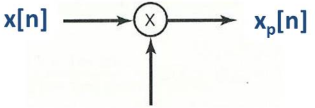
Homework: 4.50 & 4.51 of the attachment

Tutorial Problems: 7.41, 7.44, 7.47, 7.49

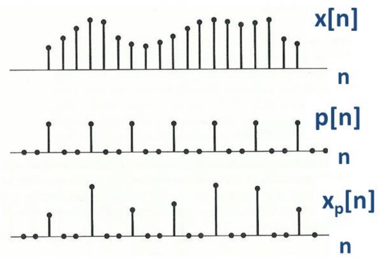
Sampling on Discrete-Time Signals

- In digital signal processing, we may sample the discrete-time signals
- Example: image, audio, video compression
- Its mathematical model is give below:



$$p[n] = \sum_{k=-\infty}^{+\infty} \delta[n - kN]$$

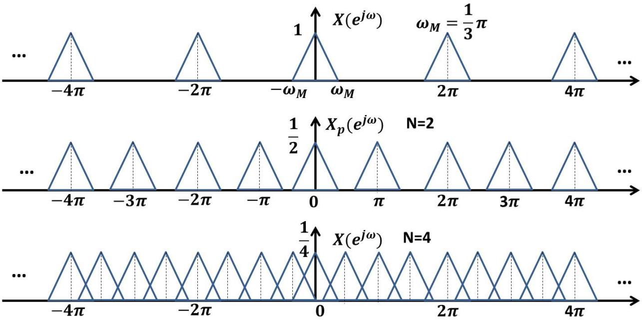
$$x_p[n] = \sum_{k=-\infty}^{+\infty} x[kN]\delta[n - kN]$$



Frequency Analysis

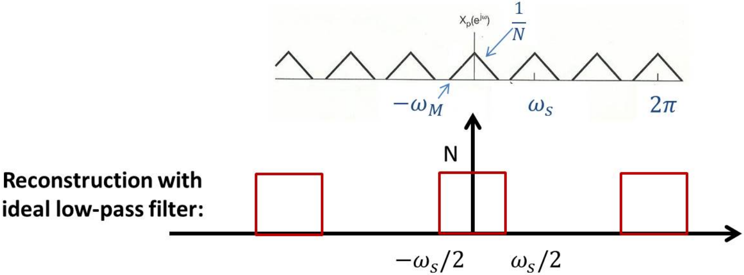
$$P(e^{j\omega}) = \frac{2\pi}{N}\sum_{k=-\infty}^{\infty}\delta(\omega - k\omega_s) \quad where\ \omega_s = 2\pi/N$$

$$X_p(e^{j\omega}) = \frac{1}{2\pi}\int_{2\pi} P(e^{j\theta})X(e^{j(\omega-\theta)})d\theta = \frac{1}{N}\sum_{k=0}^{N-1} X(e^{j(\omega-k\omega_s)})$$



Reconstruction

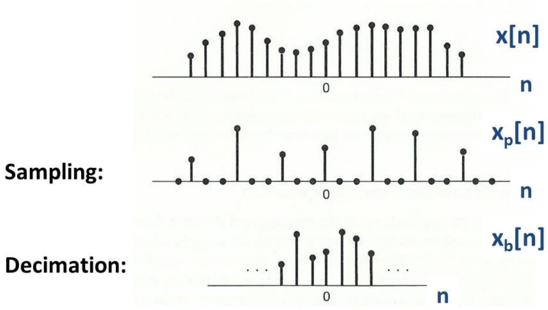
- Perfect reconstruction is applicable when $\omega_s > 2\omega_M \leftrightarrow N < \frac{\pi}{\omega_M}$



- Aliasing occurs when $\omega_s < 2\omega_M$

Decimation

- After sampling, there will be a great amount of redundancy
- Decimation: discrete-time sampling + remove zeros

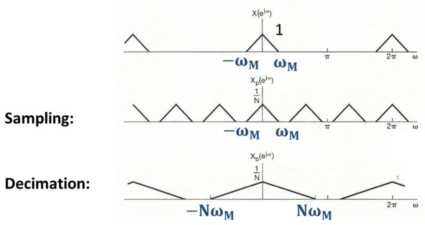


$$x_b[n] = x_p[nN] = x[nN]$$

Frequency Analysis

$$X_b(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x_b[n]e^{-j\omega n} = \sum_{n=-\infty}^{\infty} x_p[n]e^{-j\omega n/N} = X_p(e^{j\omega/N})$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} X(e^{j\omega/N-k\omega_s})$$



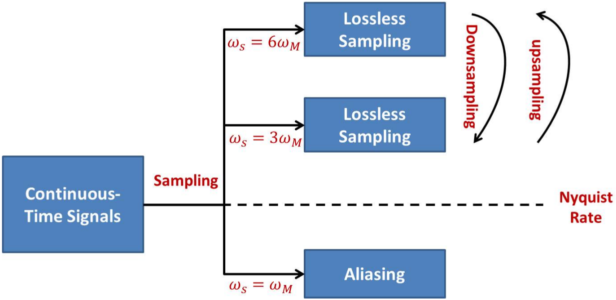
Condition for Perfect Reconstruction: $N\omega_M < \pi$

Intuitively, what happen in frequency domain when we do decimation?

Summary of DT Sampling

- Two operations: sampling vs. decimation
- What's their difference in frequency domain?
- $N\omega_M < \pi$ is always the criterion for lossless sampling

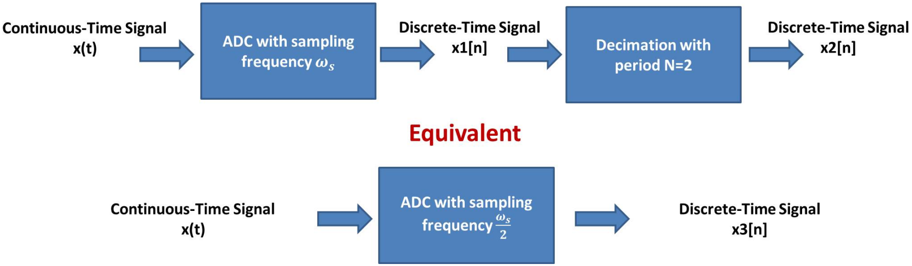
Anything Else?



- Downsampling: to reduce the sampling frequency
- Upsampling: to generate a DT signal with higher sampling frequency
- As long as Nyquist rate is satisfied, the transform between low-sampling-frequency version and high-sampling-frequency versions is lossless.
- How to do downsampling and upsampling in discrete-time domain?

Downsampling in DT Domain

- Downsampling: a general procedure to reduce the sampling frequency

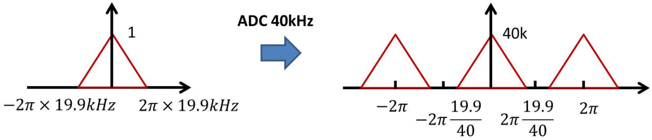


Obviously, $x_2[n] = x_3[n]$

Downsampling: from $x_1[n]$ to $x_3[n]$

Downsampling Example (1/2)

- Suppose we have a clip of voice, $x(t)$, with bandwidth =19.9kHz
- It can be converted to DT signal with sampling frequency 40kHz, denoted as $x_1[n]$

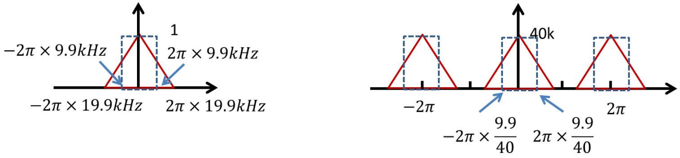


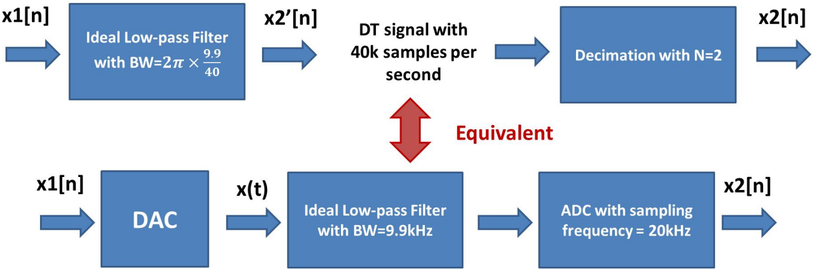
- Based on $x_1[n]$, if we want to save the voice information within 9.9kHz into another DT signal, what can we do?

One Choice:

How can we generate x2[n] in discrete-time domain?

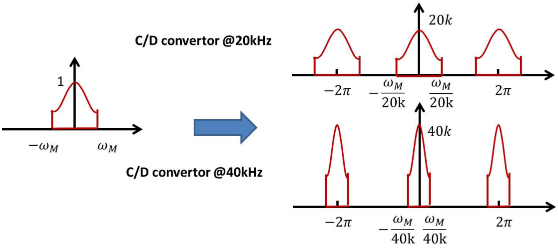
Downsampling Example (2/2)





Upsampling

- Upsampling: a procedure to generate a sequence with higher sampling frequency
- Superpose the following two digital sound clips
  - Audio clip 1: Bandwidth= 19.9kHz, sampled at 40kHz
  - Audio clip 2: Bandwidth= 9.9kHz, sampled at 20kHz
- How to double the sampling frequency of audio clip 2 (40kHz), so that both clips can be merged?



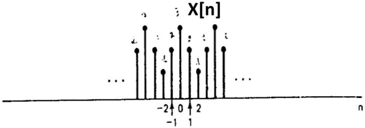
Recap: Time Reversal and Expansion

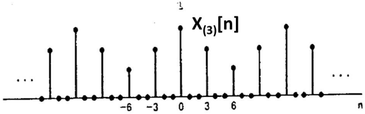
- Time Reversal

$$x[-n] \longleftrightarrow X(e^{-j\omega})$$

- Time Expansion
  - Define $x_{(k)}[n] = \begin{cases} x[n/k], & \text{if } n \text{ is a multiple of k} \\ 0, & \text{Otherwise} \end{cases}$, then

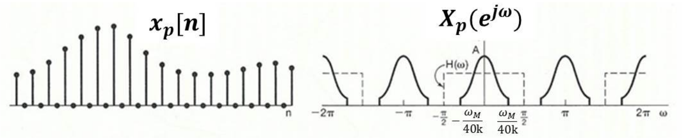
$$x_{(k)}[n] \longleftrightarrow X(e^{jk\omega}) \tag{1}$$





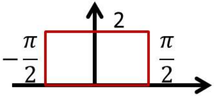
- Time expansion (Insert zeros):

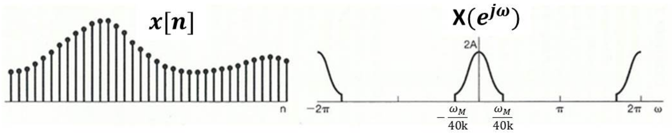
$$x_p[n] = x_{b(2)}[n] \longleftrightarrow X_p(e^{j\omega}) = X_b(e^{j2\omega})$$



- Low-pass filtering:

Ideal Low-Pass Filter:





Problem

Given a CT signal with bandwidth ω_M. Suppose we already sampled it by an ADC with sampling frequency $7\omega_M$. How to generate a DT signal with sampling frequency $4\omega_M$ in DT domain?