CHAPTER 7

- if $x(t)$ is periodic with period T, so that

$$x(t+T) = x(t) \quad \forall\, t \in \mathbb{R},$$

- Then $x(t)$ can be written as a Fourier series according to

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_s t}$$

where $\omega_s = \frac{2\pi}{T}$

and where the Fourier series coefficients a_k are given by

$$a_k = \frac{1}{T}\int_{-T/2}^{T/2} x(t)\, e^{-jk\omega_s t}\, dt$$

- The Fourier transform of $x(t)$ is given by

$$X(\omega) = \mathcal{F}\left[\sum_{k=-\infty}^{\infty} a_k e^{jk\omega_s t}\right]$$

$$= \sum_{k=-\infty}^{\infty} a_k \mathcal{F}\left[e^{jk\omega_s t}\right]$$

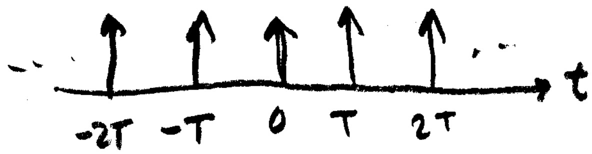
$$= 2\pi \sum_{k=-\infty}^{\infty} a_k \delta(\omega - k\omega_s)$$

→ A train of equidistant weighted impulses.

⇒ we see that the F.T. of a periodic signal is pure impulses.

- Now consider the periodic signal

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



- write as a Furier Series: $\omega_s = \frac{2\pi}{T}$

$$a_k = \frac{1}{T}\int_{-T/2}^{T/2} p(t)\, e^{-jk\omega_s t}\, dt$$

$$= \frac{1}{T}\int_{-T/2}^{T/2} \underbrace{\sum_{n=-\infty}^{\infty} \delta(t+nT)}_{} e^{-jk\omega_s t}\, dt$$

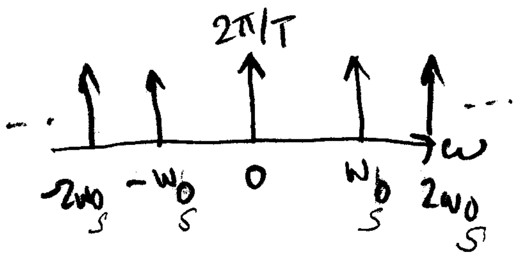
only the n=0 term is non zero in the region of integration $[-T/2, T/2]$

$$= \frac{1}{T}\int_{-T/2}^{T/2} \delta(t)\, e^{-jk\omega_s t}\, dt = \frac{1}{T}\cdot 1 = \frac{1}{T}$$

(for all k).

So $p(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_s t} = \frac{1}{T}\sum_{k=-\infty}^{\infty} e^{jk\omega_s t}$

And $P(\omega) = \frac{2\pi}{T}\sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_s) = \frac{2\pi}{T}\sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_s)$



where $\omega_s = \frac{2\pi}{T}$.

SAMPLING

- Sampling is a way to convert a continuous-time signal into a discrete-time signal.

- This is done by setting the discrete time signal $x[n]$ equal to uniformly spaced samples of the continuous-time signal $x(t)$.

→ An "analog to digital converter", or ADC (also called an "A to D" or an "A/D" converter) is a device that can do this.

- Intuitively, if we sample $x(t)$ "fast enough", then no information will be lost...

  → That is, in theory we could recover $x(t)$ from $x[n]$.

- Why do this?

  Here are some reasons:

  1. You are asked to design a filter to remove noise from an analog signal.

     Suppose you design the circuit to do it.

     → If the noise changes at some later time, you have to design a new circuit.

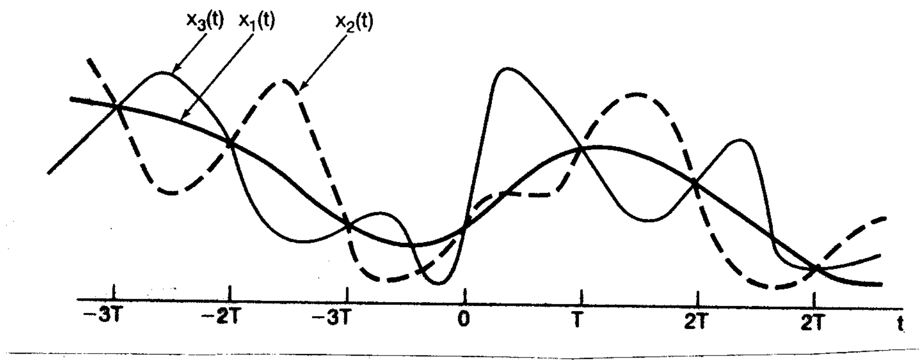
     → If you had sampled the signal and used a digital filter instead, the change would require new software, but not new hardware.

  2. If the analog signal $x(t)$ is represented by samples $x[n]$, then it does not degrade over time. Also, it does not degrade when you copy it.

     EX: Audio Compact Disk.

- What is meant by sampling "fast enough"?

→ Consider the three signals $x_1(t)$, $x_2(t)$, and $x_3(t)$ below:



- If we generate $x[n]$ by taking a sample every T seconds, then from $x[n]$ we can't tell which signal we started with.

→ Information is lost during sampling in this case.

→ Clearly, sampling every T seconds is not fast enough for these signals.

- When we sample a continuous-time signal $x(t)$ every T seconds, we get a discrete-time signal

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

→ So $x[0] = x(0)$, $x[1] = x(T)$, $x[-1] = x(-T)$

$x[2] = x(2T)$, $x[-2] = x(-2T)$ ...

- To answer the question of how fast is "fast enough" to sample, we need to represent the sampling mathematically.

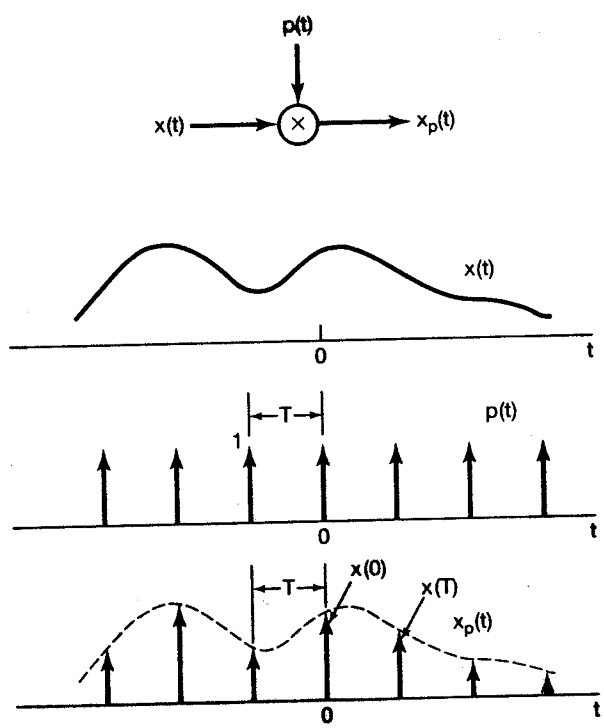
- To do this, we will use a periodic impulse-train $p(t)$ called a "sampling function":

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

- "T" is called the "sampling period", or "sampling interval".
- The sampling frequency is $\omega_s = \frac{2\pi}{T}$ rad/sec.
- The sampled signal is $x_p(t) = x(t)p(t)$.

$\Rightarrow$ Since $x(t)\delta(t-nT) = x(nT)\delta(t-nT)$, we can get the values for $x[n]$ by "picking off" the weights of the impulses in $x_p(t)$:

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

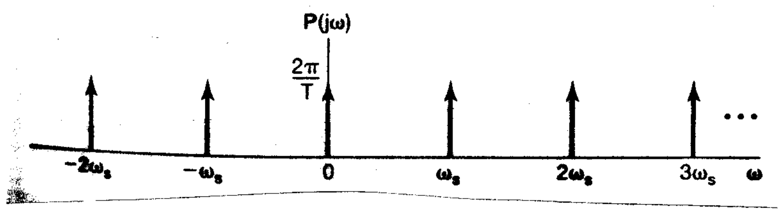


- If we can get back $x(t)$ from $x_p(t)$, then the sampling is "fast enough", and no information is lost.

  → In other words, $x[n]$ contains all of the information that was in $x(t)$.

- As we saw last time, the Fourier transform of the impulse train $p(t)$ is itself an impulse train in frequency:

$$P(\omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_s)$$



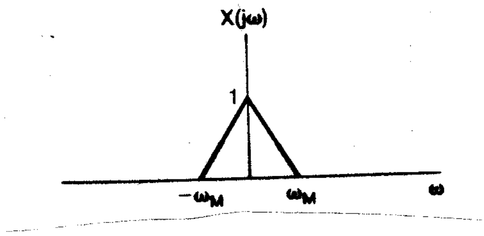
- Using the frequency convolution property of the Fourier transform, we see that

$$X_p(\omega) = \frac{1}{2\pi} X(\omega) * P(\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s)$$

- So $X_p(\omega)$ contains periodic repetitions of $X(\omega)$, centered at integer multiples of ω_s.

- Since $\omega_s = \frac{2\pi}{T}$, sampling faster makes these periodic repetitions further apart.

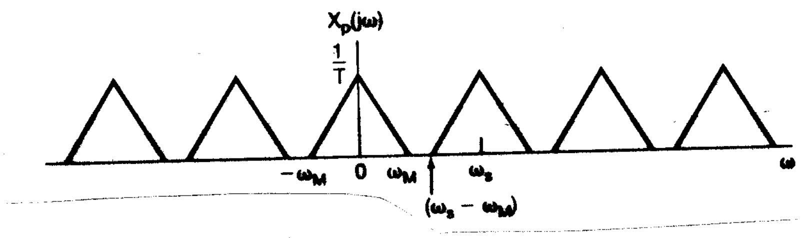
EX:



In this case, $x(t)$ is bandlimited, so that

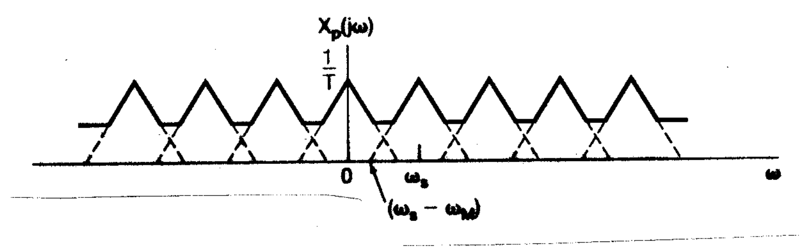
$$X(\omega)=0,\ |\omega|>\omega_M.$$

→ If we sample "fast", then the periodic repetitions of $X(\omega)$ do not overlap:



→ Then we can recover $x(t)$ from $x_p(t)$ using a low-pass filter.

→ If we sample too slowly, then the periodic repetitions of $X(\omega)$ overlap and we cannot recover $x(t)$ from $x_p(t)$:



→ This is called "aliasing".

- From the figures on page 7.5, we can see how to prevent aliasing.

- we need to have $\omega_M < \omega_s - \omega_M$

→ In other words, $\underline{\underline{\omega_s > 2\omega_M}}$. ☆☆

⇒ So we need $T < \frac{\pi}{\omega_M}$

→ In other words, the higher the frequency content of the signal $x(t)$, the faster we have to sample to prevent aliasing (information loss).

Sampling Theorem:
Let $x(t)$ be a band-limited signal with $X(j\omega) = 0$ for $|\omega| > \omega_M$. Then $x(t)$ is uniquely determined by its samples $x(nT)$, $n = 0, \pm 1, \pm 2, \ldots$, if

$$\omega_s > 2\omega_M,$$

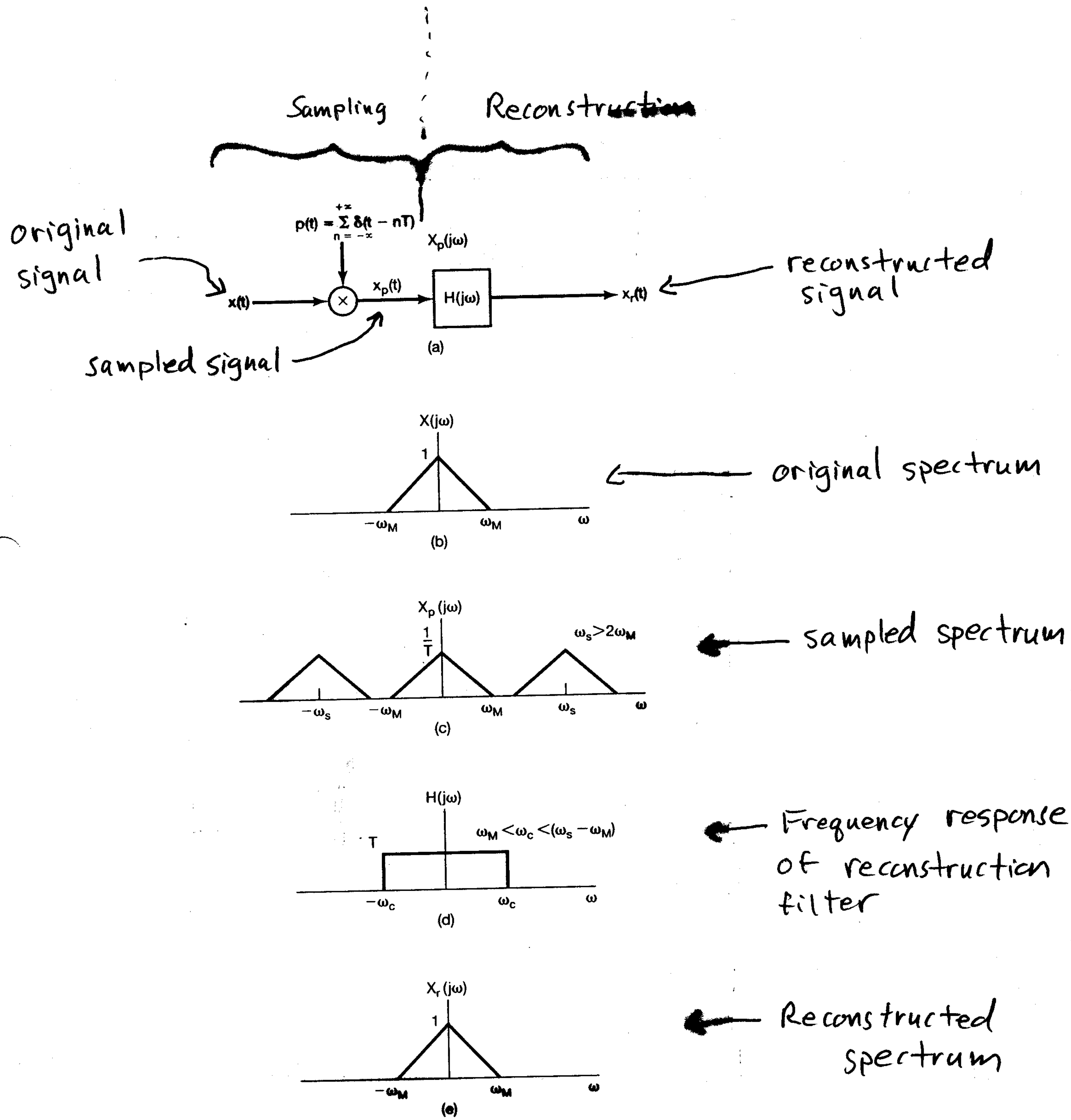
where

$$\omega_s = \frac{2\pi}{T}.$$

Given these samples, we can reconstruct $x(t)$ by generating a periodic impulse train in which successive impulses have amplitudes that are successive sample values. This impulse train is then processed through an ideal lowpass filter with gain T and cutoff frequency greater than ω_M and less than $\omega_s - \omega_M$. The resulting output signal will exactly equal $x(t)$.

- $2\omega_M$, the minimum sampling frequency to prevent aliasing, is called the "Nyquist rate", or the "Nyquist frequency" for the signal $x(t)$.

Reconstruction



- Since the reconstruction filter is an ideal low-pass filter, the impulse response is

$$h(t) = \frac{\omega_c T \sin \omega_c t}{\pi \omega_c t} = \frac{T \sin \omega_c t}{\pi t}$$

- The reconstructed signal is

$$x_r(t) = x_p(t) * h(t)$$

$$= \left[\sum_{n=-\infty}^{\infty} x(nT)\delta(t-nT) \right] * h(t)$$

$$= \sum_{n=-\infty}^{\infty} x(nT) h(t-nT)$$

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

$\Rightarrow$ If $\omega_s > 2\omega_M$ and $\omega_M < \omega_c < \omega_s - \omega_M$,

then $x_r(t) = x(t)$.

<u>NOTE</u>: in practice, a real impulse-train can not be generated. Instead, a device called a "zero-order hold" is used to implement the A-D converter.

$\Rightarrow$ This requires some modifications to the reconstruction filter. See section 7.1.2 of the book for the details.

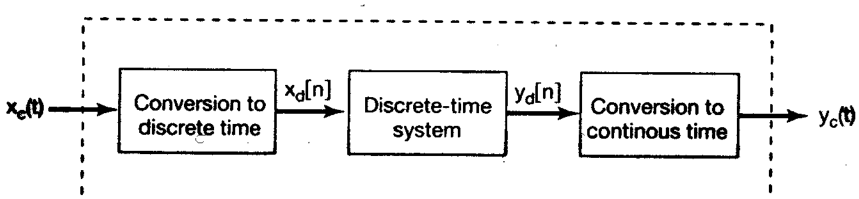
NOTE: Many real-world signals are not band limited. To prevent aliasing, a low-pass filter must be applied prior to A-D conversion to bandlimit the signal.

→ Such a filter is called an "anti-aliasing filter".

NOTE: The reconstruction filter performs digital to analog conversion. It is called a "D to A" converter, a "DAC", or a "D/A".

Discrete-Time Processing of Continuous-Time Signals

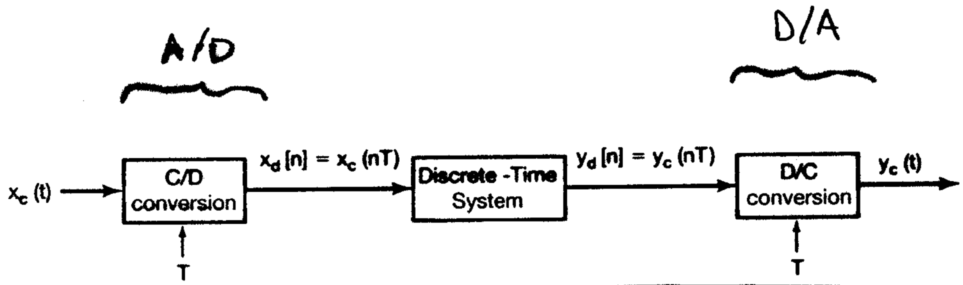
- Today, many continuous-time systems are implemented using A-D conversion, a digital signal processor (DSP), and D-A conversion.
- This gives us flexibility, because the system properties can be changed by simply writing new software, without the need to change any hardware.



$x_c(t)$: continuous-time input signal
$y_c(t)$: continuous-time output signal
$x_d[n]$: discrete-time input signal
$y_d[n]$: discrete-time output signal

$$x_d[n] = x_c(nT)$$
$$y_d[n] = y_c(nT)$$

- More specifically,



- The relationship between $x_d[n]$ and $x_c(t)$ obviously depends on the sampling rate $\omega_s = \frac{2\pi}{T}$:

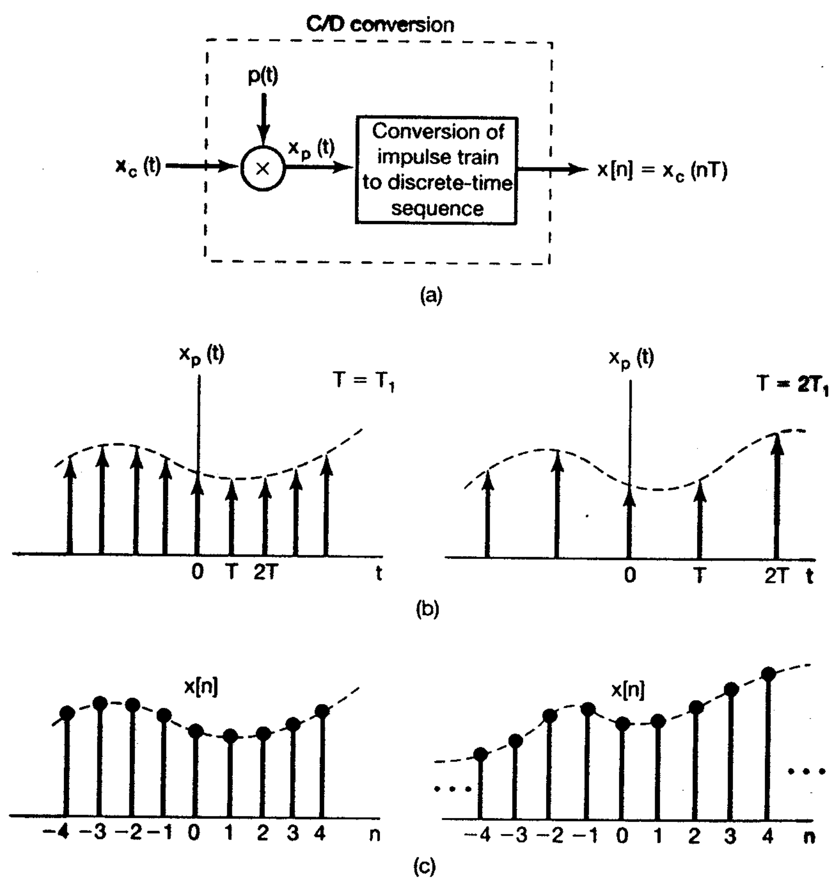


Figure 7.21 Sampling with a periodic impulse train followed by conversion to a discrete-time sequence: (a) overall system; (b) $x_p(t)$ for two sampling rates. The dashed envelope represents $x_c(t)$; (c) the output sequence for the two different sampling rates.

- To study the discrete-time processing of continuous-time signals, we must look at what happens in the frequency domain.

- Thus, we will need to distinguish between discrete-time or "digital" frequencies and continuous-time or "analog" frequencies.

- The book uses "ω" for continuous-time frequency:

$$x_c(t) \overset{\mathcal{F}}{\longleftrightarrow} X_c(\omega) \qquad y_c(t) \overset{\mathcal{F}}{\longleftrightarrow} Y_c(\omega)$$

and "Ω" for discrete-time frequency:

$$x_d[n] \overset{\mathcal{F}}{\longleftrightarrow} X_d(e^{j\Omega}) \qquad y_d[n] \overset{\mathcal{F}}{\longleftrightarrow} Y_d(e^{j\Omega})$$

<u>NOTE</u>: This is backwards from the convention most often used in the literature. Most authors use "ω" for discrete frequency and "Ω" for continuous-time frequency.

- Recall: $x_p(t) = \sum_{n=-\infty}^{\infty} x_c(nT)\,\delta(t-nT)$

- Since $\delta(t-nT) \overset{\mathcal{F}}{\longleftrightarrow} e^{-j\omega nT}$, we have

$$X_p(\omega) = \sum_{n=-\infty}^{\infty} x_c(nT)\, e^{-j\omega nT} \qquad (*)$$

$\longrightarrow$

- By definition, the Fourier transform of $x_d[n]$ is

$$X_d(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x_d[n] e^{-j\Omega n}$$

→ since $x_d[n] = x_c(nT)$,

$$X_d(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x_c(nT) e^{-j\Omega n} \qquad (**)$$

$$= \sum_{n=-\infty}^{\infty} x_c(nT) e^{-j\omega T n} \Big|_{\omega = \frac{\Omega}{T}} \checkmark$$

- Comparing $(**)$ to $(*)$ on page 7.11, we see that

$$X_d(e^{j\Omega}) = X_p\left(\frac{\Omega}{T}\right). \qquad \left\{ \begin{array}{l} \text{"}\omega = \frac{\Omega}{T}\text{"} \\ \text{"}\Omega = \omega T\text{"} \end{array} \right.$$

- Recall that $X_p(\omega)$ contains periodic repetitions of $X_c(\omega)$.

- So $X_d(e^{j\Omega})$ is also periodic, as it must be since it is a discrete-time Fourier transform.

⇒ The period of $X_p(\omega)$ is $\omega_s = \frac{2\pi}{T}$

⇒ The period of $X_d(e^{j\Omega})$ is 2π, like all discrete-time Fourier transforms.

→ We see once again that the sampling frequency $\omega_s = \frac{2\pi}{T}$ determines the relationship between $X_c(\omega)$, $X_p(\omega)$, and $X_d(e^{j\Omega})$.

- In fact, as we saw on page 7.4,

$$X_p(\omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c(\omega - k\omega_s)$$

- So,

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

Faster Sampling

Slower Sampling, closer to aliasing

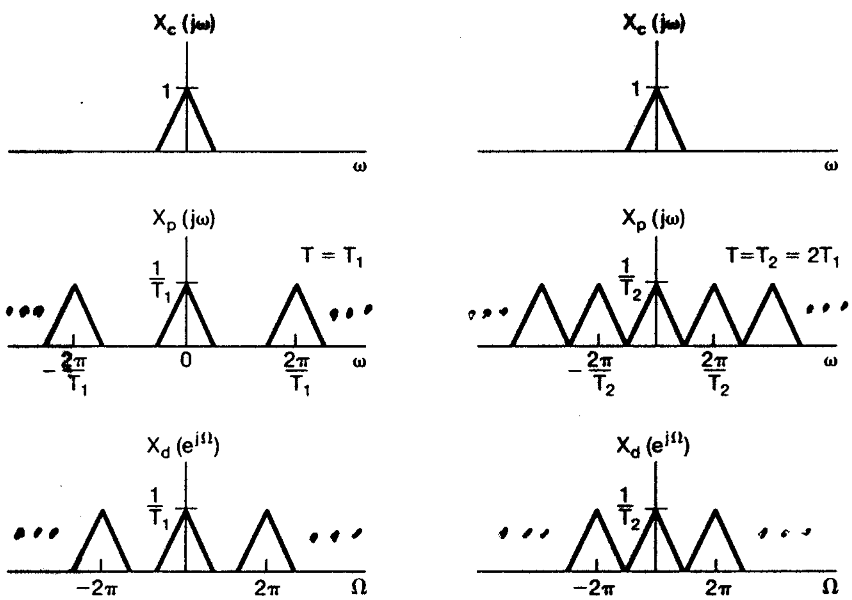


Figure 7.22 Relationship between $X_c(j\omega)$, $X_p(j\omega)$, and $X_d(e^{j\Omega})$ for two different sampling rates.

- So the overall strategy to implement a continuous-time system $H_c(\omega)$ is:

1. Convert input signal $x_c(t)$ to an impulse train $x_p(t)$.
2. Convert $x_p(t)$ to a discrete-time signal $x_d[n]$.
3. Process $x_d[n]$ with a discrete-time system $H_d(e^{j\Omega})$ implemented with a computer or DSP to get $y_d[n]$.
4. Convert $y_d[n]$ to an impulse train $y_p(t)$.
5. Use a low-pass reconstruction filter to convert $y_p(t)$ into the desired continuous-time output signal $y_c(t)$.

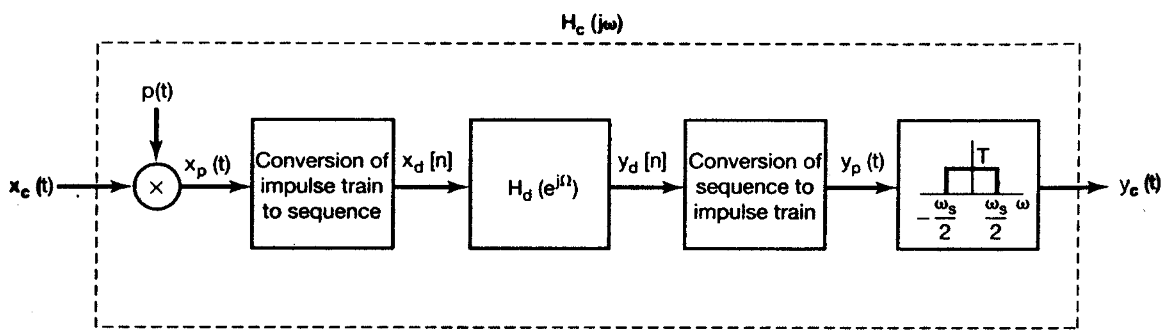


Figure 7.24 Overall system for filtering a continuous-time signal using a discrete-time filter.

- The overall system is a continuous-time system.
- The overall frequency response is $H_c(\omega)$.
- This is implemented using a discrete-time system with frequency response $H_d(e^{j\Omega})$.

- In the frequency domain,

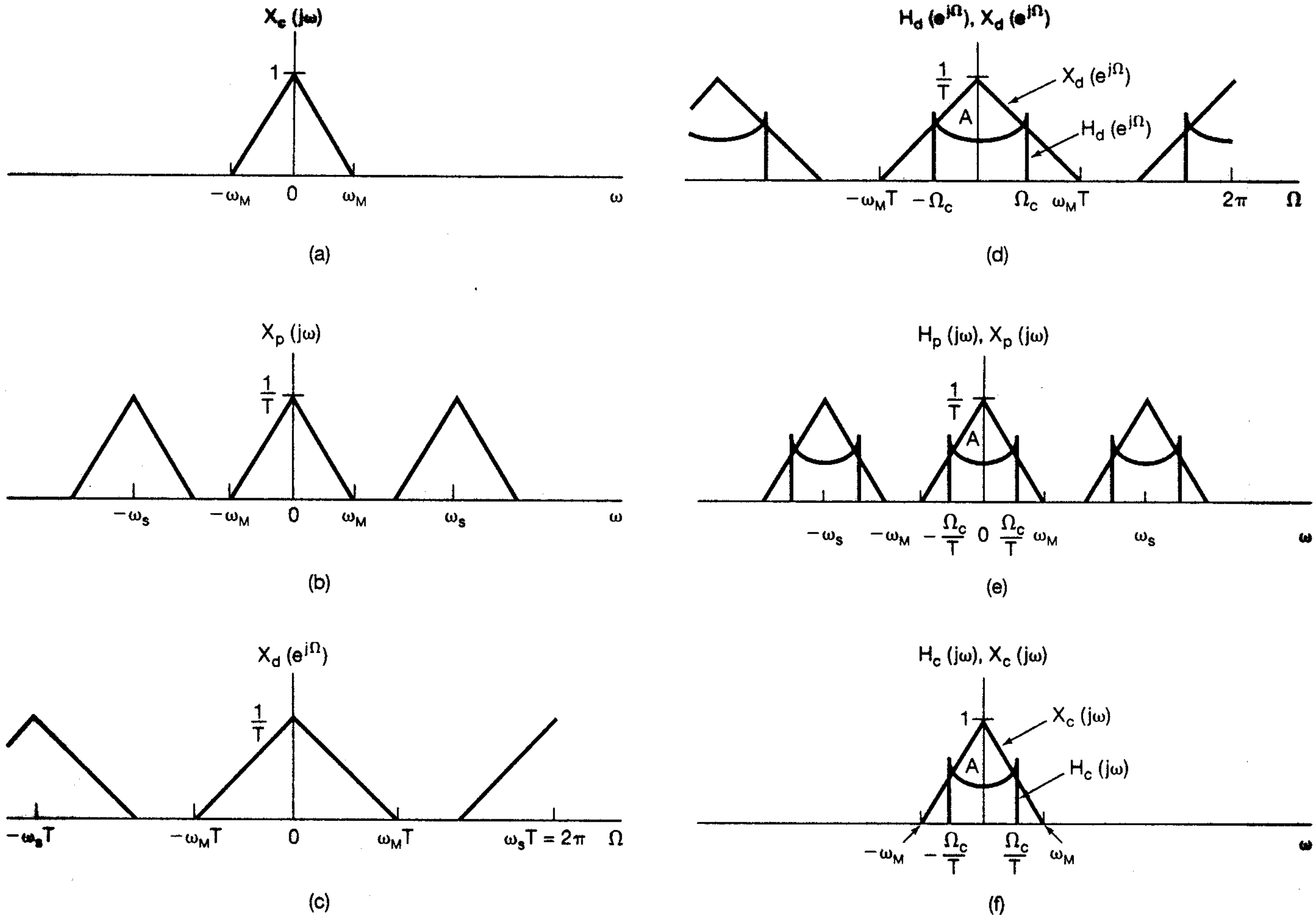


Figure 7.25 Frequency-domain illustration of the system of Figure 7.24: (a) continuous-time spectrum $X_c(j\omega)$; (b) spectrum after impulse-train sampling; (c) spectrum of discrete-time sequence $x_d[n]$; (d) $H_d(e^{j\Omega})$ and $X_d(e^{j\Omega})$ that are multiplied to form $Y_d(e^{j\Omega})$; (e) spectra that are multiplied to form $Y_p(j\omega)$; (f) spectra that are multiplied to form $Y_c(j\omega)$.

- what is the relationship between $H_c(\omega)$ and $H_d(e^{j\Omega})$?

- For the overall system, $Y_c(\omega) = X_c(\omega) H_c(\omega)$ (*)

- For the discrete-time system, $Y_d(e^{j\Omega}) = X_d(e^{j\Omega}) H_d(e^{j\Omega})$.

→ writing ωT instead of Ω, we have (recall: $\Omega = \omega T$)

$$Y_d(e^{j\omega T}) = X_d(e^{j\omega T}) H_d(e^{j\omega T}) \qquad (**)$$

→ But $X_d(e^{j\Omega}) = X_p(\frac{\Omega}{T})$ and $Y_d(e^{j\Omega}) = Y_p(\frac{\Omega}{T})$, as we saw on page 7.12.

→ So $X_d(e^{j\omega T}) = X_p(\omega)$ and $Y_d(e^{j\omega T}) = Y_p(\omega)$.

→ Thus, plugging into (**), we have

$$Y_p(\omega) = X_p(\omega) H_d(e^{j\omega T}) . \qquad (***)$$

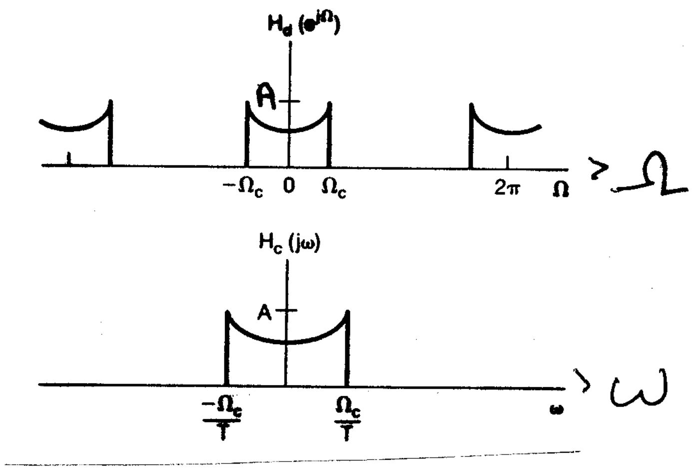
→ Now, $X_c(\omega)$ and $Y_c(\omega)$ are just the fundamental periods of $X_p(\omega)$ and $Y_p(\omega)$. Plugging into (***), we have

$$Y_c(\omega) = X_c(\omega) H_d(e^{j\omega T}) .$$ → find period

→ Comparing this to (*) above, we see that

$$H_c(\omega) = \text{fundamental period of } H_d(e^{j\omega T})$$
$$= \begin{cases} H_d(e^{j\omega T}) , & |\omega| < \frac{\omega_s}{2} \\ 0 , & |\omega| > \frac{\omega_s}{2} . \end{cases}$$

- Thus, the equivalent "analog" frequency response is equal to the fundamental period of the "digital" frequency response up to a scaling of the frequency axis by $\frac{1}{T}$:



- Given a desired continuous-time frequency response $H_c(\omega)$, we design the discrete-time system by setting

$$H_d(e^{j\Omega}) = H_c\left(\frac{\Omega}{T}\right) \qquad \left(\text{fundamental period}\right)$$

→ or, considering the periodicity of $H_d(e^{j\Omega})$,

$$H_d(e^{j\Omega}) = \sum_{k=-\infty}^{\infty} H_c\left(\frac{\Omega - 2\pi k}{T}\right)$$

<u>NOTE:</u> Multiplying $x_c(t)$ by $p(t)$ to get $x_p(t)$ is not a linear shift invariant operation. However, <u>provided</u> that the sampling theorem is satisfied, a discrete-time LSI system $H_d(e^{j\Omega})$ can be used to implement a continuous-time LSI system $H_c(\omega)$.

EX: digital differentiator:

- Recall: if $x(t) \xleftrightarrow{\mathcal{F}} X(\omega)$,

  then $\frac{d}{dt} x(t) \xleftrightarrow{\mathcal{F}} j\omega X(\omega)$.

- So, a system with frequency response $H(\omega) = j\omega$ is a differentiator:

$$x_c(t) \longrightarrow \boxed{H(\omega) = j\omega} \longrightarrow y_c(t) = \frac{d}{dt} x_c(t)$$

- If the input signals $x_c(t)$ are bandlimited, so that $X_c(\omega) = 0$, $|\omega| \geq \omega_c$, then this can be implemented with a bandlimited frequency response

$$H_c(\omega) = \begin{cases} j\omega, & |\omega| < \omega_c \\ 0, & |\omega| > \omega_c. \end{cases}$$

- To implement this system digitally, we must sample at a frequency $\omega_s = 2\omega_c$ or greater.

- The fundamental period of the equivalent discrete-time frequency response is

$$H_d(e^{j\Omega}) = j\frac{\Omega}{T}, \quad |\Omega| < \pi.$$

$H_c(\omega)$

$H_d(e^{j\Omega})$