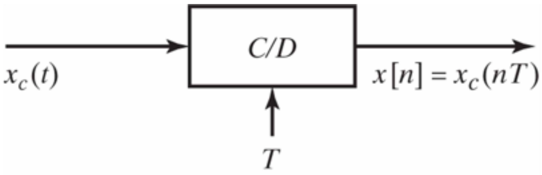
Ideal Sampling of Continuous-Time Signals

Periodic Sampling

Continous-time signals are usually sampled periodically every T seconds:

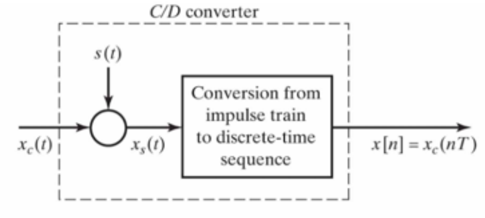
$$x[n] = x_c(nT) \qquad -\infty < n < \infty$$

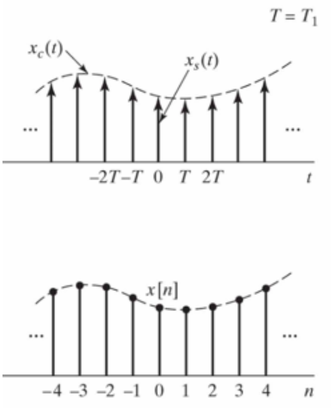


Since T is the sampling period, its reciprocal $F_s = \frac{1}{T}$ (samples/sec) is the sampling frequency, or in radians per second: $\Omega_s = \frac{2\pi}{T}$.

Sampling With a Periodic Impulse Train

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





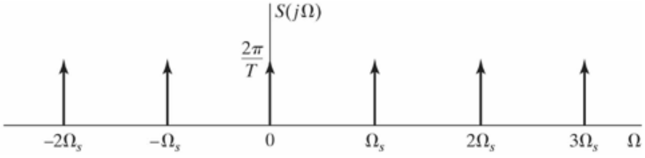
Frequency-Domain Perspective of Sampling

The impulse train that we used in the model of the sampling process,

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

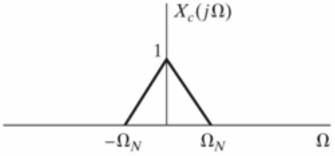
has a Fourier transform

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



Frequency-Domain Perspective of Sampling (Continued)

The continuous-time signal, $x_c(t)$, has a Fourier-transform represented by this arbitrary spectrum:



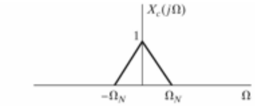
Notice this spectrum is bandlimited:

$$X_c(j\Omega) = 0 \quad \text{for } |\Omega| \geq \Omega_N$$

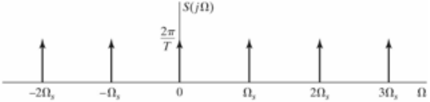
Frequency-Domain Perspective of Sampling (Continued)

In the sampling process, we multiply the impulse train, $s(t)$, the continuous-time signal, $x_c(t)$, which is equivalent to convolving their spectra:

$$X_s(j\Omega) = \frac{1}{2\pi} X_c(j\Omega) * S(j\Omega)$$

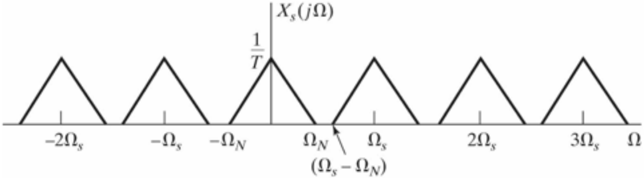


$*$



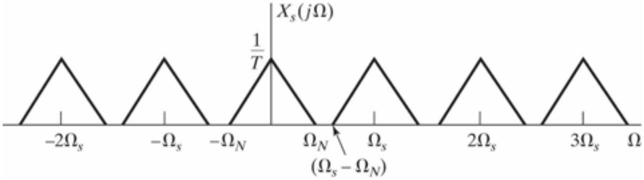
Spectrum of sampled signal:

$$X_s(j\Omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_c\left(j\left(\Omega - k\Omega_S\right)\right)$$

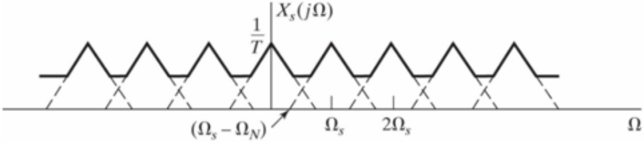


Sampling Rate to Avoid Aliasing Distortion

$\Omega_s - \Omega_N \geq \Omega_N$, or $\Omega_s \geq 2\Omega_N$, Fourier transform of sampled signal:



$\Omega_s < 2\Omega_N$, causes *aliasing*:



Nyquist-Shannon Sampling Theorem

Let $x_c(t)$ be a bandlimited signal with

$$X_c(j\Omega) = 0 \quad \text{for } |\Omega| \geq \Omega_N.$$

Then $x_c(t)$ is uniquely determined by its samples $x[n] = x_c(nT)$, $n = 0, \pm 1, \pm 2, \ldots$, if

$$\Omega_s = \frac{2\pi}{T} \geq 2\Omega_N.$$

Ω_N is the *Nyquist frequency*.
$2\Omega_N$ is the *Nyquist rate*.

Relationship Between Continuous-Time and Discrete-Time Frequency Variables with Periodic Sampling

Discrete-Time Sinusoidal Signals

$$x[n] = A\cos(\omega n + \theta) = A\cos(2\pi f n + \theta) \quad \text{where } n \text{ is an integer}$$

- A discrete-time sinusoid is periodic only if its frequency f is a rational number: $f = \frac{k}{N}$, where k and N are integers (common factors cancelled, N is period)
- Discrete-time sinusoids whose frequencies are separated by an integer multiples of 2π are identical: $\cos[(\omega + 2\pi)\,n + \theta] = \cos(\omega n + 2\pi n + \theta) = \cos(\omega n + \theta)$.
- The highest rate of oscillation in a discrete-time sinusoid is when frequency $w = \pm\pi$ (or $f = \pm 1/2$); *fundamental range*: $-\pi \leq \omega \leq \pi$ $(-\frac{1}{2} \leq f \leq \frac{1}{2})$

Relative or Normalized Frequency

Periodic sampling creates relationship between time variables t and n:

$$t = nT = \frac{n}{F_s}$$

This leads to a relationship between the continuous-time frequency F (or Ω) and the discrete-time frequency f (or ω), assuming the signal is sampled at $F_s = 1/T$ samples per second.

$$x_c(t) = A\cos\left(2\pi F t + \theta\right)$$

$$x[n] = x_c(nT) = A\cos\left(2\pi F nT + \theta\right) = A\cos\left(\frac{2\pi nF}{F_s} + \theta\right)$$

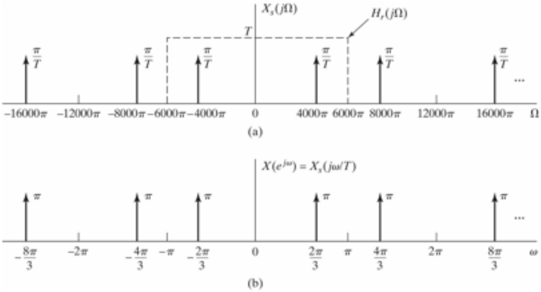
$$\boxed{f = \frac{F}{F_s}} \quad \text{or,} \quad \boxed{\omega = \Omega T}$$

Example: Sampled Sinusoidal Signal

If we sample $x_c(t) = \cos(4000\pi t)$ with a sampling period of $T = 1/6000$ we obtain:

$$x[n] = x_c(nT) = \cos\left(4000\pi T n\right) = \cos\left(\frac{2\pi}{3}n\right)$$

The sample rate is $\Omega_s = 2\pi/T = 12000\pi$ rad/sec which is more than twice the frequency of the original signal so there is no aliasing.



The DTFT is a function of the normalized frequency: $\omega = \Omega T$

Example: Sampling Two Analog Signals

Determine the discrete-time signals from sampling analog signals at $f_s = 40$ Hz:

$$x_1(t) = \cos\left(2\pi 10 t\right)$$
$$x_2(t) = \cos\left(2\pi 50 t\right)$$

replace $t = nT = \frac{n}{f_s} = \frac{n}{40}$:

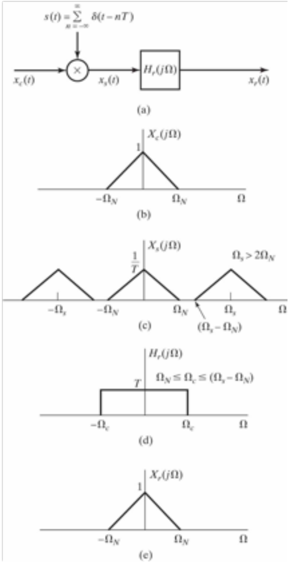
$$x_1[n] = \cos\left(2\pi \frac{10}{40} n\right) = \cos\left(\frac{\pi}{2} n\right)$$
$$x_2[n] = \cos\left(2\pi \frac{50}{40} n\right) = \cos\left(\frac{5\pi}{2} n\right)$$

However, $\cos\left(\frac{5\pi}{2} n\right) = \cos\cos\left(\frac{5\pi}{2} n - 2\pi n\right) = \cos\left(\frac{\pi}{2} n\right)$ so $x_2[n] = x_1[n]$.

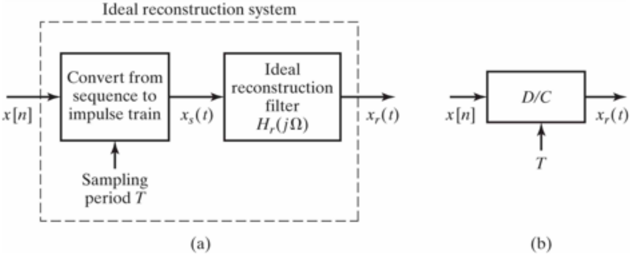
50 Hz is an *alias* of 10 Hz when the sampling frequency is 40 Hz

Reconstruction of a Signal From Its Samples

Recovery of Signal From Its Samples: Frequency Domain



Recovery of Signal From Its Samples: Time Domain



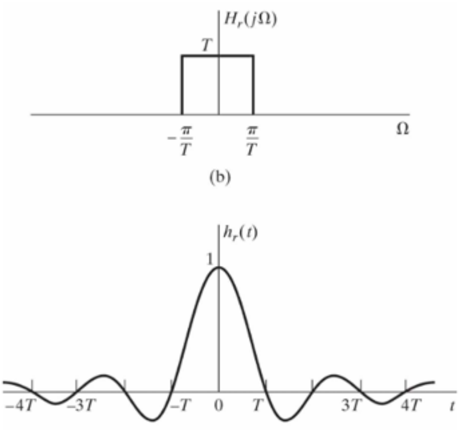
Sampled signal represented by impulse train of successive values:

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

Reconstructed signal is output of ideal lowpass filter, with impulse response, $h_r(t)$:

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

Impulse Responses of Ideal Reconstruction Filter



Cutoff frequency:

$$\Omega_c = \Omega_s/2 = \pi/T$$

Impulse response:

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

Using normalized sinc function:

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

Reconstruction of the Original Signal

Output of reconstruction filter:

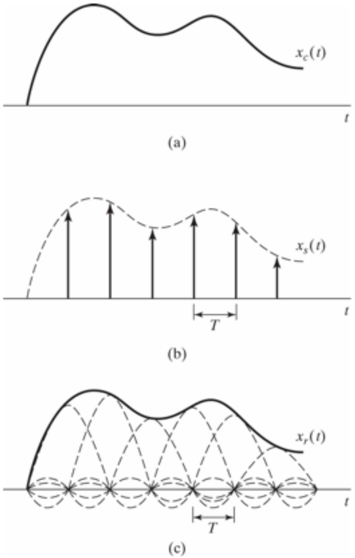
$$x_r(t) = \sum_{n=\infty}^{\infty} x[n] h_r(t - nT)$$

where impulse response is

$$h_r(t) = \mathrm{sinc}(t/T)$$

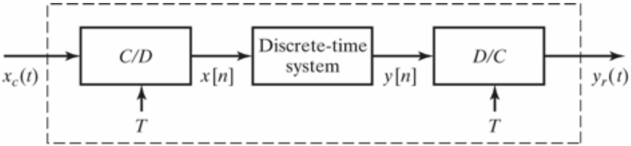
Output:

$$x_r(t) = \sum_{n=\infty}^{\infty} x[n] \, \mathrm{sinc}\left((t - nT)/T\right)$$



Discrete-Time Processing of Continuous-Time Signals

Summary of Mathematical Representations



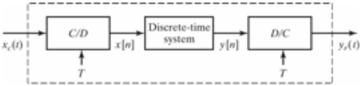
$$x[n] = x_c(nT)$$

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

$$y_r(t) = \sum_{n=-\infty}^{\infty} y[n]\,\mathrm{sinc}\left[(t-nT)/T\right]$$

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

Overall Frequency Response



For the LTI discrete-time system:

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

The Fourier-transform of the output of entire system is:

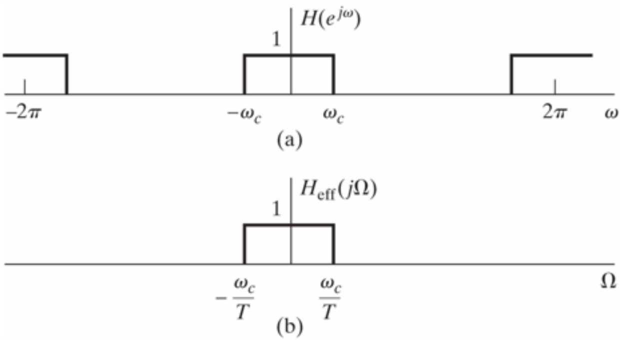
$$Y_r(j\Omega) = H_{\text{eff}}(j\Omega)X_c(j\Omega)$$

where

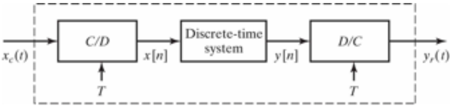
$$H_{\text{eff}}(j\Omega) = \begin{cases} H(e^{j\Omega T}), & |\Omega| < \pi/T \\ 0, & |\Omega| \geq \pi/T \end{cases}$$

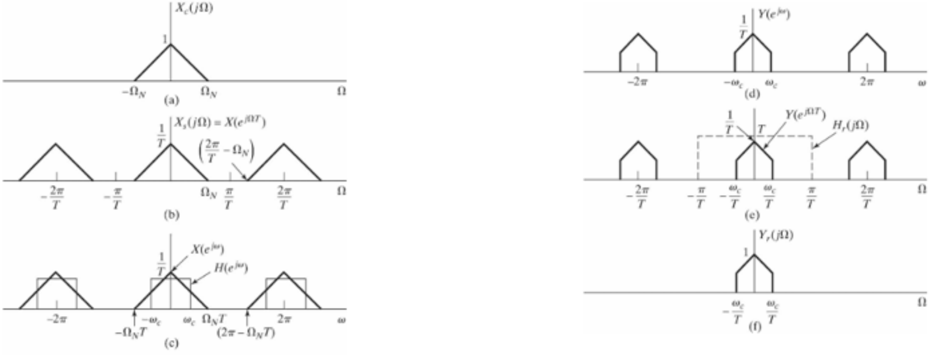
Example: Ideal Lowpass Filter

$$H(e^{j\omega}) = \begin{cases} 1, & |\omega| < \omega_c \\ 0, & \omega_c < |\omega| \leq \pi \end{cases} \qquad H_{\text{eff}}(j\Omega) = \begin{cases} 1, & |\Omega T| < \omega_c \ \text{ or } \ |\Omega| < \omega_c/T \\ 0, & |\Omega T| \geq \omega_c \ \text{ or } \ |\Omega| \geq \omega_c/T \end{cases}$$



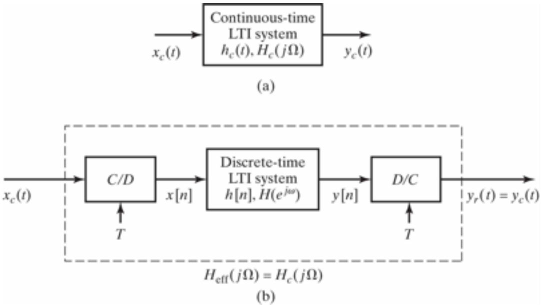
Example: Ideal Lowpass Filter (Continued)





Impulse Invariance for Implementing Continuous-Time System in Discrete-Time

Impulse Invariance (Given a Desired Continuous-Time System)



Desired: $h_c(t)$, $H_c(j\Omega)$

Choose:

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

Select T such that:

$$H_c(j\Omega) = 0, \quad |\Omega| \geq \pi/T$$

Then the *impulse-invariant* version of $h_c(t)$ is:

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

Example: Lowpass Filter Using Impulse Invariance

Use the impulse invariance technique to find an ideal lowpass discrete-time filter with cutoff frequency $\omega_c < \pi$, given this continuous-time ideal lowpass filter:

$$H_c(j\Omega) = \begin{cases} 1, & |\Omega| < \Omega_c \\ 0, & |\Omega| \geq \Omega_c \end{cases} \qquad \text{where } \Omega_c = \omega_c/T < \pi/T$$

The inverse Fourier transform is the impulse response of this system:

$$h_c(t) = \frac{\sin(\Omega_c t)}{\pi t} = \frac{\Omega_c}{\pi} \operatorname{sinc}\left(\frac{\Omega_c t}{\pi}\right)$$

Finding the impulse-invariant system $h[n] = T h_c(nT)$:

$$h[n] = T h_c(nT) = T \frac{\sin(\Omega_c n T)}{\pi n T} = \frac{\sin(\omega_c n)}{\pi n} \quad \longleftrightarrow \quad H(e^{j\omega}) = \begin{cases} 1, & |\omega| < \omega_c \\ 0, & \omega_c \leq |\omega| \leq \pi \end{cases}$$

Example: Rational System Function Using Impulse Invariance

Use the impulse invariance technique to find discrete-time system to implement:

$$h_c(t) = Ae^{s_0 t}u(t)$$

Applying the impulse invariance method: $h[n] = Th_c(nT) = ATe^{s_0 Tn}u[n]$

Finding z-transform:

$$H(z) = \frac{AT}{1 - e^{s_0 T}z^{-1}} \qquad |z| > |e^{s_0 T}|$$

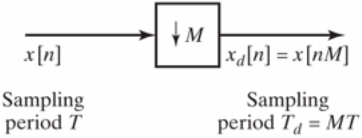
Gives frequency response, assuming $Re(s_0) < 0$:

$$H(e^{j\omega}) = \frac{AT}{1 - e^{s_0 T}e^{-j\omega}} \qquad \neq \qquad H_c(j\tfrac{\omega}{T}) = \frac{A}{j\frac{\omega}{T} - s_0}$$

(original continuous-time system did not have bandlimited frequency response)

Sample Rate Conversion: Downsampling

Sample Rate Reduction: Downsampling



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

- Reduce sample rate by integer factor M
- Same as sampling $x_c(t)$ if $T_d = MT$,
- Assume $X_c(j\Omega) = 0$ for $|\Omega| \geq \Omega_N$
- Using mapping $\Omega = \omega/T$: No aliasing if $\pi/T_d = \pi/(MT) \geq \Omega_N$
- Can reduce sampling rate by factor of M without aliasing if original sample rate is at least M times Nyquist rate (or bandwidth reduced by factor of M)

Downsampling: Frequency Domain Perspective

DTFT of $x[n]$, the original sampled signal:

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

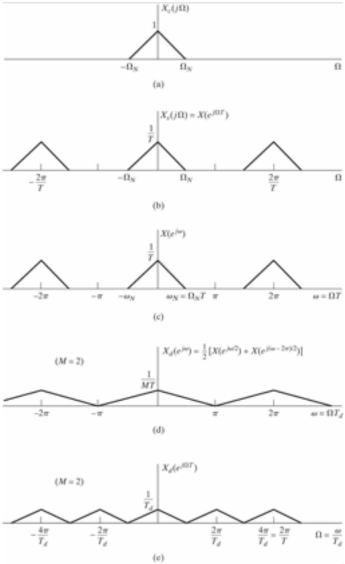
DTFT of $x_d[n] = x[nM]$, in terms of $X_c(j\Omega)$:

$$X_d(e^{j\omega}) = \frac{1}{MT}\sum_{r=-\infty}^{\infty} X_c\left[j\left(\tfrac{\omega}{MT} - \tfrac{2\pi r}{MT}\right)\right]$$

or DTFT of $x_d[n] = x[nM]$, in terms of the spectrum of $x[n]$

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

Example: Downsampling, $M = 2$, With No Aliasing



To avoid aliasing, ensure $X(e^{j\omega})$ is bandlimited:

$$X(e^{j\omega}) = 0, \quad \omega_N \leq |\omega| \leq \pi$$

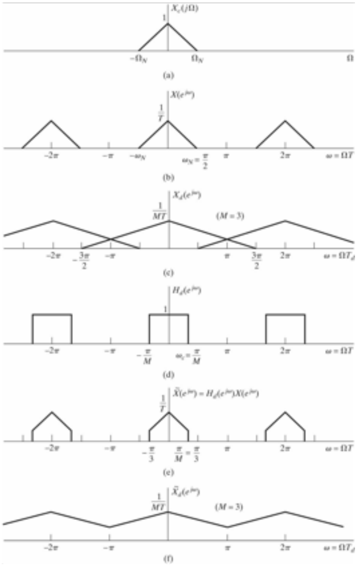
and $2\pi/M$ is above the Nyquist rate:

$$2\pi/M \geq 2\omega_N$$

or

$$\pi/M \geq \omega_N$$

Example: Downsampling, $M = 3$, With Aliasing and Prefilter

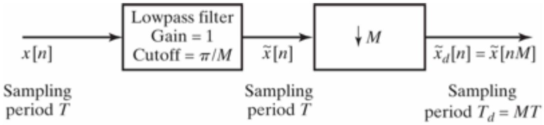


Highest frequency:
$\omega_N = \Omega_N T$

To avoid aliasing:
$\omega_M \leq \pi \quad \text{or} \quad \omega_N \leq \pi/M$

In this example: $M\omega_N = 3\pi/2 \nleq \pi$

Decimation: Lowpass filtering followed by compression

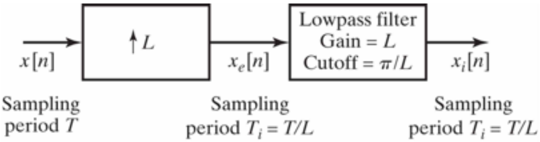


Sample Rate Conversion: Upsampling and Changing Sample Rate by a Rational Factor

Sample Rate Reduction: Upsampling

Goal is to obtain: $x_i[n] = x_c(nT_i)$

where $T_i = T/L$, from original sequence: $x[n] = x_c(nT)$



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

Upsampling: A frequency Domain Perspective

After upsampling:

$$x_e[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]$$

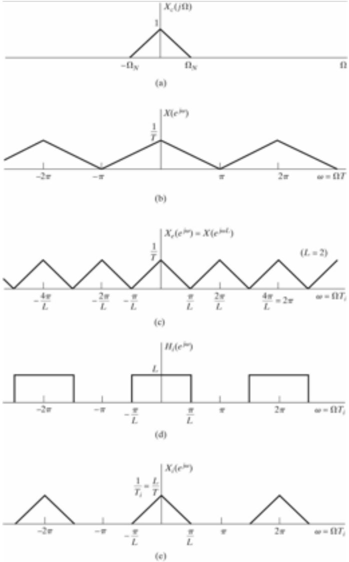
Fourier transform of upsampled sequence:

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

DTFT is frequency scaled version of input:

$$\omega \to \omega L \quad \text{so that} \quad \omega L = \Omega T \quad \text{and} \quad \omega = \Omega (T/L) = \Omega T_i$$

Example: Upsampling (Interpolation)



Upsampling: Interpolation Formula

Output of expander and input to lowpass filter:

$$x_e[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n - kL]$$

Impulse response of lowpass filter:

$$h_i[n] = \frac{\sin(\pi n/L)}{\pi n/L} \quad \text{note: } h_i[0] = 1 \text{ and } h_i[n] = 0, \;\; n = 0, \pm L, \pm 2L, ...$$

Output of lowpass filter:

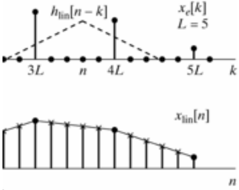
$$x_i[n] = \sum_{k=-\infty}^{\infty} x[k] \frac{\sin(\pi(n - kL)/L)}{\pi(n - kL)/L}$$

With this ideal filter: $\quad x_i[n] = x[n/L] = x_c(nT/L) = x_c(nT_i), \; n = 0, \pm L, \pm 2L, ...$

Simple and Practical Interpolation Filters

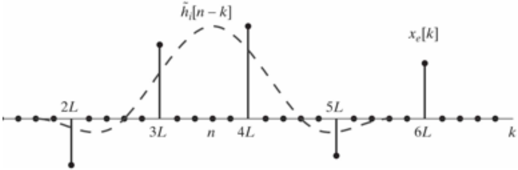
Linear Interpolation:

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

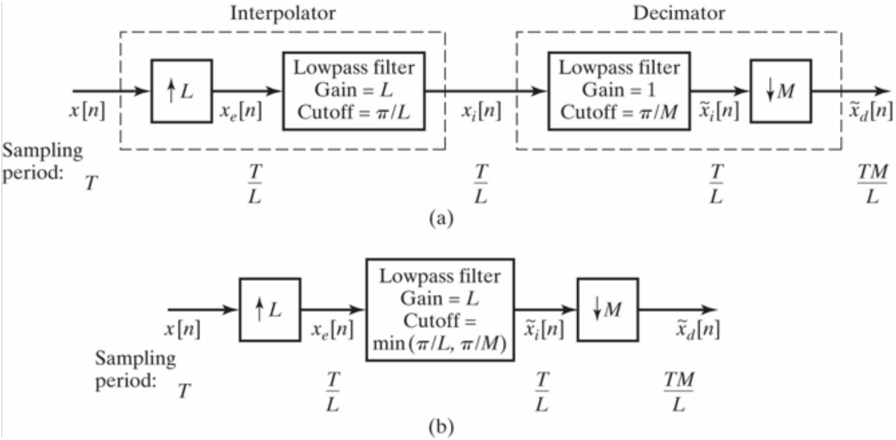


FIR Filter for interpolation, where $\tilde{h}_i[n] = 0$ for $|n| \geq KL$:

$$\tilde{x}_i[n] = \sum_{k=n-KL+1}^{n+KL-1} x_e[k]\tilde{h}_i[n-k]$$



Changing Sample Rate by Rational Factor

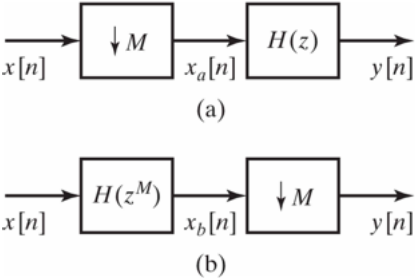


If $M > L$: increase in sample period (decrease sampling rate).

Multirate Signal Processing

Interchange of Lowpass Filter with Compressor

These two systems are equivalent:



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

From downsampling, output of second system:

$$\begin{aligned} Y(e^{j\omega}) &= \frac{1}{M}\sum_{i=0}^{M-1} X_b(e^{j(\omega/M - 2\pi i/M)}) \\ &= \frac{1}{M}\sum_{i=0}^{M-1} X(e^{j(\omega/M - 2\pi i/M)})H(e^{j(\omega - 2\pi i)}) \end{aligned}$$

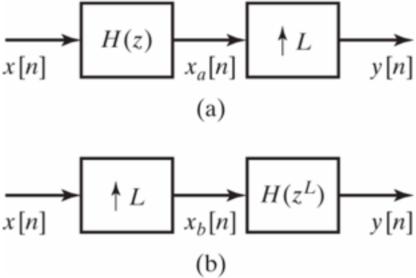
Since $H(e^{j(\omega-2\pi i)}) = H(e^{j\omega})$:

$$\begin{aligned} Y(e^{j\omega}) &= H(e^{j\omega})\frac{1}{M}\sum_{i=0}^{M-1} X(e^{j(\omega/M - 2\pi i/M)}) \\ &= H(e^{j\omega})X_a(e^{j\omega}) \end{aligned}$$

Interchange of Lowpass Filter with Expander

These two systems are equivalent:

(Linear filter modified)



Spectrum at expander output (2nd system):

$$X_b(e^{j\omega}) = X(e^{j\omega L})$$

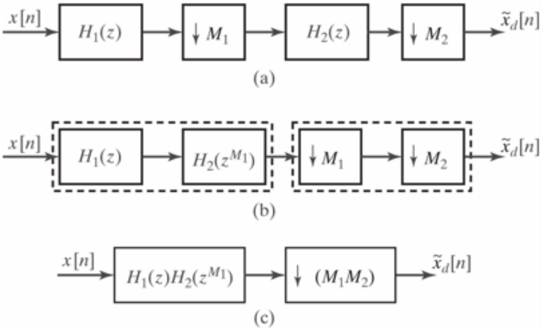
Spectrum at output (1st system):

$$\begin{aligned} Y(e^{j\omega}) &= X_a(e^{j\omega L}) \\ &= X(e^{j\omega L})H(e^{j\omega L)} \\ &= H(e^{j\omega L})X_b(e^{j\omega}) \end{aligned}$$

(which corresponds to output of second system)

Multistage Decimation To Reduce Computation

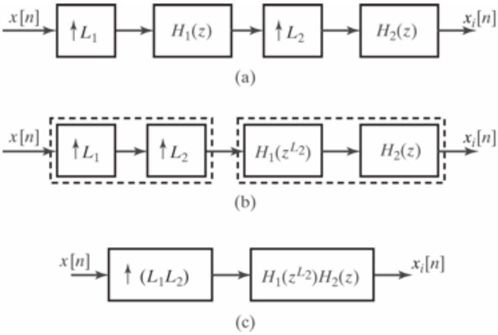
Two-stage decimation system with overall rate reduction $M = M_1 M_2$:



Equivalent single-stage lowpass filter: $H(z) = H_1(z)H_2(z^{M_1})$
Single-stage required lowpass cutoff frequency: $\pi/(M_1M_2)$
Two-stage required lowpass cutoff frequencies: π/M_1 and π/M_2

Multistage Interpolation To Reduce Computation

Two-stage interpolation system with overall sample rate increase $L = L_1 L_2$:

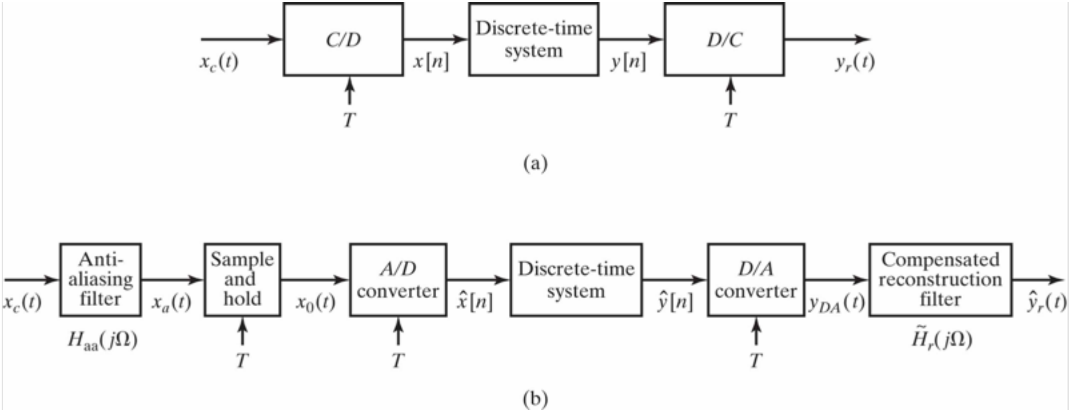


Equivalent single-stage lowpass filter:

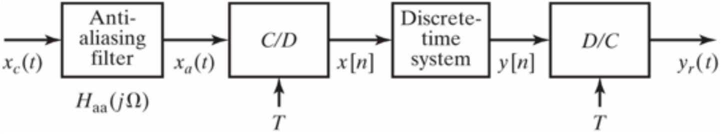
$$H(z) = H_1(z^{L_2})H_2(z)$$

Prefiltering to Avoid Aliasing

Digital Processing of Analog Signals



Anti-Aliasing Filter to Avoid Aliasing



Remember frequency scaling: $\omega = \Omega T$

Anti-aliasing filter:

$$H_{aa}(j\Omega) = \begin{cases} 1, & |\Omega| < \Omega_c \leq \pi/T \\ 0, & |\Omega| \geq \Omega_c \end{cases}$$

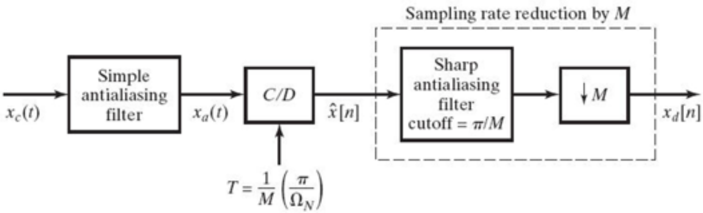
From $x_a(t)$ to $y_r(t)$:

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

Since $H_{aa}(j\Omega)$ can't be ideally bandlimited, the overall frequency response is:

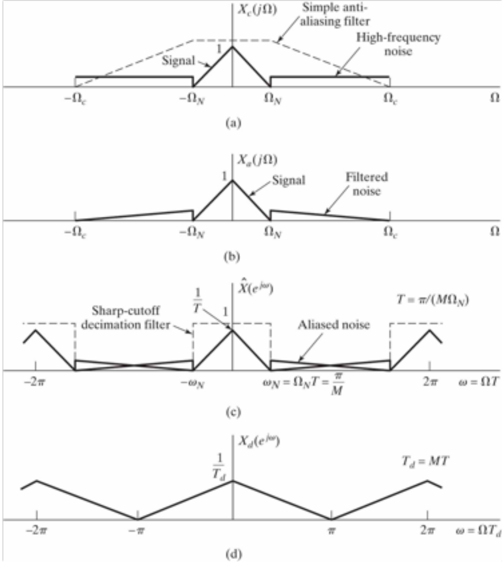
$$H_{eff}(j\Omega) \approx H_{aa}(j\Omega)H(e^{j\Omega T})$$

Oversampling to Simplify Anti-Aliasing Filter



- Simple anti-aliasing filter with gradual cutoff at Ω_N
- Significant attenuation at $M\Omega_N$
- Sample at rate higher than $2\Omega_N$ (sample at $2M\Omega_N$)
- Implement sharp antialiasing filter in discrete-time domain when downsampling by factor of M
- Subsequent discrete-time processing completed at lower sampling rate

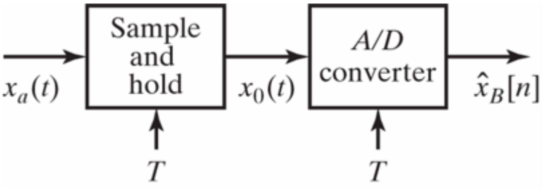
Spectrum of Oversampling Followed by Decimation

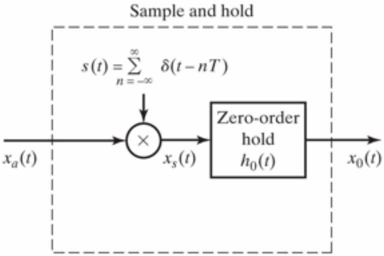


Analog-to-Digital Conversion

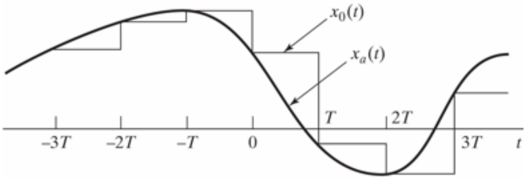
Ideal Sample-and-Hold

Analog-to-digital conversion:



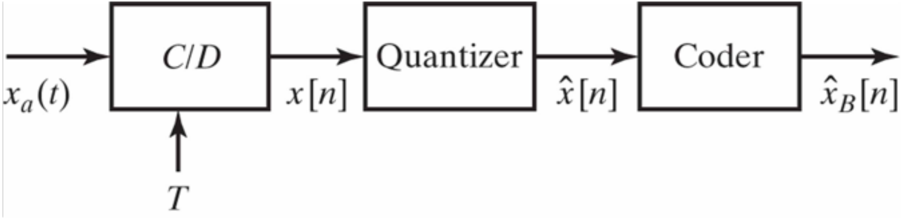


(a)



(b)

Analog-to-Digital Conversion System

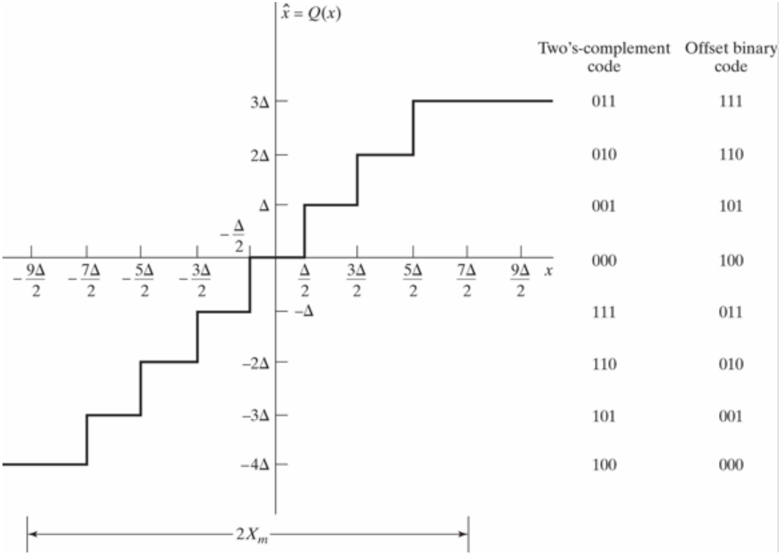


Quantizer maps input sample to finite set of values:

$$\hat{x}[n] = Q(x[n])$$

Coder will label each quantization level with binary code of $(B + 1)$-bits.

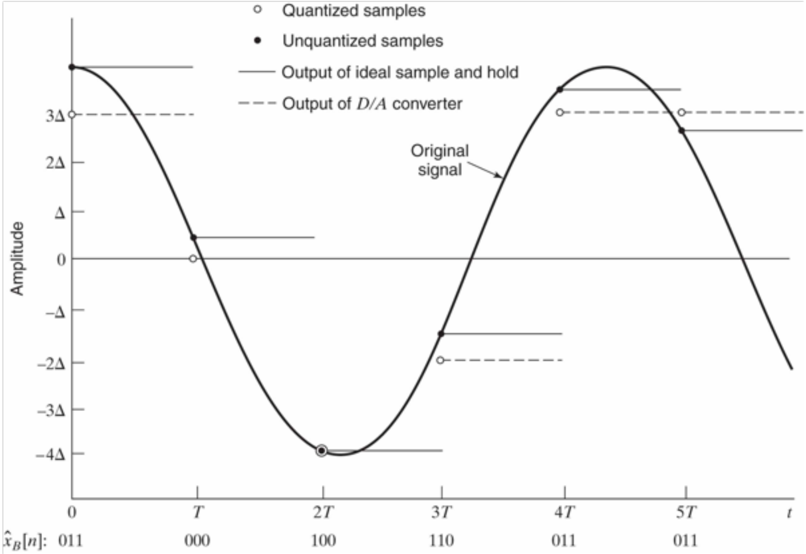
Quantizer for A/D Conversion (8 Levels with $B+1$ bits, $B=2$)



Quantizer step size:

$$\Delta = \frac{2X_m}{2^{B+1}} = \frac{X_m}{2^B}$$

Example: Sampling, Quantization, Coding, D/A Conversion (3-bits)



Analysis of Quantization Errors

Quantization Error

Quantization error is difference between quantized sample and true sample value:

$$e[n] = \hat{x}[n] - x[n]$$

If Δ is the step-size and the input samples stay within the full range of

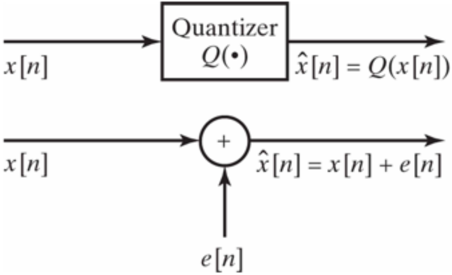
$$(-X_m - \Delta/2) < x[n] \le (X_m - \Delta/2)$$

then the quantization error will be between:

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

otherwise, then $|e[n]| > \Delta/2$ and samples are *clipped* (quantizer is *overloaded*)

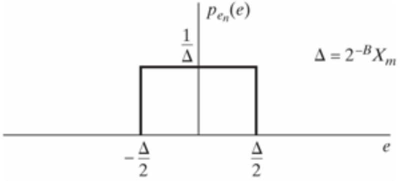
Model of Quantizer



- $e[n]$ is sample sequence of stationary random process
- $e[n]$ is uncorrelated with $x[n]$
- $e[n]$ is a white noise process
- uniform distribution over the range of quantization error

Statistical Model for Quantization Noise

Probability density function of $e[n]$, the quantization error using $(B+1)$ bits:



Variance (and quantization noise power):

$$\sigma_e^2 = \int_{-\Delta/2}^{\Delta/2} e^2 \frac{1}{\Delta} de = \frac{\Delta^2}{12} = \frac{2^{-2B}X_m^2}{12} \qquad \text{since } \Delta = \frac{X_m}{2^B}$$

Since autocorrelation is $\phi_{ee}[m] = \sigma_e^2 \delta[m]$, the power spectral density is:

$$P_{ee}(e^{j\omega}) = \sigma_e^2 = \frac{2^{-2B}X_m^2}{12} \qquad |\omega| \leq \pi$$

Signal-to-Quantization-Noise Ratio (SNR)

SNR for $(B+1)$-bit uniform quantizer is

$$\begin{aligned} SNR_Q &= 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right) = 10\log_{10}\left(\frac{12\cdot 2^{2B}\sigma_x^2}{X_m^2}\right) \\ &= 6.02B + 10.8 - 20\log_{10}\left(\frac{X_m}{\sigma_x}\right) \end{aligned}$$

- SNR increases about 6 dB for each bit added (doubling of quantization levels)
- To prevent last term from becoming large and negative, signal amplitude should be matched to full-scale amplitude of A/D