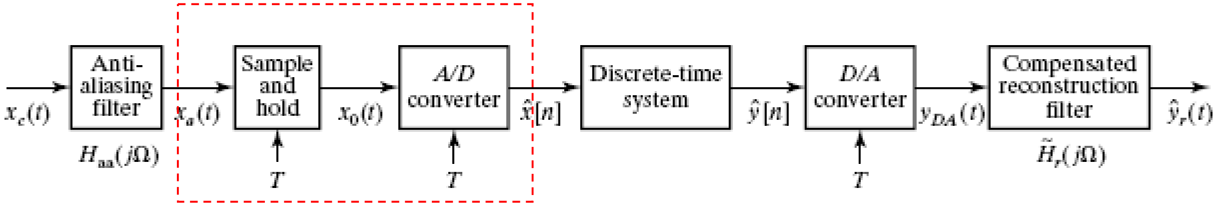


Figure 4.46 (a) Representation of an ideal sample-and-hold.
(b) Representative input and output signals for the sample-and-hold.

Quantization



- Transform the input sample $x[n]$ (continuous in amplitude) into one of a discrete variable in a finite set of prescribed values
  - Quantization is an non-linear operation
  - Figure 4.45 can now be represented by Figure 4.47 where the ideal C/D converter represents the sampling performed by the sample-and-hold
  - The C/D, quantizer, and coder together represent the operation of the sample and hold, and A/D converter

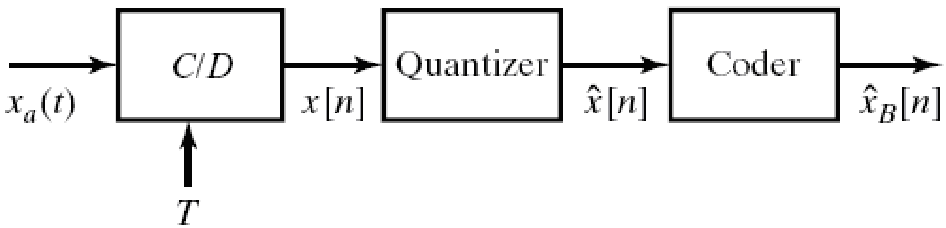
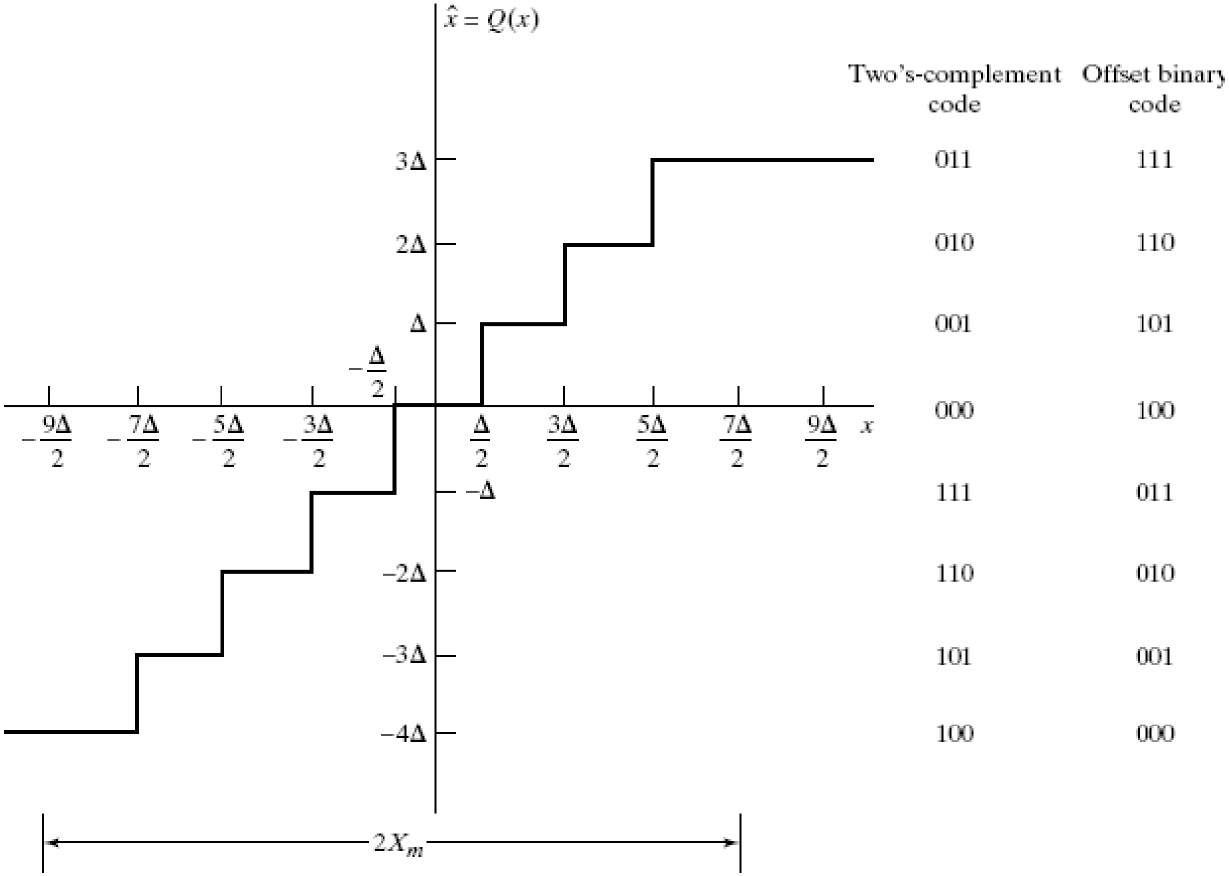


Figure 4.47 Conceptual representation of the system in Figure 4.45.



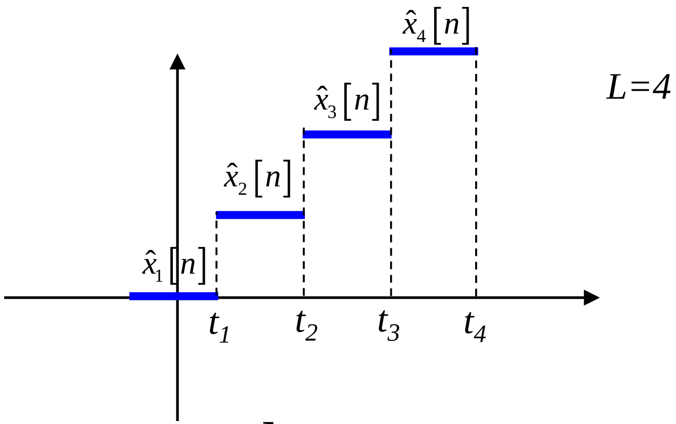
Example: 3-bit uniform quantizer

Figure 4.48 shows a 3-bit quantizer, where $B+1=3$. $\Delta = \dfrac{2X_m}{2^{B+1}} = \dfrac{X_m}{2^B}$

is defined as the distance between the transition levels

Figure 4.48 Typical quantizer for A/D conversion.

Quantizer Parameters



- Parameters
  - Decision/transition levels
    - Partition the dynamic range of input signal
    - Define as t_k, for $k = 1, 2, \ldots L/2\text{-}1$ (using the example in Fig. 4.48), $L = 8$. Considering only the positive side as negative side is identical
  - Quantization/reconstruction levels
    - The output values of a quantizer; a quantization level represents all samples between two nearby decision levels

$$\hat{x}[n] = Q(x[n]), \quad \text{if } t_k \le x[n] \le t_{k+1}, \text{ where } \hat{x}[n] \text{ is the quantized sample}$$

Note: It is actually more precise to express $\hat{x}[n]$ as $\hat{x}_k[n]$.

Quantization Error for the Uniform Quantizer

- Quantized samples and the true sample $x[n]$ are different due to the use of the quantizer. This difference can be expressed as the quantization error $e[n] = \hat{x}[n] - x[n]$

  For the case of the 3-bit quantizer as shown in Fig. 4.48, i.e. $B+1 = 3$, if $\Delta/2 < x[n] \le 3\Delta/2$, then it is clear that $-\frac{\Delta}{2} < e[n] < \frac{\Delta}{2}$. This will hold for the 3-bit quantizer with a dynamic range of $-9\Delta/2 < x[n] \le 7\Delta/2$
- This implies $\hat{x}[n] = x[n] + \frac{\Delta}{2}$ (at most)
- Statistical characteristics of :
  - $e[n]$ is stationary (probability distribution unchanged)
  - $e[n]$ is uncorrelated with $x[n]$
  - $e[n]$, $e[n+1]$, … are uncorrelated (white)
  - $e[n]$ has a uniform distribution
- The preceding characteristics are (approximately) valid if the signal is sufficiently *complex* and the quantization steps are sufficiently *small*

Example

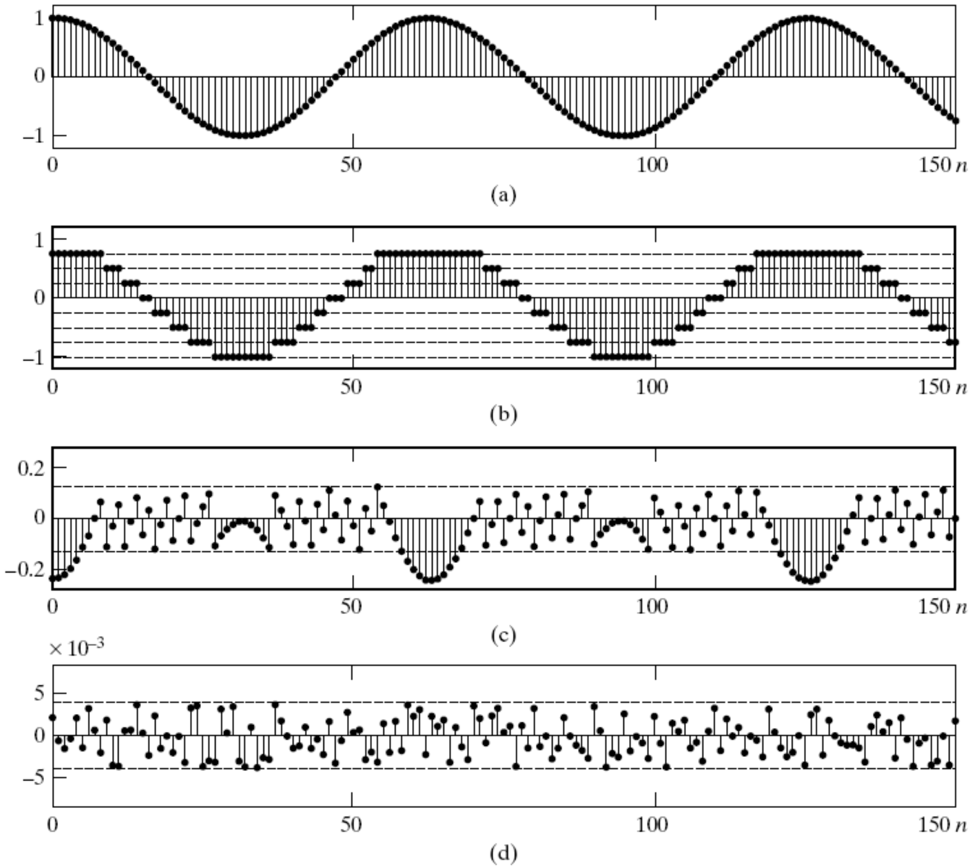


Figure 4.51 Example of quantization noise. (a) Unquantized samples of the signal $x[n] = 0.99\cos(n/10)$. (b) Quantized samples of the cosine waveform in part (a) with a 3-bit quantizer. (c) Quantization error sequence for 3-bit quantization of the signal in (a). (d) Quantization error sequence for 8-bit quantization of the signal in (a).

- $x[n] = 0.99\cos(n/10)$
- using 3-bit and 8-bit quantizer ($B+1 = 3$ and 8). $X_m = 1$.
- scale of the quantization error is adjusted so that the range $\pm\Delta/2$ is indicated by the dashed lines
- for 3-bit quantizer, the error is highly correlated with the quantized signal. Also the intervals around the positive peaks, the error is greater than $\Delta/2$ in magnitude because the signal level is too large for this setting of the quantizer parameters.
- For 8-bit quantizer, the error is not correlated with the unquantized signal. The range of the error is kept between $-\Delta/2$ and $+\Delta/2$.

Error Analysis: Mean-Square Error

- MSE of $e[n]$ (= variance if zero-mean)

$$\sigma_e^2 = E\left\{(e-\overline{e})^2\right\} = \int_{-\Delta/2}^{\Delta/2} e^2 \frac{1}{\Delta} de = \frac{\Delta^2}{12}$$

- Since $\Delta = X_m/2^B$, then

$$\sigma_e^2 = \frac{2^{-2B} X_m^2}{12}$$

- SNR (signal-to-noise) ratio due to quantization

$$SNR = 10\log_{10}\frac{\sigma_x^2}{\sigma_e^2} = 10\log_{10}\frac{12\cdot 2^{2B}\sigma_x^2}{X_m^2} = 10.8 + 6.02B - 20\log_{10}\frac{X_m}{\sigma_x}$$

  σ^2 is the RMS value of the signal amplitude

  X_m is a parameter of the ADC and it would be fixed in a practical system
- Remarks
  - One bit buys a 6dB SNR improvement
  - If the input is Gaussian, a small percentage (0.064%) of the input samples would have an amplitude greater than $4\sigma_x$. If σ_x is chosen to be $X_m/4$, SNR ≈ $6B$-1.25 dB. Then we can avoid clipping the peaks of the signal
  - For example, a 90 to 96 dB SNR requires a 16-bit quantizer

Design and Error Analysis of Quantizers

- Quantizer design
  - Optimal quantizer design requirement PDF of data
  - Choice of quantizer:
    - Uniform
      - Popular – easily to do
      - Good if the signal PDF is uniform (most likely not optimal in any sense)
    - Non-uniform
      - Lloyd-max (optimal mean-squared error design) (uniform is a special case)
        - Given L and $p_X(x[n])$, we desire

$$\min_{t_k, \hat{x}_k[n]} \varepsilon = \sum_{k=1}^{L} \int_{t_k}^{t_{k+1}} \left(x[n] - \hat{x}_k[n]\right)^2 p_X\left(x[n]\right) dx$$

Lloyd-Max Quantizer (Optimal in MSE sense)

Note that

$$\varepsilon = E\left[\left(x[n]-\hat{x}_k[n]\right)^2\right] = \int_{x[n]}\left(x[n]-\hat{x}_k[n]\right)^2 p_X\left(x[n]\right)dx$$

Therefore, the optimal MSE quantizer needs to minimize ε, i.e.

$$\min_{t_k,\hat{x}_k[n]} \varepsilon = \min_{t_k,\hat{x}_k[n]} \int_{x[n]}\left(x[n]-\hat{x}_k[n]\right)^2 p_X\left(x[n]\right)dx$$

Assuming that $p_X\left(x[n]\right)$ is Gaussian, then

$$\int_{x[n]}\left(x[n]-\hat{x}_k[n]\right)^2 p_X\left(x[n]\right)dx = \underbrace{\int_{-\infty}^{t_1}\left(x[n]-\hat{x}_0[n]\right)^2 p_X\left(x[n]\right)dx}_{=0} + \int_{t_1}^{t_2}\left(x[n]-\hat{x}_1[n]\right)^2 p_X\left(x[n]\right)dx + \cdots + \underbrace{\int_{t_L}^{\infty}}_{=0}$$

$$= \sum_{k=1}^{L}\int_{t_k}^{t_{k+1}}\left(x[n]-\hat{x}_k[n]\right)^2 p_X\left(x[n]\right)dx$$

$$\Rightarrow \frac{d\varepsilon}{dt_k} = \frac{d}{dt_k}\left[\int_{t_{k-1}}^{t_k}\left(x[n]-\hat{x}_{k-1}[n]\right)^2 p_X\left(x[n]\right)dx + \int_{t_k}^{t_{k+1}}\left(x[n]-\hat{x}_k[n]\right)^2 p_X\left(x[n]\right)dx\right] = 0$$

$$= \underbrace{\left(t_k-\hat{x}_{k-1}[n]\right)^2 p_X\left(x[n]\right)} - \underbrace{\left(t_k-\hat{x}_k[n]\right)^2 p_X\left(x[n]\right)} = 0$$

Recall that

$$\frac{d}{dt}\int_t^a f(x)\,dx = \frac{d}{dt}\left[F(a)-F(t)\right]$$

$$= -\frac{d}{dt}F(t)$$

$$= -f(t)$$

Lloyd-Max Quantizer

Note that

$$\frac{d\varepsilon}{dt_k} = \left(t_k - \hat{x}_{k-1}[n]\right)^2 p_X\left(x[n]\right) - \left(t_k - \hat{x}_k[n]\right)^2 p_X\left(x[n]\right) = 0$$

$$\Rightarrow \left(t_k - \hat{x}_{k-1}[n]\right)^2 = \left(t_k - \hat{x}_k[n]\right)^2$$

$$\Rightarrow \begin{cases} t_k - \hat{x}_{k-1}[n] = t_k - \hat{x}_k[n] & \Rightarrow \quad \hat{x}_{k-1}[n] = \hat{x}_k[n] \quad \text{(impossible!)} \\ t_k - \hat{x}_k[n] = -\left(t_k - \hat{x}_{k-1}[n]\right) & \Rightarrow \quad t_k = \dfrac{\hat{x}_{k-1}[n] + \hat{x}_k[n]}{2} \end{cases}$$

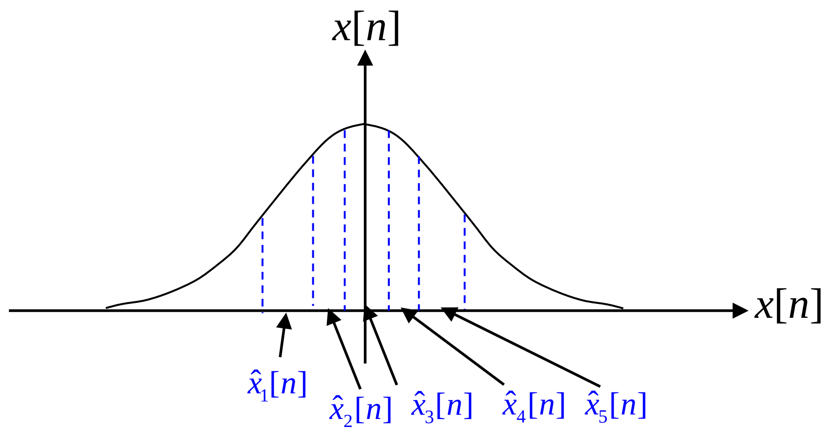
$\therefore$ The optimal MSE quantizer should design t_k such that it is halfway in between two adjacent quantization levels

$$\frac{d\varepsilon}{d\hat{x}_k[n]} = 2\int_{t_k}^{t_{k+1}} \left(x[n] - \hat{x}_k[n]\right) p_X\left(x[n]\right) dx = 0$$

$$\Rightarrow \hat{x}_k[n] = \frac{\int_{t_k}^{t_{k+1}} x[n]\, p_X\left(x[n]\right) dx}{\int_{t_k}^{t_{k+1}} p_X\left(x[n]\right) dx}$$

$\therefore$ The optimal MSE quantizer should design $\hat{x}_k[n]$ such that it is at the centroid of its region

Decision Level for Lloyd-Max Quantizer (Gaussian Distribution)



5-level Lloyd-Max quantizer for Gaussian-distributed signal $x[n]$

Note: The decision levels are closer together in areas of higher probability

Uniform Quantizer (Special case of Lloyd Max)

Suppose $p_X\left(x[n]\right)=\frac{1}{L}$. Compute $\hat{x}_k[n]$ using the relationships derived above for the Lloyd-Max quantizer.

D/A Conversion

- Previously, we've discussed how a bandlimited signal can be reconstructed using an ideal D/C converter from a sequence of samples using ideal lowpass filtering. The reconstruction is represented as: $X_r(j\Omega) = X(e^{j\Omega T})H_r(j\Omega)$, where $H_r(j\Omega)$ is our ideal lowpass filter.

$$H_r(j\Omega) = \begin{cases} T, & |\Omega| < \dfrac{\pi}{T} \\ 0, & |\Omega| > \dfrac{\pi}{T} \end{cases} \Leftrightarrow \frac{\sin(\pi t / T)}{\pi t / T}$$

- A physically realizable counterpart of to the D/C converter is the D/A converter followed by a realizable lowpass filter
- Examples of practical filters: *zero-order hold* and *first-order hold*.

Mathematical Model for D/A

- The D/A takes a sequence of binary code words as its input and produces a continuous-time output of the form

$$x_{DA}(t) = \sum_{n=-\infty}^{\infty} X_m \hat{x}_B[n] h_0(t-nT) = \sum_{n=-\infty}^{\infty} \hat{x}[n] h_0(t-nT)$$

= quantized input * impulse response of zero-order hold (see p. 52)

- Using the additive-noise model above, we can represent the effect of the quantization as

$$x_{DA}(t) = \sum_{n=-\infty}^{\infty} x[n] h_0(t-nT) + \sum_{n=-\infty}^{\infty} e[n] h_0(t-nT)$$
$$= x_0(t) \quad + \quad e_0(t)$$

- *Purpose:* Find a compensation filter to compensate for the distortion caused by the non-ideal $h_0(t)$ so that its output is close to the analog original . *$e_0(t)$* can only be reduced by increasing the number of bits we used for the quantizer.

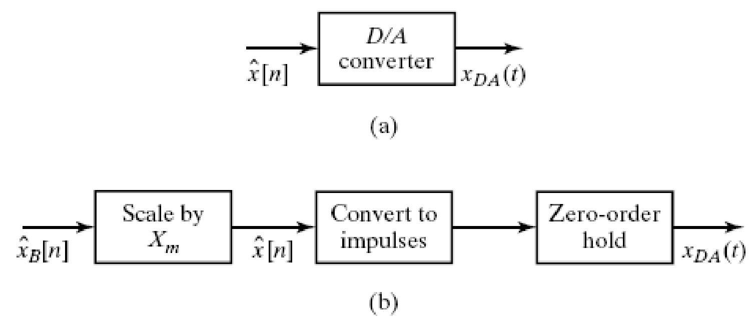
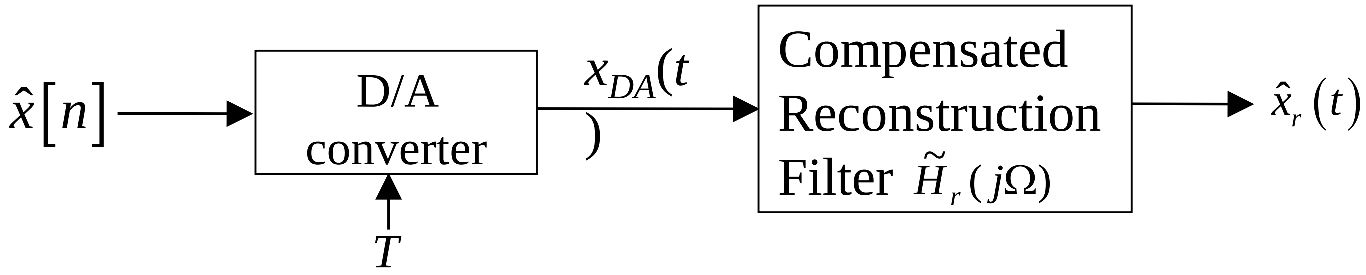


Figure 4.53 (a) Block diagram of D/A converter. (b) Representation in terms of a zero-order hold.

D/A Reconstruction Filter



In frequency domain

$$X_0(j\Omega) = \sum_{n=-\infty}^{\infty} x[n] H_0(j\Omega) e^{-j\Omega nT} = \left(\sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega nT} \right) H_0(j\Omega) = X(e^{j\Omega T}) H_0(j\Omega)$$

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

$\Rightarrow \quad X_0\left(j\Omega\right) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_a\left(j\left(\Omega - \frac{2\pi k}{T} \right)\right) H_0\left(j\Omega\right)$ as the interpolation filter $H_0\left(j\Omega\right)$ is used to remove the replicas

If $H_0\left(j\Omega\right)$ is not an ideal lowpass filter, we design a compensated reconstruction filter,

$$\tilde{H}_r\left(j\Omega\right) = \frac{H_r\left(j\Omega\right)}{H_0\left(j\Omega\right)},$$

where $H_r\left(j\Omega\right)$ is the ideal lowpass filter.

D/A Reconstruction Filter

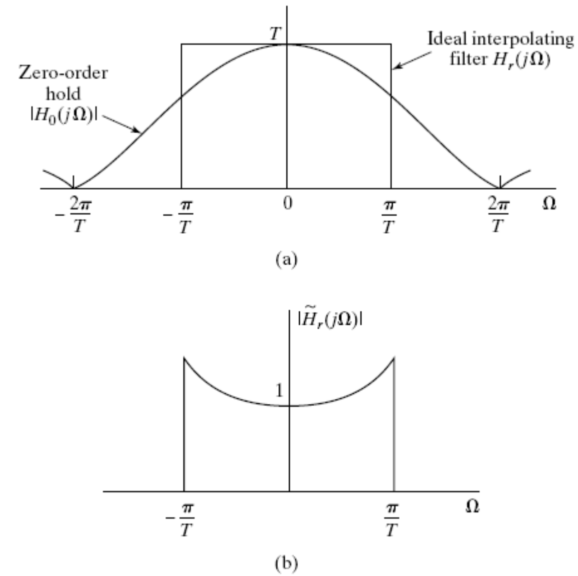


Figure 4.54 (a) Frequency response of zero-order hold compared with ideal interpolating filter. (b) Ideal compensated reconstruction filter for use with a zero-order-hold output.

Then the output of the filter will be $x_a(t)$ if the input is $x_0(t)$

Zero - order hold:

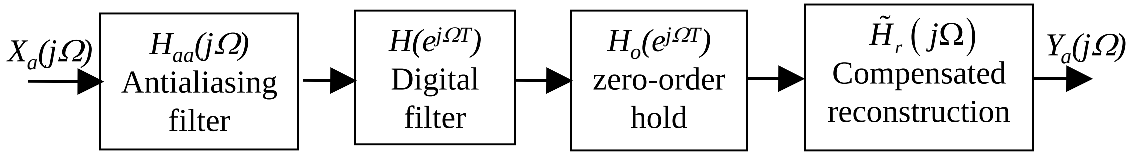
$$h_0(t)=\begin{cases}1, & 0<t<T\\ 0, & otherwise\end{cases} \Leftrightarrow H_0(j\Omega)=\frac{2\sin(\Omega T/2)}{\Omega}e^{-j\Omega T/2}$$

Thus, the compensated reconstruction filter is

$$\tilde{H}_r(j\Omega)=\begin{cases}\dfrac{\Omega T/2}{\sin(\Omega T/2)}e^{j\Omega T/2}, & |\Omega|<\pi/T\\ 0, & |\Omega|>\pi/T\end{cases}$$

Remark: A "practical filter cannot achieve this approximation

Overall System



The frequency response of the effective filter is

$$H_{eff}(j\Omega) = \tilde{H}_r(j\Omega) \cdot H_0(j\Omega) \cdot H(e^{j\Omega T}) \cdot H_{aa}(j\Omega).$$

The power spectrum density of the quantization error at the output of the compensated reconstruction filter is (from stochastic processes)

$$P_e(j\Omega) = \left| \tilde{H}_r(j\Omega) \cdot H_0(j\Omega) \cdot H(e^{j\Omega T}) \right|^2 \sigma_e^2,$$

where $\sigma_e^2 = \dfrac{\Delta^2}{12}$.