ELEC 221 Lecture 18
CT/DT conversion and sampling DT signals

Thursday 14 November 2024

- Quiz 8 on Tuesday (L17 and L18)
- Assignment 4 due Saturday 23 Nov at 23:59 (do 4.2, 4.3, 4.4 after today; can try 4.5)
- Tutorial assignment 4 in Monday's tutorial (image processing)

We modeled **sampling** of CT signals as multiplication of a (band-limited) signal with a periodic impulse train:

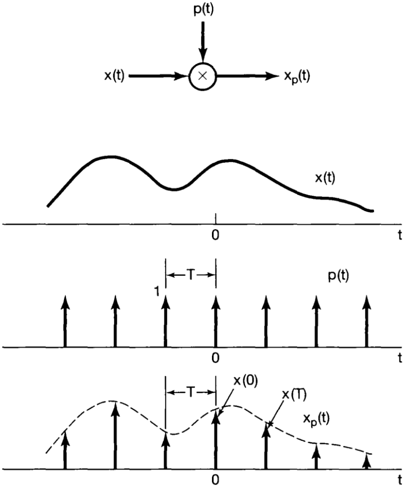


Image credit: Oppenheim 7.1

We went to the frequency domain to get a better understanding:

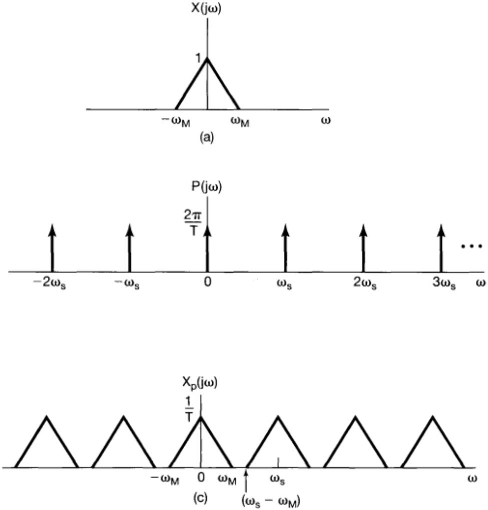


Image credit: Oppenheim 7.1

Last time

We recovered the original signal by applying a low pass filter

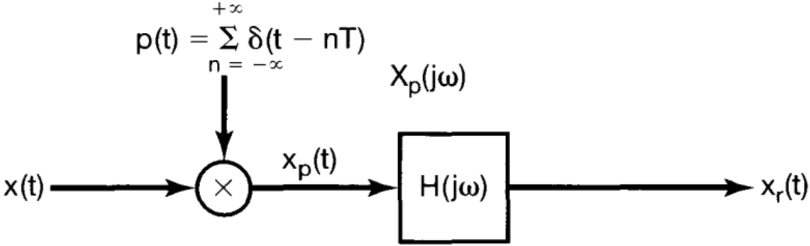


Image credit: Oppenheim 7.1

Last time

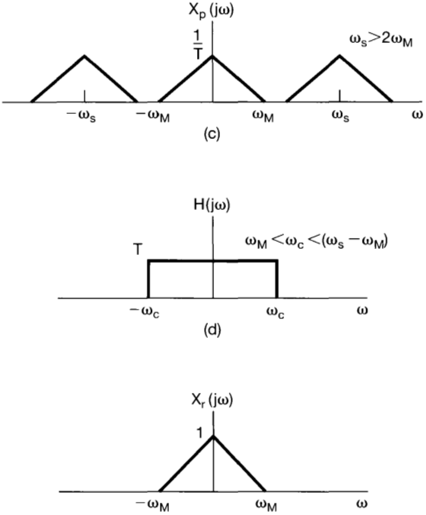


Image credit: Oppenheim 7.1

This only works if the sampling rate is higher than the **Nyquist rate**, i.e., $\omega_s > 2\omega_m$

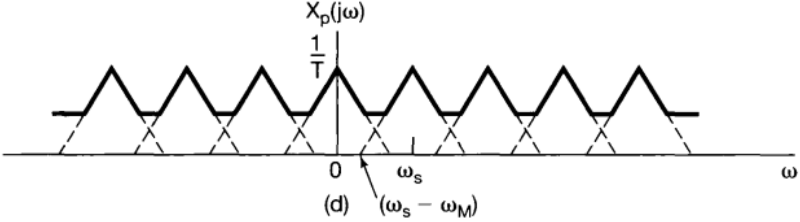
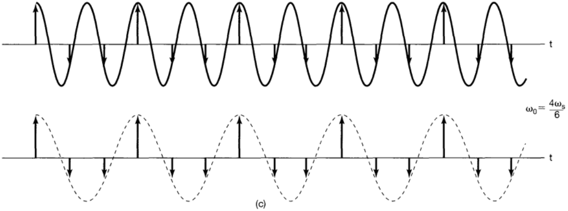


Image credit: Oppenheim 7.1

Last time

If the frequency isn't high enough, **aliasing** occurs.



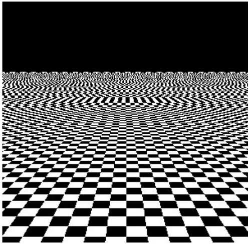


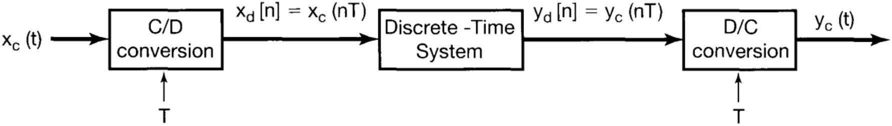
Image credit: Oppenheim 7.3, `https://texturingraphics.wordpress.com/what-is-texture-mapping/anti-aliasing-problem-and-mipmapping/`

Today

Learning outcomes:

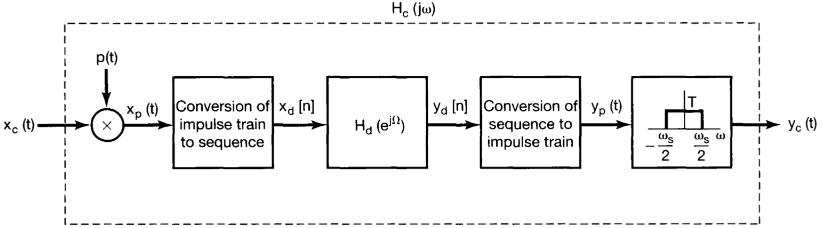
- describe the frequency domain effects of sampling from a discrete signal
- decimate and interpolate a DT signal
- determine how decimation and interpolation affect the spectrum of a DT signal

Often convenient to process CT signals by first converting to DT, processing, then converting back.



What is the theory that makes this possible?

Image credit: Oppenheim 7.20



Let's explore what happens at the level of the spectra

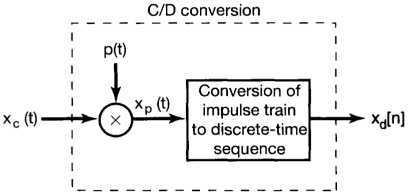
Note: we have *two frequencies*, one in CT, one in DT. Write:

$$X_c(j\omega), \quad Y_c(j\omega)$$

$$X_d(e^{j\Omega}), \quad Y_d(e^{j\Omega})$$

Image credit: Oppenheim 7.24

First: how are $X_p(j\omega)$ and $X_d(e^{j\Omega})$ related?



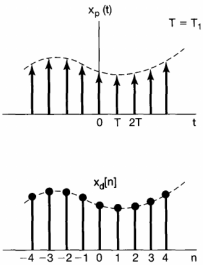


Image credit: Oppenheim 7.21

Relate $X_d(e^{j\Omega})$ back to the original spectrum $X_c(j\omega)$

Converting between DT $\leftrightarrow$ CT

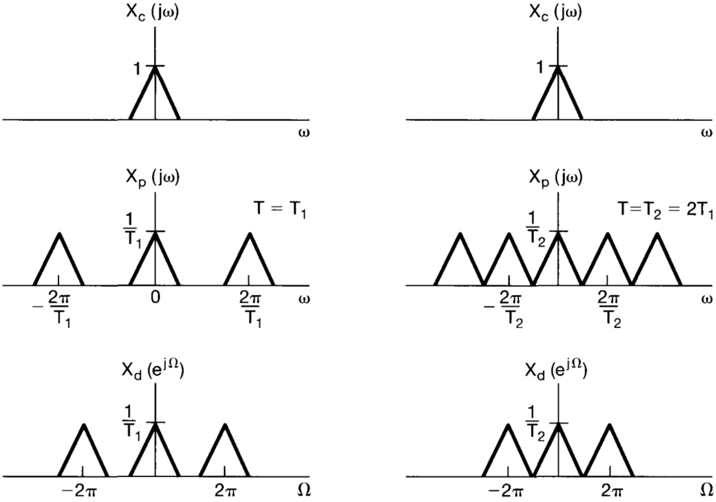
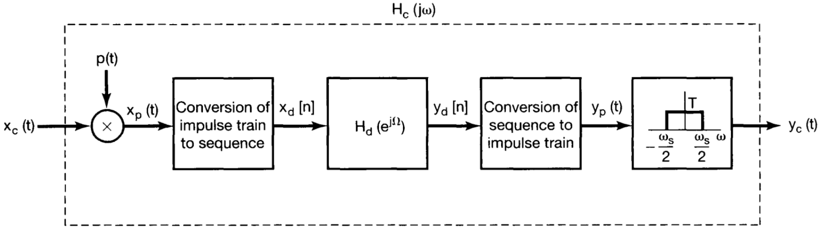


Image credit: Oppenheim 7.4



The converted signal $x_d[n]$ enters a DT system:

Image credit: Oppenheim 7.4

Converting between DT $\leftrightarrow$ CT

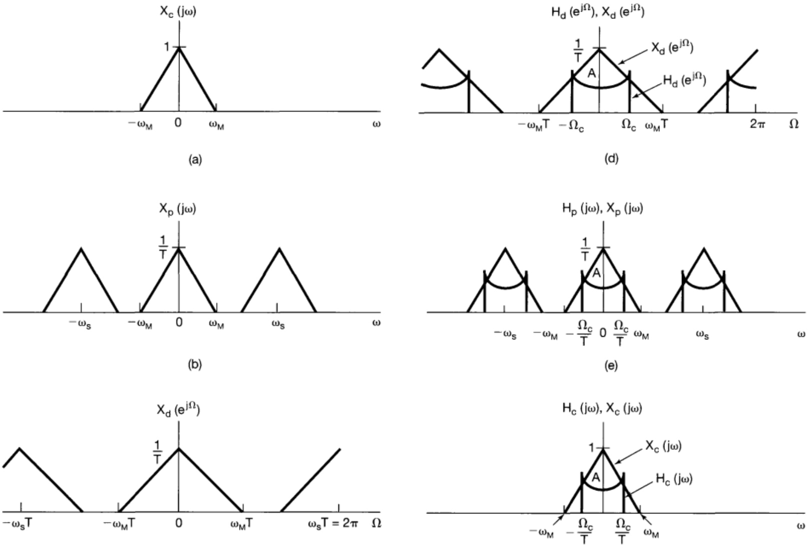
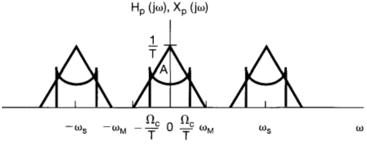


Image credit: Oppenheim 7.4



(e)

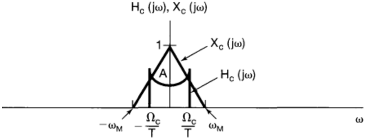
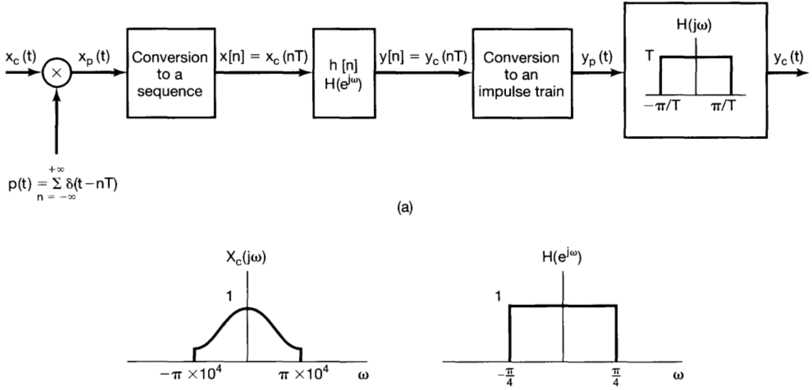


Image credit: Oppenheim 7.4

Example



Sketch: $X_p(j\omega)$, $X(e^{j\omega})$, $Y(e^{j\omega})$, $Y_p(j\omega)$, $Y_c(j\omega)$ if $1/T = 20kHz$.

Image credit: Oppenheim Problem 7.29

Example

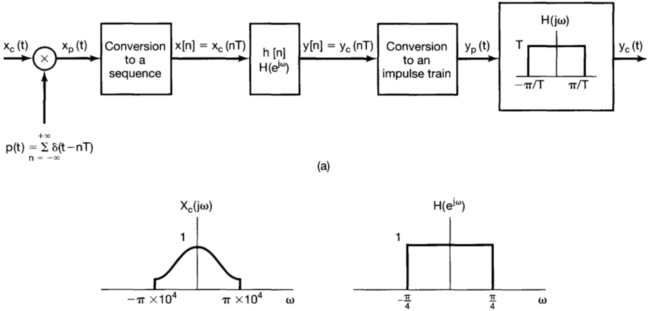
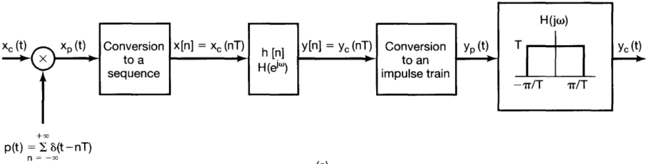


Image credit: Oppenheim Problem 7.29

Example



(a)

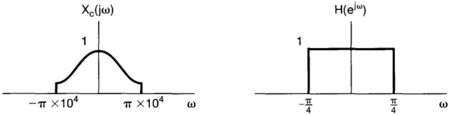


Image credit: Oppenheim Problem 7.29

Example

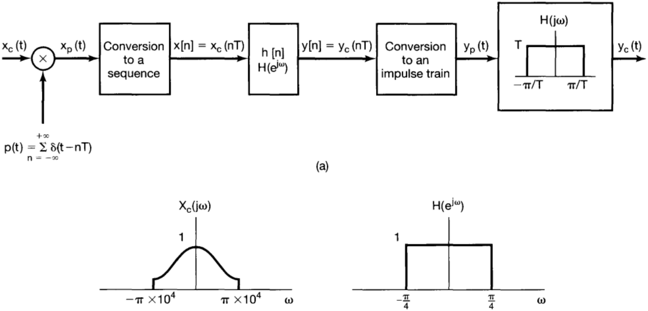


Image credit: Oppenheim Problem 7.29

Sample with DT impulse train of period N:

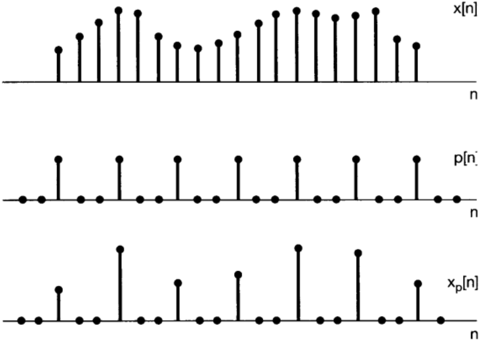


Image credit: Oppenheim 7.5

Sampling of discrete-time signals

Take similar approach as we did in CT:

Sampling of discrete-time signals

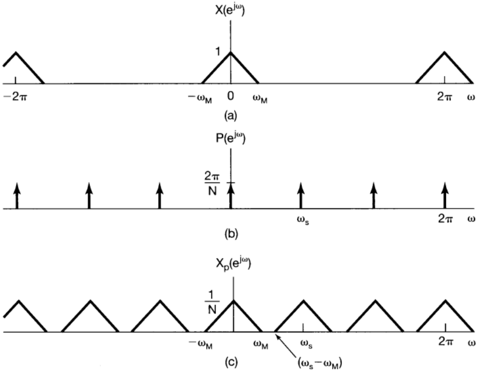


Image credit: Oppenheim 7.33

Aliasing can happen in DT; some differences due to DT frequency range (π is the highest frequency).

Exercise: suppose $x[n]$ has $X(e^{j\omega})$ that is 0 for $3\pi/7 \leq |\omega| \leq \pi$. What is the largest sampling period N we can use without aliasing?

Decimation

Sampling and then transmitting a DT signal in this way is inefficient

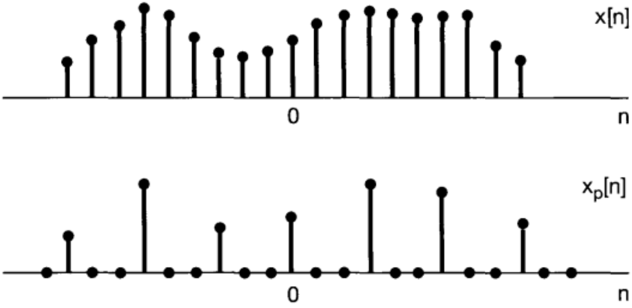


Image credit: Oppenheim 7.31

Decimation

We can *compress* the representation:

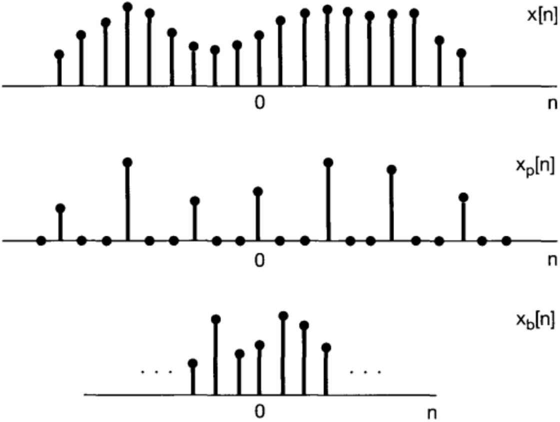
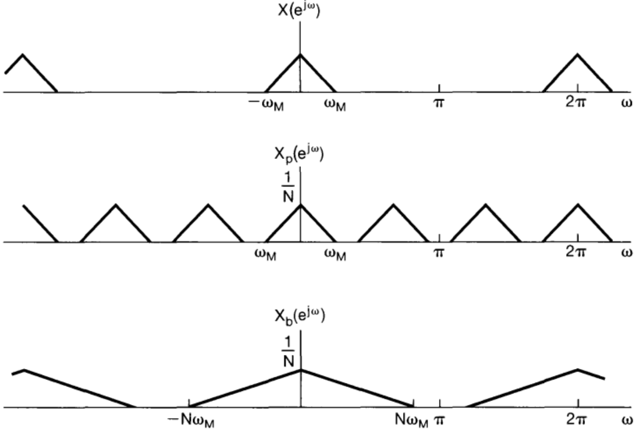


Image credit: Oppenheim 7.34

Decimation

Frequency domain effect:

Decimation spreads out the spectrum



If original signal was CT, say that decimation has *downsampled* it.

Image credit: Oppenheim 7.35

Interpolation (upsampling)

Opposite of decimation: add $N-1$ points between.

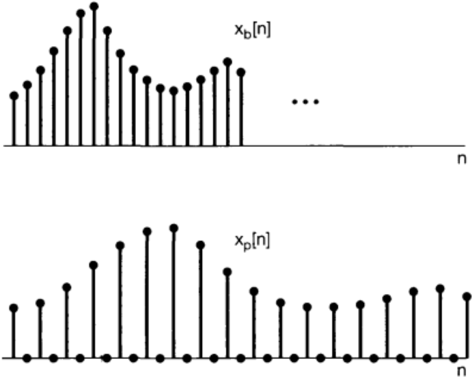


Image credit: Oppenheim 7.5

Interpolation (upsampling)

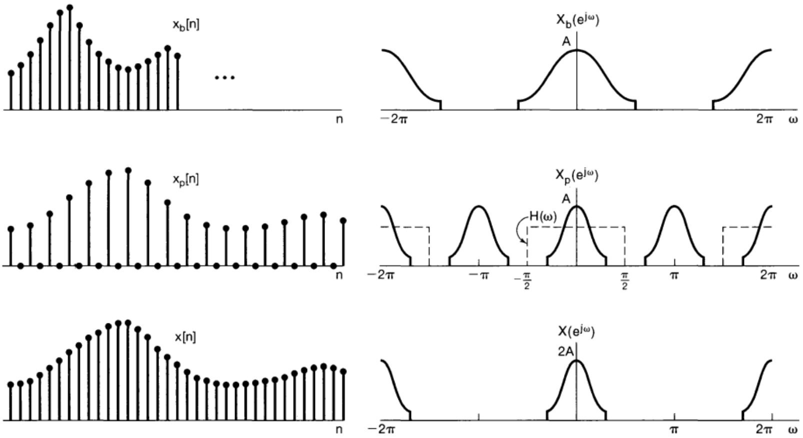


Image credit: Oppenheim 7.5

Oppenheim problem 7.19,

7.19. Consider the system shown in Figure P7.19, with input $x[n]$ and the corresponding output $y[n]$. The zero-insertion system inserts two points with zero amplitude between each of the sequence values in $x[n]$. The decimation is defined by

$$y[n] = w[5n],$$

where $w[n]$ is the input sequence for the decimation system. If the input is of the form

$$x[n] = \frac{\sin \omega_1 n}{\pi n},$$

determine the output $y[n]$ for the following values of ω_1:

(a) $\omega_1 \leq \frac{3\pi}{5}$

(b) $\omega_1 > \frac{3\pi}{5}$

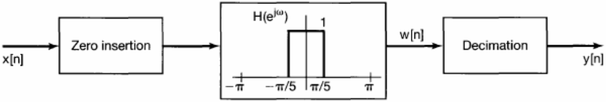


Figure P7.19

Example: down/upsampling

First case: $\omega_1 \leq \frac{3\pi}{5}$

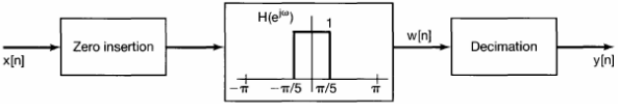


Figure P7.19

Second case: $\omega_1 > \frac{3\pi}{5}$

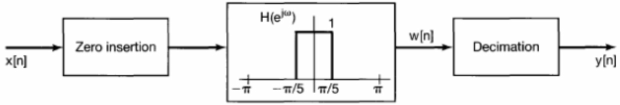


Figure P7.19

For next time

Content:

- moving into topic of modulation / communication systems

Action items:

1. Work on assignment 4
2. Prepare for quiz 8 on Tuesday (L17 and L18 material)
3. Tutorial Assignment 4 Monday

Recommended reading:

- From this class: Oppenheim 7.4-7.6
- Suggested problems: 7.17, 7.18, 7.20, 7.30, 7.32
- For next class: Oppenheim 8.0-8.4