POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

Multirate processing

Multirate processing

Given a signal x(n), sampled with sampling frequency (or sampling rate) Fs, multirate processing concerns processing the signal with different sampling rate Fs’ ≠ Fs:

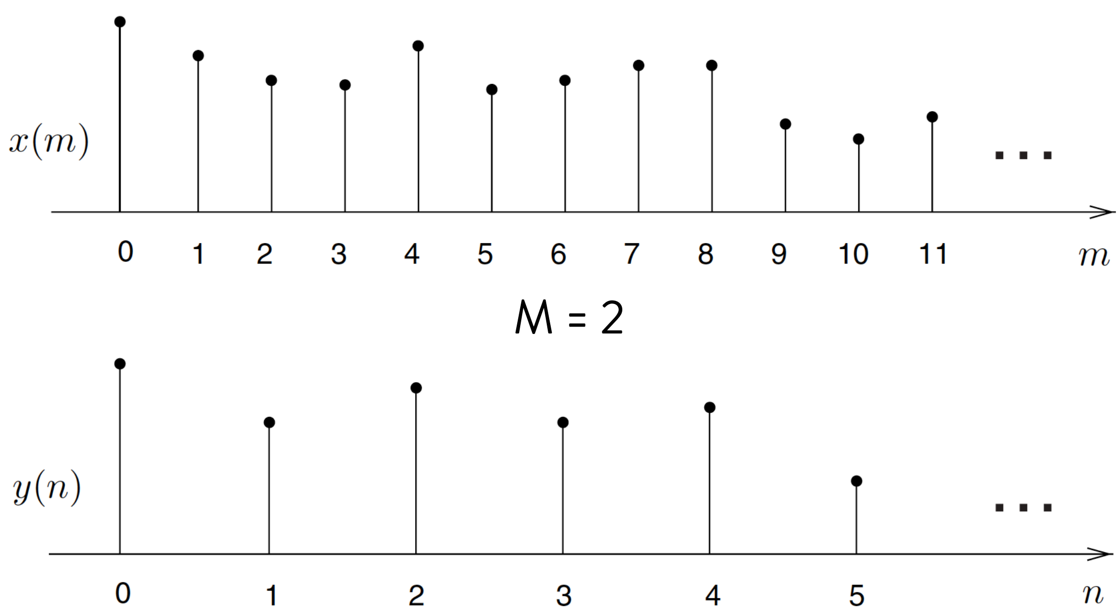
- Downsampling and decimation are related to Fs’ < Fs
- Upsampling and interpolation are related to Fs’ > Fs

Downsampling

Downsampling of a factor M means to keep one sample every M samples and discard the rest

$$y(n) = x(nM)$$

$$x(n) \longrightarrow \boxed{\downarrow M} \longrightarrow y(n)$$



Downsampling

$$y(n) = x(nM)$$

$$x(n) \longrightarrow \boxed{\downarrow M} \longrightarrow y(n)$$

- In Z domain,

$$Y(z) = \sum_{n=-\infty}^{+\infty} y(n) z^{-n} = \sum_{n=-\infty}^{+\infty} x(nM) z^{-n} = \sum_{m=-\infty}^{+\infty} x(m) \left[\frac{1}{M} \sum_{k=0}^{M-1} e^{\frac{j2\pi km}{M}} \right] z^{-\frac{m}{M}}$$

with

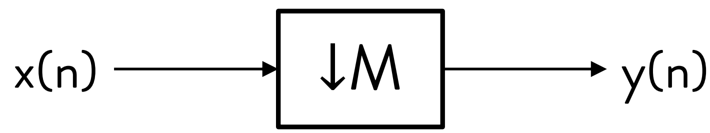
$$\frac{1}{M} \sum_{k=0}^{M-1} e^{\frac{j2\pi km}{M}} = \begin{cases} 1 & m \text{ is multiple of } M \\ 0 & \text{otherwise} \end{cases}$$

$$\Downarrow$$

$$Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} x(m) \left[e^{-\frac{j2\pi k}{M}} \cdot z^{\frac{1}{M}} \right]^{-m} = \frac{1}{M} \sum_{k=0}^{M-1} X\left(e^{-\frac{j2\pi k}{M}} \cdot z^{\frac{1}{M}} \right)$$

Downsampling

y(n) = x(nM)

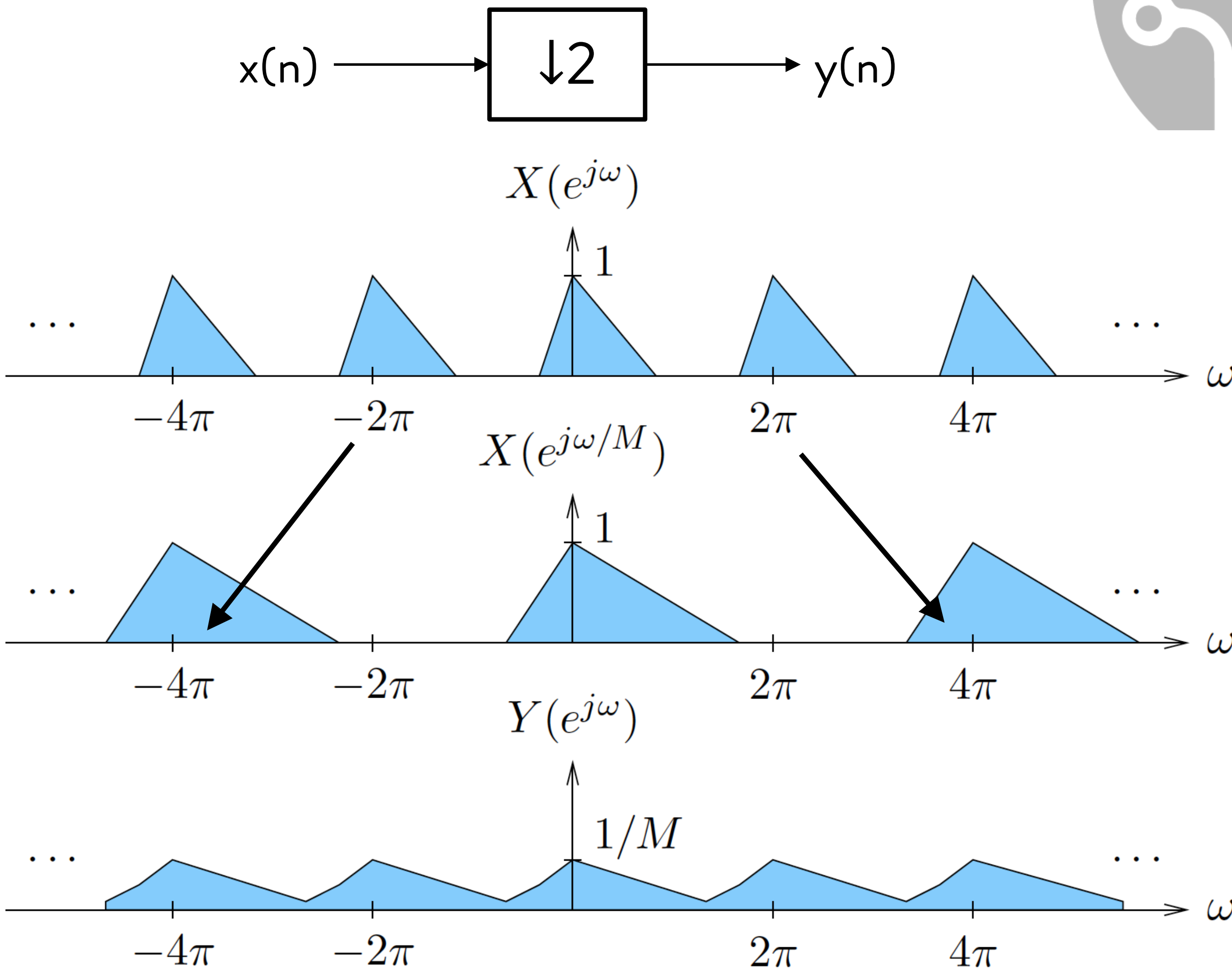


- In frequency domain Y(z) becomes

$$\boxed{Y(f) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(\frac{f-k}{M}\right)}$$

- The DTFT of y(n) is composed of copies of the DTFT of x(n) **expanded by M** and **repeated with period 1** in normalized frequency (or Fs in Hertz, or 2π in angular frequencies)
- **The gain is reduced by a factor of M**

Downsampling: example

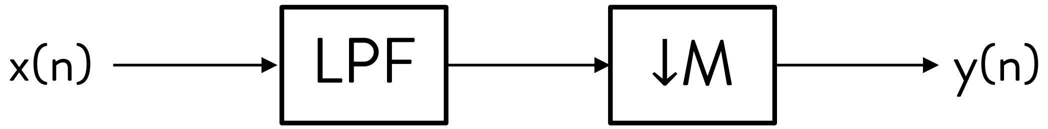


Aliasing in frequency occurs if the DTFT of x(n) is not limited to 1/(2M) (or π/M, or Fs/(2M))

Decimation

Decimation is related to downsampling the signal, but avoids frequency aliasing:

- The signal x(n) is filtered with a low-pass filter having cut-off frequency = 1/2M
- Then, the filtered signal is downsampled by a factor M



$$H(f) = \begin{cases} 1 & |f| \leq \frac{1 * Fs}{2M} \\ 0 & \text{otherwise} \end{cases}$$

or pi/M, as always

Es 27: downsampling and decimation

Given x(n) defined as the sum of two sinusoidal signals, sampled at Fs = 500 Hz with duration 3 seconds, one with frequency 50 Hz and the other one with frequency 100Hz:

- Downsample x(n) with downsampling factor M = 4
- Decimate x(n) with decimation factor M = 4, using a FIR filter with order 64.
- Plot the DFTs of x(n), of the downsampled and of the decimated signals vs frequency [Hz] in the same figure and comment on the results.
- Try also M = 2 and see what happens

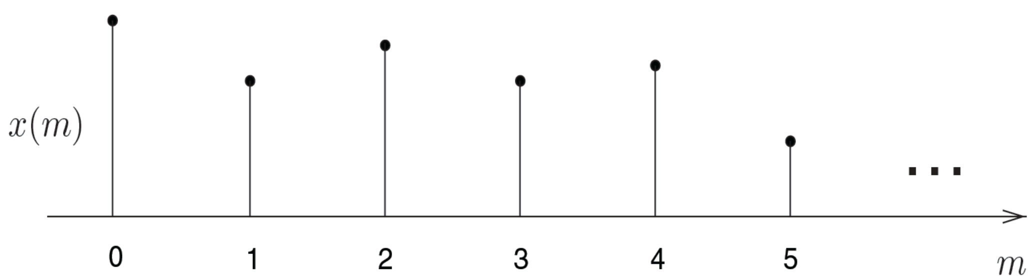
Upsampling

Upsampling of a factor L means to insert L -1 zeros

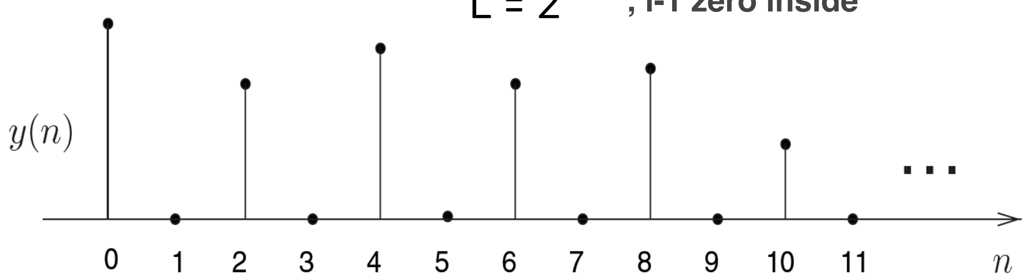
between the input signal samples

$$y(n) = \begin{Bmatrix} x(n/L) & n = kL \\ 0 & \text{otherwise} \end{Bmatrix}$$

$$x(n) \longrightarrow \boxed{\uparrow L} \longrightarrow y(n)$$



L = 2 , l-1 zero inside



Upsampling

$$y(n) = \left\{ \begin{matrix} x(n/L) & n = kL \\ 0 & \text{otherwise} \end{matrix} \right\}$$

$$x(n) \longrightarrow \boxed{\uparrow L} \longrightarrow y(n)$$

- In Z domain,

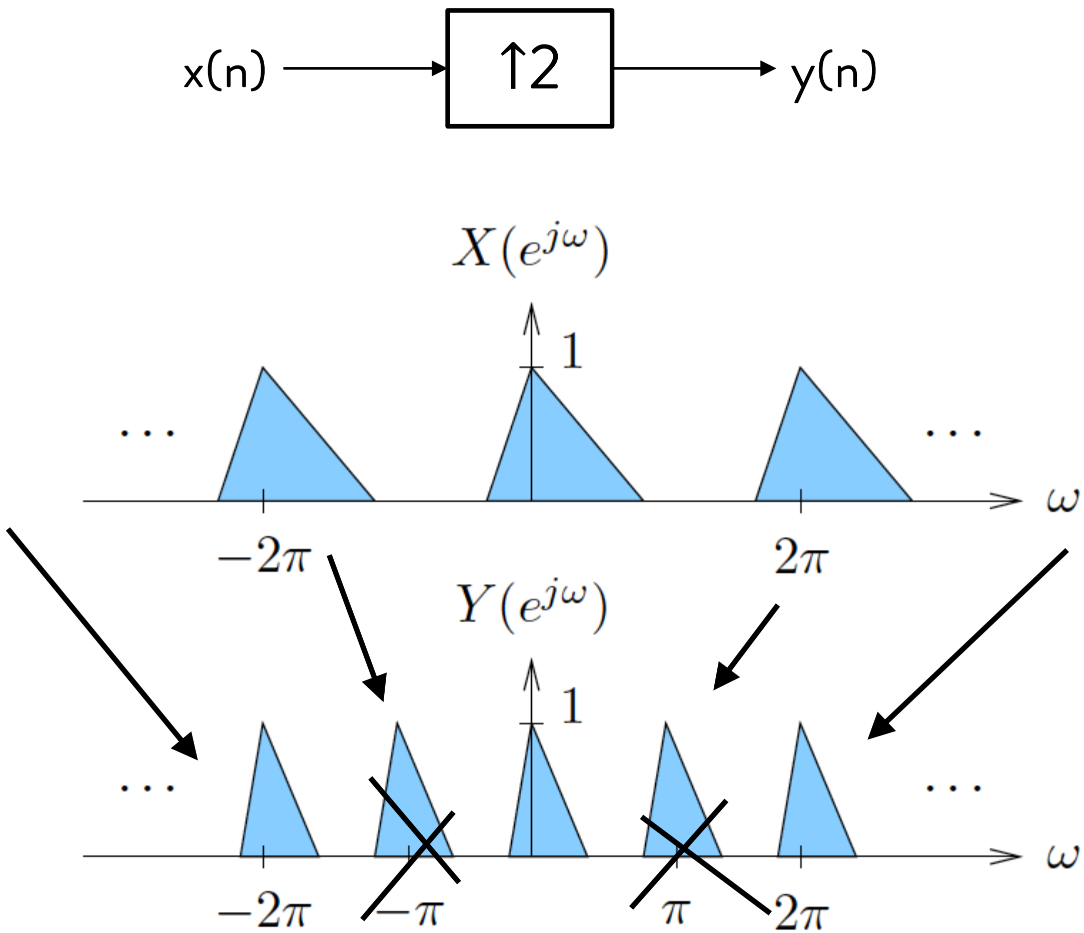
$$Y(z) = \sum_{n=-\infty}^{+\infty} y(n) z^{-n} = \sum_{k=-\infty}^{+\infty} x\left(\frac{kL}{L}\right) z^{-kL} = \sum_{k=-\infty}^{+\infty} x(k) z^{-kL} = X(z^L)$$

- In frequency domain,

$$\boxed{Y(f) = X(fL)}$$

- Upsampling compresses the DTFT by a factor of L

Upsampling: example



Spectral replicas do not overlap: upsampling just causes a compression of the spectrum, which has a new period of 1/L (or 2π/L, or Fs/(L))

Interpolation

ma nn voglio la copia in PI, quindi la devo eliminare, adottando un post filtraggio per eliminare la replica in PI

Idea: instead of zeros, what if we interpolate signal values?

- First, upsample the signal by a factor L
- Then, filter the signal with a low-pass filter with cut-off = 1/2L, which filters out the replicas and interpolate the signal samples

$$x(n) \longrightarrow \boxed{\uparrow L} \longrightarrow \boxed{\text{LPF}} \longrightarrow y(n)$$

$$H(f) = \begin{cases} L & |f| \leq \frac{1}{2L} \\ 0 & \text{otherwise} \end{cases}$$

or PI/L

Es 28: upsampling and interpolation

Given the downsampled signal defined in Es27 with M = 4,

- Create the signal x1 by upsampling the signal with a factor L = 4

Given the decimated signal defined in Es27 with M = 4,

- Create the signal x2 by interpolating the signal with a factor L = 4, using a FIR filter with order 64.
- Open a figure and create three subplots:
  1. In 1° subplot, plot the stem of the original signal x(n) until N = 120 time samples, x-axis in seconds.
  2. In 2° subplot, plot the stem of the downsampled and decimated signals with the same temporal duration as above
  3. In 3° subplot, plot the stem of x1 and x2 with the same temporal duration

Rational sampling rate conversion

Sampling rate change by a factor L/M can be easily

Implemented by cascading an interpolator with a decimator:

$$x(n) \longrightarrow \boxed{\uparrow L} \longrightarrow \boxed{\text{LPF}} \longrightarrow \boxed{\downarrow M} \longrightarrow y(n)$$

- The low-pass filter is built to delete replicas due to upsampling and avoid frequency aliasing due to downsampling

$$H(f) = \begin{cases} L & |f| \leq \min\{\frac{1}{2L}, \frac{1}{2M}\} \\ 0 & \text{otherwise} \end{cases}$$

QUINDI SOLO nell’ UP il filtro ha un’altezza diversa da 1 !!

Decimation and interpolation with MATLAB

- You can decimate a signal x(n) using ‘decimate(x, M)’: MATLAB performs a low-pass filtering + downsampling
- You can interpolate a signal x(n) using ‘interp(x, L)’: MATLAB performs an upsampling + low-pass filtering
- Which L and M to choose for a rational sampling rate conversion? Use ‘[L, M] = rat(q)’ to find the best rational approximation of q = L/M

Es 29: upsampling and interpolation [~exam 10/09/2019, 4pts]

Given the signal $x(t) = A_1 \cos(2 \text{ pi } f_1 t) + A_2 \cos(2 \text{ pi } f_2 t)$:

- Create the signal x(n) as x(t) with t from 0 to 0.5 seconds, sampled at Fs=8000 Hz. $A_1=0.7$, $A_2=0.5$, $f_1=1800$ Hz, $f_2=3600$ Hz

  L/F. = 6000/8000

- Create the signal y(n) by resampling x(n) with 6000 Hz, without using the MATLAB functions for automatic re-sampling
- Plot the magnitude of the DFTs of x(n), the upsampled signal, the filtered signal and y(n) over 2048 samples vs normalized frequency in [0, 1)