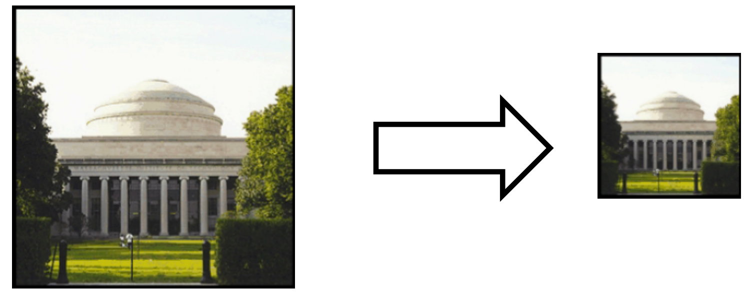
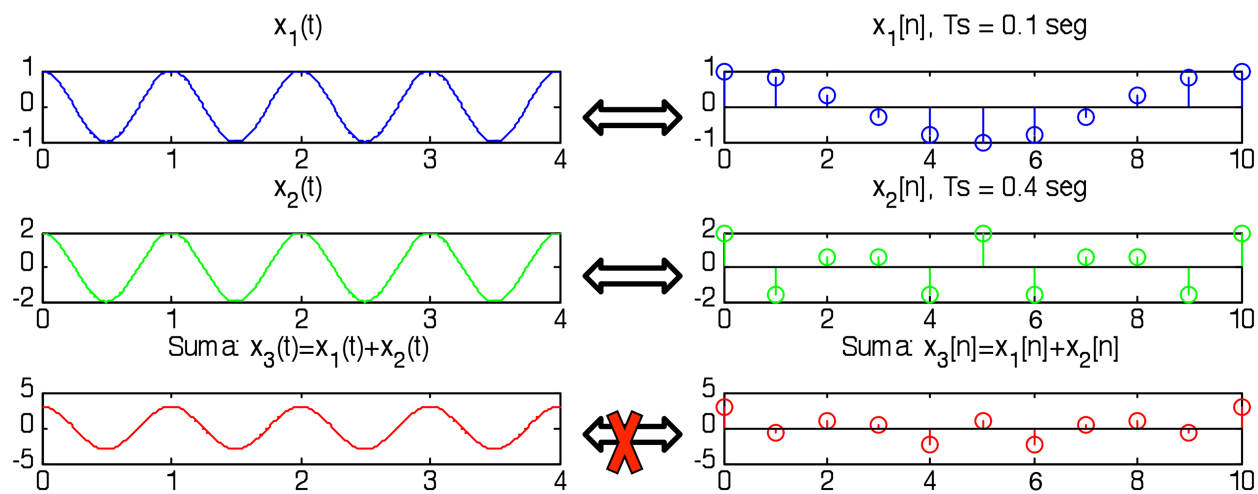
Muestreo de señales discretas

- ¿Para qué?
  - Compresión: Muestrear señales discretas es una forma de "comprimir la información"



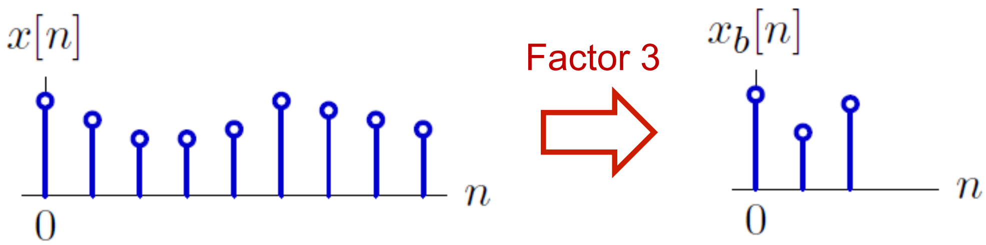
  - Para poder procesar en discreto señales continuas que han sido muestreadas a distintas tasas



  - Desde un punto de vista de análisis, los pasos son muy parecidos a los seguidos con el muestreo e interpolación de señales continuas

Muestreo de señales discretas: diezmado

- Al igual que en el caso continuo, describir el muestreo en el tiempo es muy sencillo



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

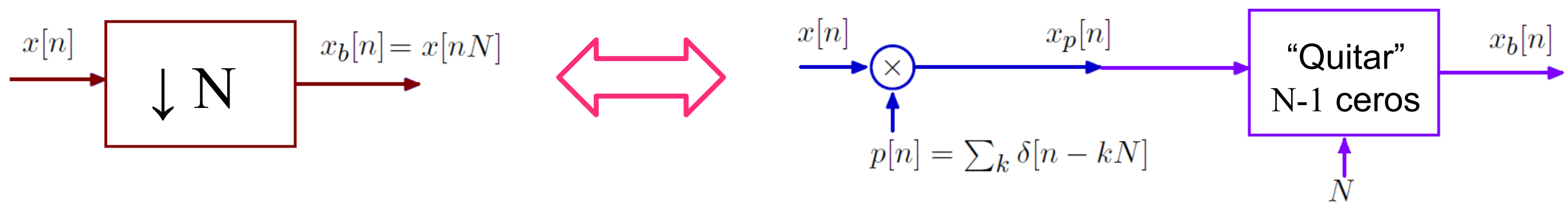
$x_b[-1] = x[-3]$
$x_b[0] = x[0]$
$x_b[1] = x[3]$
$x_b[2] = x[6]$
...

- Las preguntas claves son:
  - ¿Qué pasa en frecuencia?
  - ¿Cuándo puedo recuperar a partir de las señal muestreada la señal original?

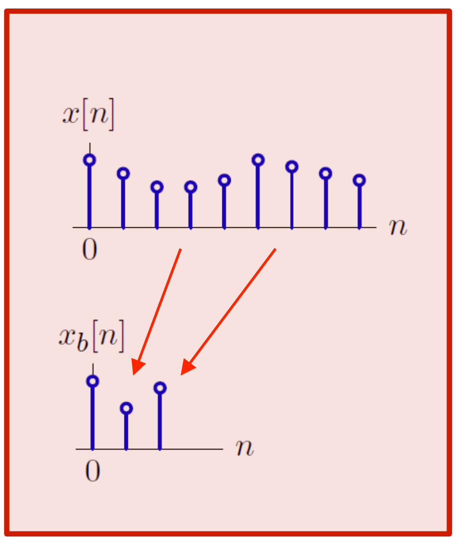
- Otras observaciones:
  - Al muestreo discreto en ocasiones se le llama diezmado (sobre todo si tenemos SD de longitud finita) ➔ Diezmamos la señal, la hacemos más corta
  - Del mismo modo, cuando no podemos recuperar la señal original, se dice que el proceso de diezmado ha provocado una pérdida de información

Diezmado en tiempo

- Al igual que en el caso continuo, entendemos el diezmado como la aplicación sucesiva de dos bloques

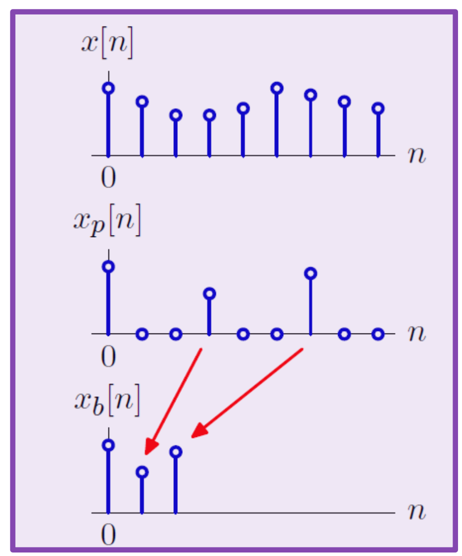


- En el dominio del tiempo, la señal es la misma ➔ Gráficamente



Un bloque

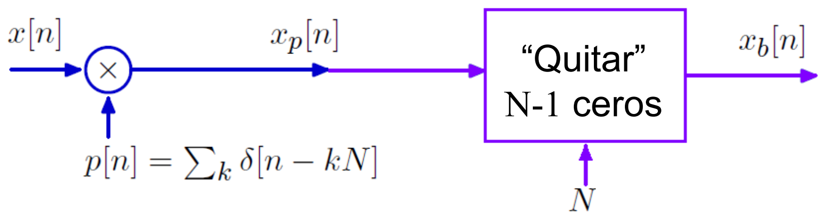
¡El resultado final es el mismo!



Dos bloques

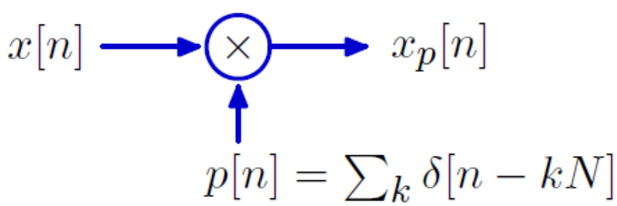
Diezmado en frecuencia

- Queremos relacionar la TF de x[n] con la de $x_b[n]$ ➔ Dos pasos



  - (1) Expresar la TF de $x_p[n]$ a partir de la TF de x[n] $\quad X(e^{j\Omega}) \rightarrow X_p(e^{j\Omega})$
  - (2) Expresar la TF de $x_b[n]$ a partir de la TF de $x_p[n]$ $\quad X_p(e^{j\Omega}) \rightarrow X_b(e^{j\Omega})$

- Paso (1):



$$P(e^{j\Omega}) = \frac{2\pi}{N}\sum_{k=-\infty}^{\infty}\delta(\Omega - \frac{2\pi}{N}k)$$

$$x_b[n] = x[n]\cdot p[n]$$

$$\Downarrow$$

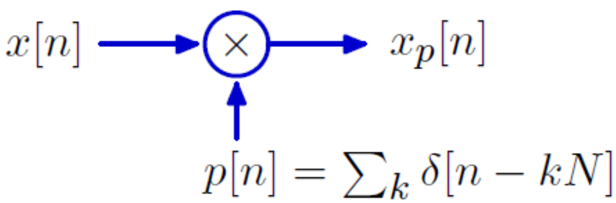
$$X_p(e^{j\Omega}) = \frac{1}{2\pi}\int_0^{2\pi} X(e^{j\theta})P(e^{j(\Omega-\theta)})d\theta$$

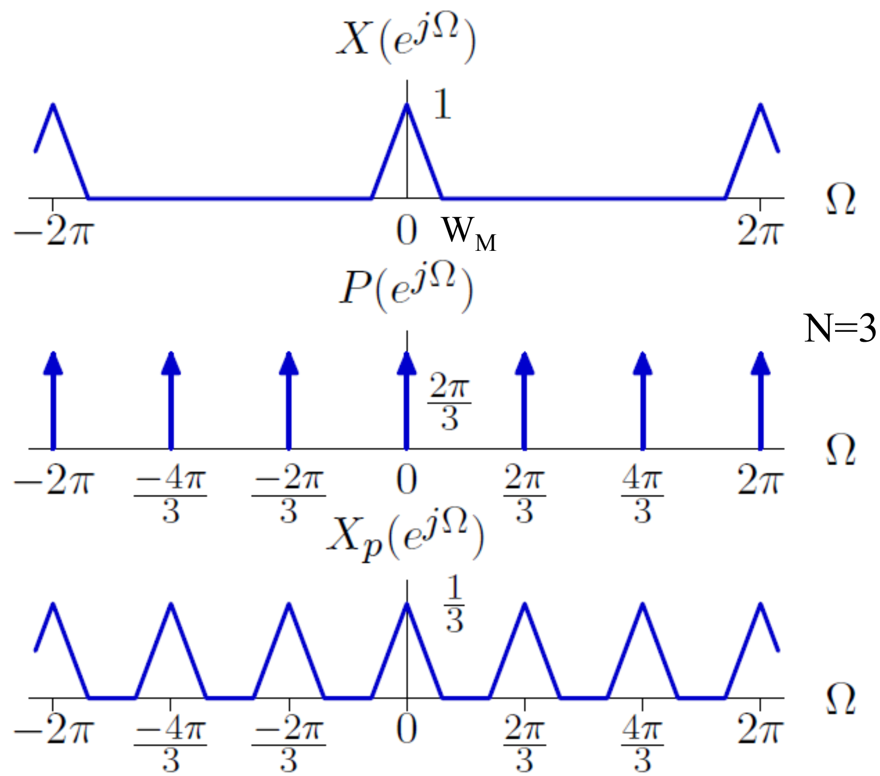
$$\Downarrow$$

$$X_p(e^{j\Omega}) = \frac{1}{N}\sum_{k=-\infty}^{\infty} X(e^{j(\Omega-\frac{2\pi}{N}k)})$$

Diezmado en frecuencia

- El paso (1) gráficamente





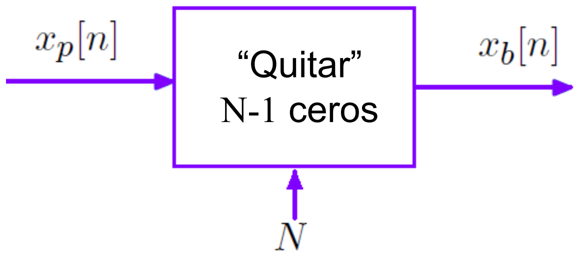
Nos aparecen réplicas en los múltiplos de $2\pi/N$ ➔ Ojo porque se pueden solapar

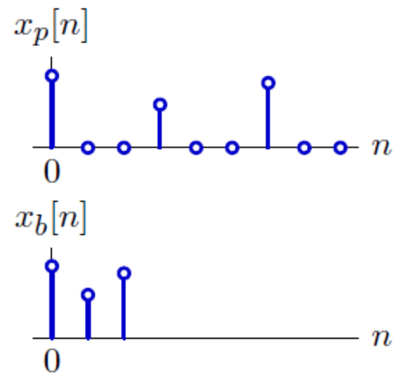
$$X_p(e^{j\Omega}) = \frac{1}{N}\sum_{k=-\infty}^{\infty} X(e^{j(\Omega - \frac{2\pi}{N}k)})$$

Si W_M es el BW de la señal $x[n]$ habrá solape si: $W_M > 2\pi/N - W_M$ ➔ $2W_M > 2\pi/N$ ➔ $N > 2\pi/W_M$

Diezmado en frecuencia

- Paso (2):





- Opción 1: a través de propiedades (expansión/compresión)
- Opción 2: a través de la fórmula de análisis

$$X_b(e^{j\Omega}) = \sum_n x_b[n]e^{-j\Omega n} = \sum_n x_p[3n]e^{-j\Omega n} = \sum_k x_p[k]e^{-j\Omega k/3} = X_p(e^{j\Omega/3})$$

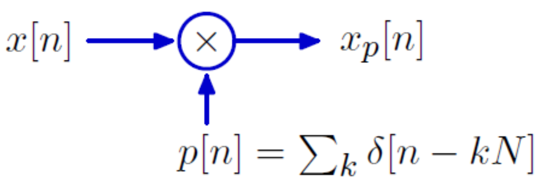
- Para el caso general

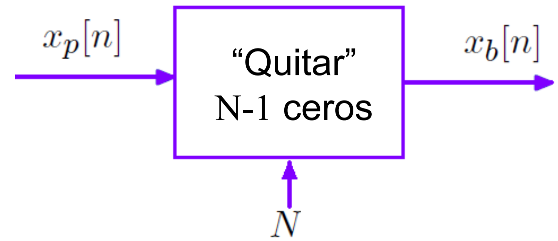
$$\boxed{X_b(e^{j\Omega}) = X_p(e^{j\Omega/N})}$$

¡Comprimir en el tiempo equivale a expandir en frecuencia!

Diezmado en frecuencia

- Pasos (1) + (2):





$$\boxed{X_p(e^{j\Omega}) = \frac{1}{N}\sum_{k=-\infty}^{\infty} X\left(e^{j(\Omega - \frac{2\pi}{N}k)}\right)}$$

$$\boxed{X_b(e^{j\Omega}) = X_p(e^{j\Omega/N})}$$

- Réplicas y expansión:

$$\boxed{X_b(e^{j\Omega}) = X_p(e^{j\Omega/N}) = \frac{1}{N}\sum_{k=-\infty}^{\infty} X\left(e^{j(\frac{\Omega}{N} - \frac{2\pi}{N}k)}\right) = \frac{1}{N}\sum_{k=-\infty}^{\infty} X\left(e^{j(\frac{\Omega - 2\pi k}{N})}\right)}$$

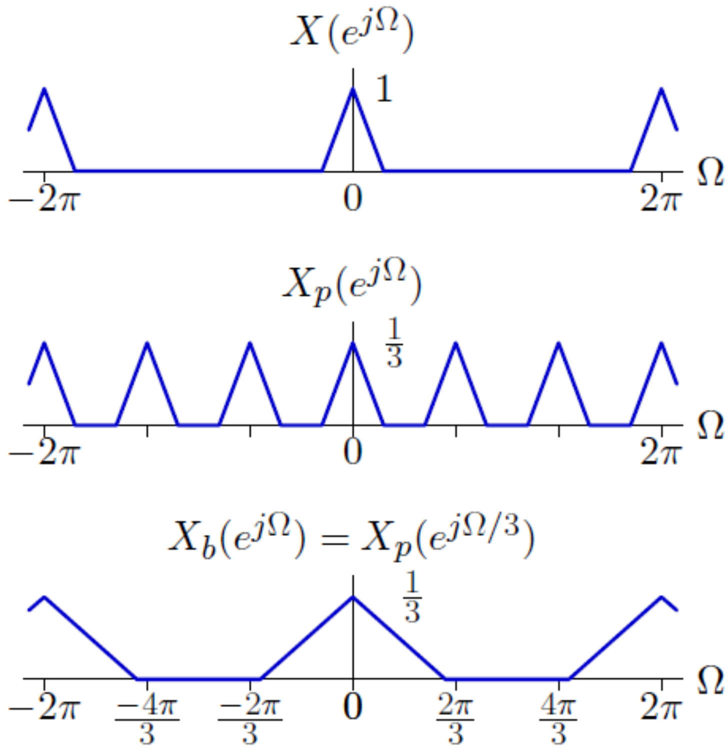
Recuérdese que el orden del desplazamiento compresión es importante ➔ Las fórmulas nos están diciendo que las réplicas están en los múltiplos de 2π

Diezmado en frecuencia: gráficamente

- Pasos (1) + (2):

Paso (1):

Paso (2):



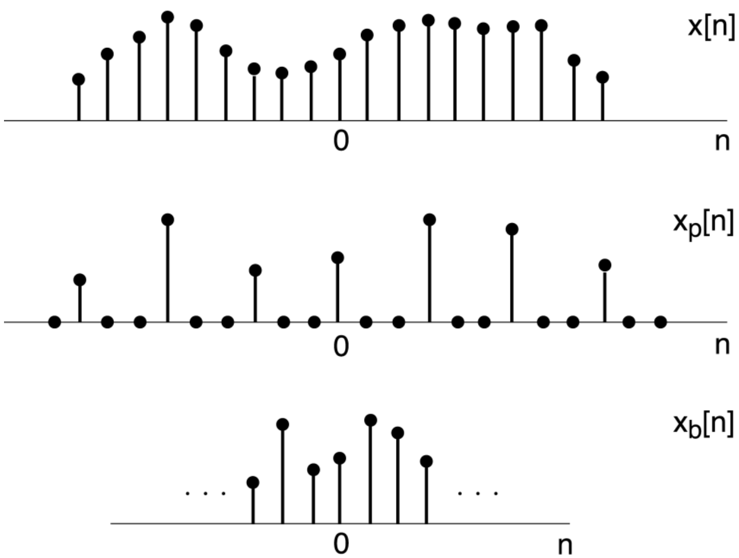
Cambio de amplitud (1/N)

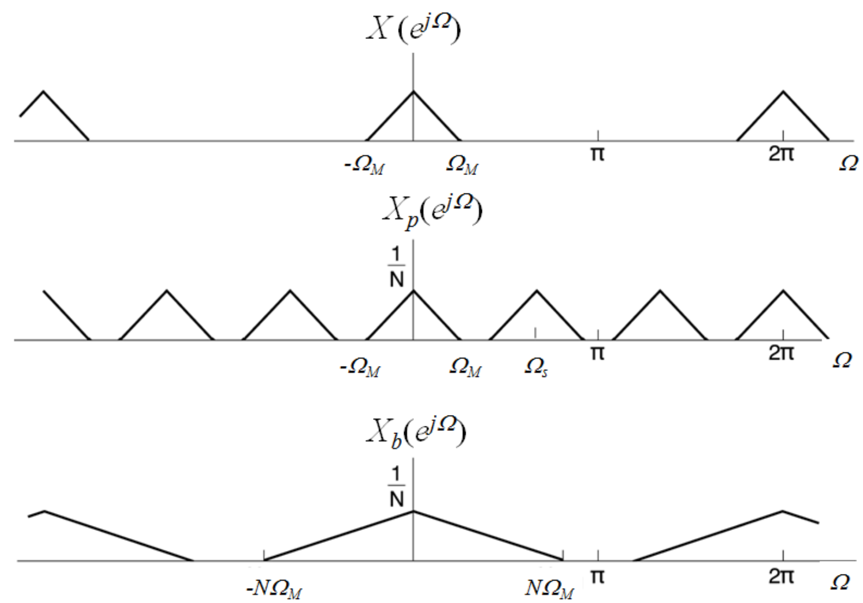
Ensanchamiento (por un factor N)

- Si no hay solape, es muy sencillo, si hay solape a veces nos liamos ➔ Método "universal"
  a) Nos quedamos con la señal original entre $-\pi$ y π
  b) Dividimos la amplitud por N y ensanchamos por un factor N
  c) Replicamos la señal obtenida en los múltiplos de 2π

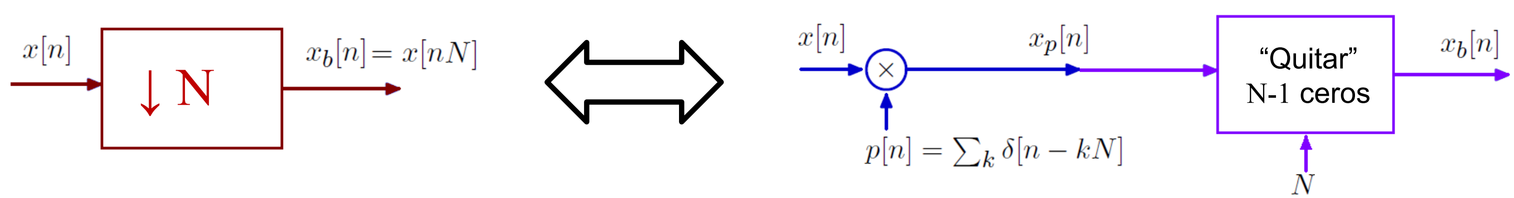
Diezmado: gráficas tiempo y frecuencia

- Pasos (1) + (2):



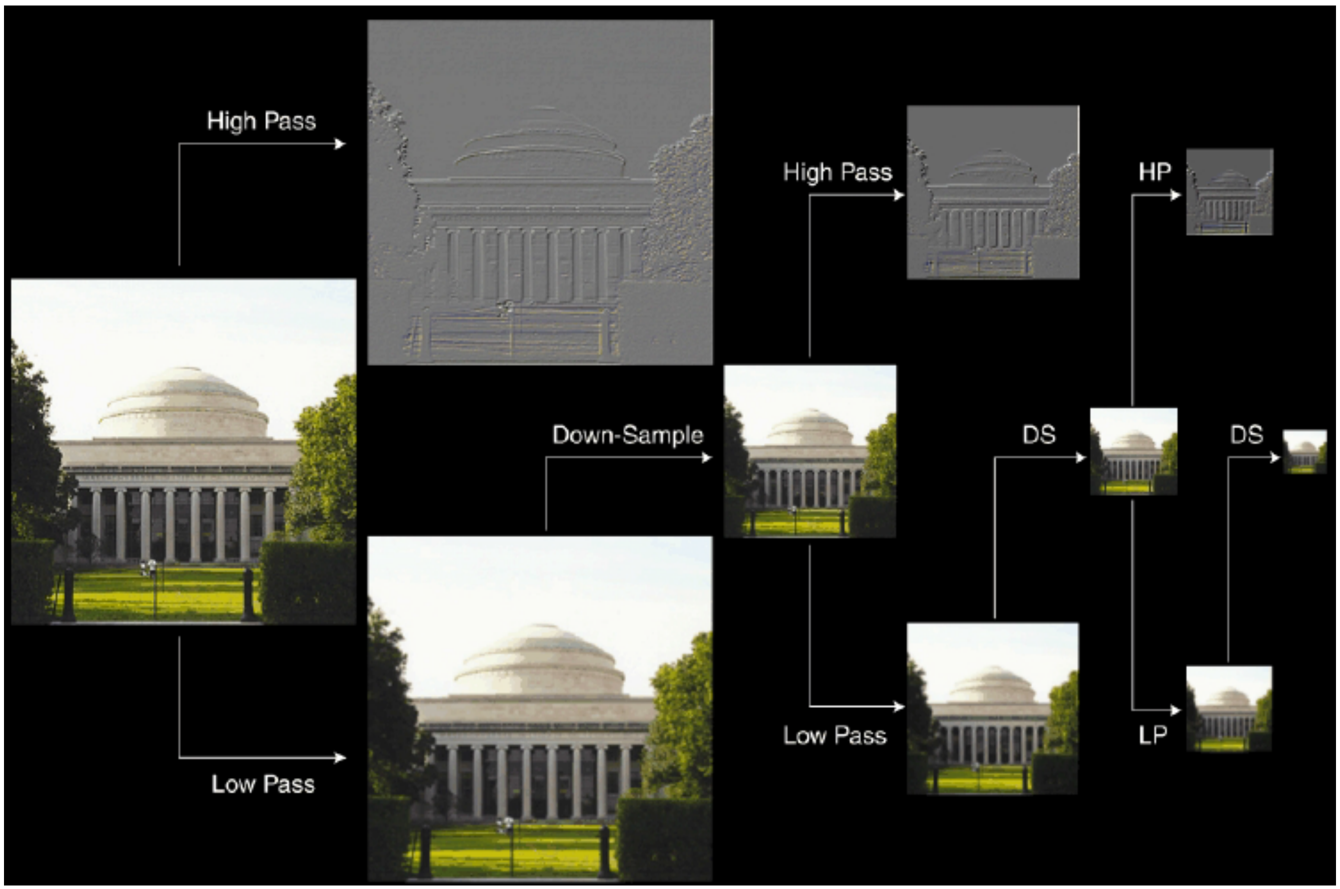


- Recordemos que la señal intermedia no existe, simplemente la dibujamos para entender mejor lo que está pasando ➔ "Dibujamos un bloque, para entenderlo lo dividimos en dos"



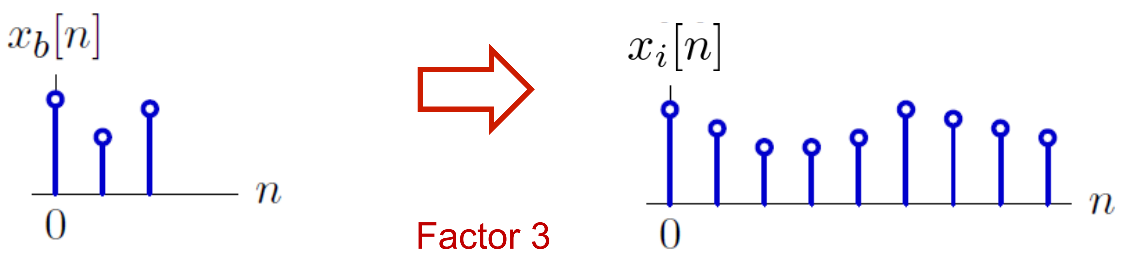
Diezmado: cuestiones prácticas

- Si el ancho de banda es menor que π/N no hay solapamiento
- Al igual que en el caso continuo, conviene poner un filtro paso bajo antisolapamiento ➔ frecuencia del filtro π/N
- Ejemplo:

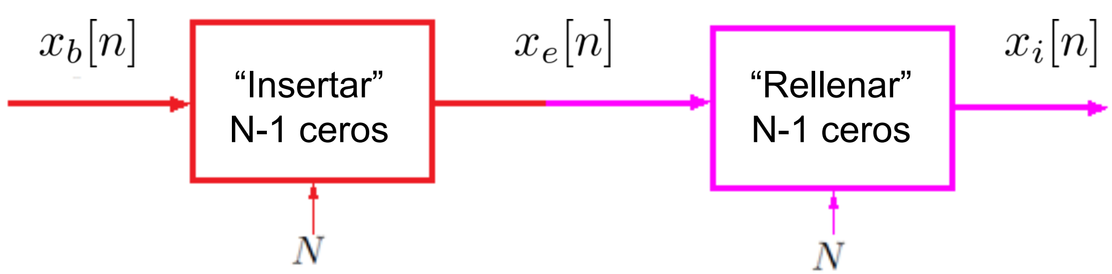


Interpolación de señales discretas

- Interpolación de señales discretas: ¿para qué?
  - Para recuperar una señal previamente diezmada
  - Para “expandir” una señal discreta
  - Para conseguir una señal equivalente a haber muestreado más rápido

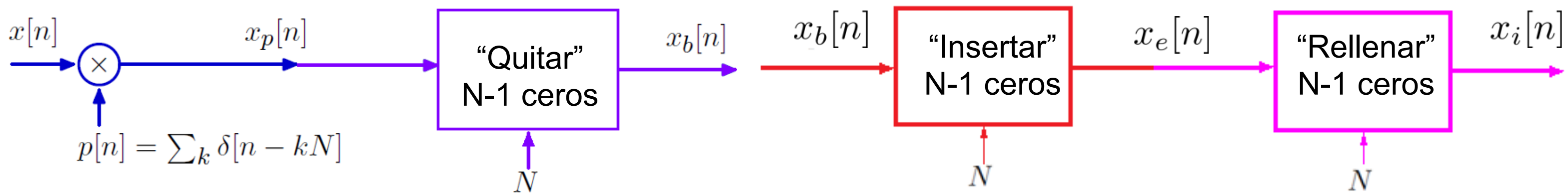


- Lo ejecutamos en dos pasos y, además, “utilizamos” dos bloques
  - Paso/Bloque (1): “inserta ceros” Paso/Bloque (2): “rellena ceros”

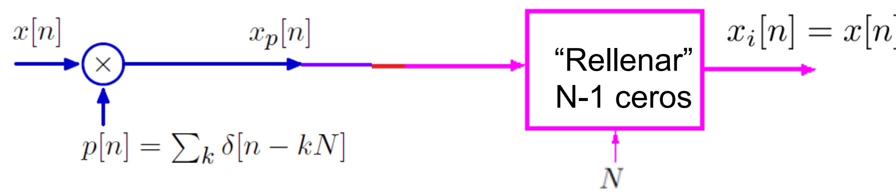


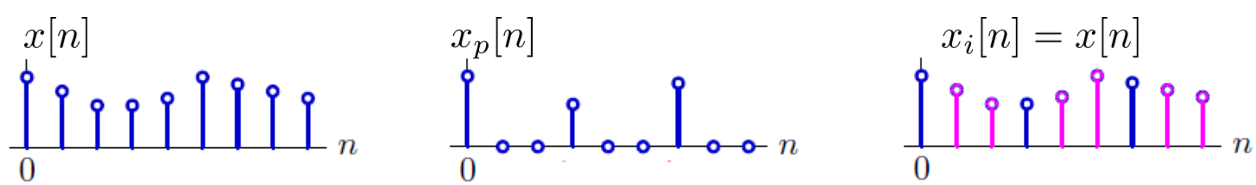
Interpolación de señales discretas

- ¿Cómo diseñamos esos bloques? ➔ Idea: recuperar señal diezmada



- Observaciones:
  - El objetivo de diseño es conseguir que $x_i[n]=x[n]$
  - Las señales $x_p[n]$ y $x_e[n]$ son iguales ➔ Para conseguir nuestro objetivo basta con analizar y diseñar



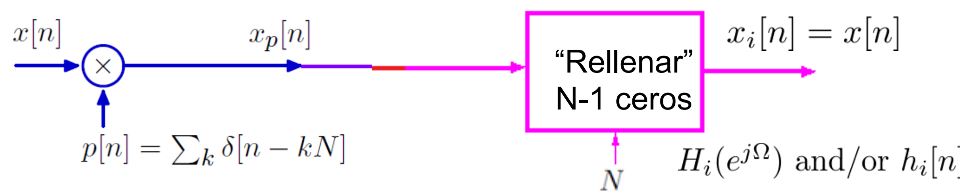


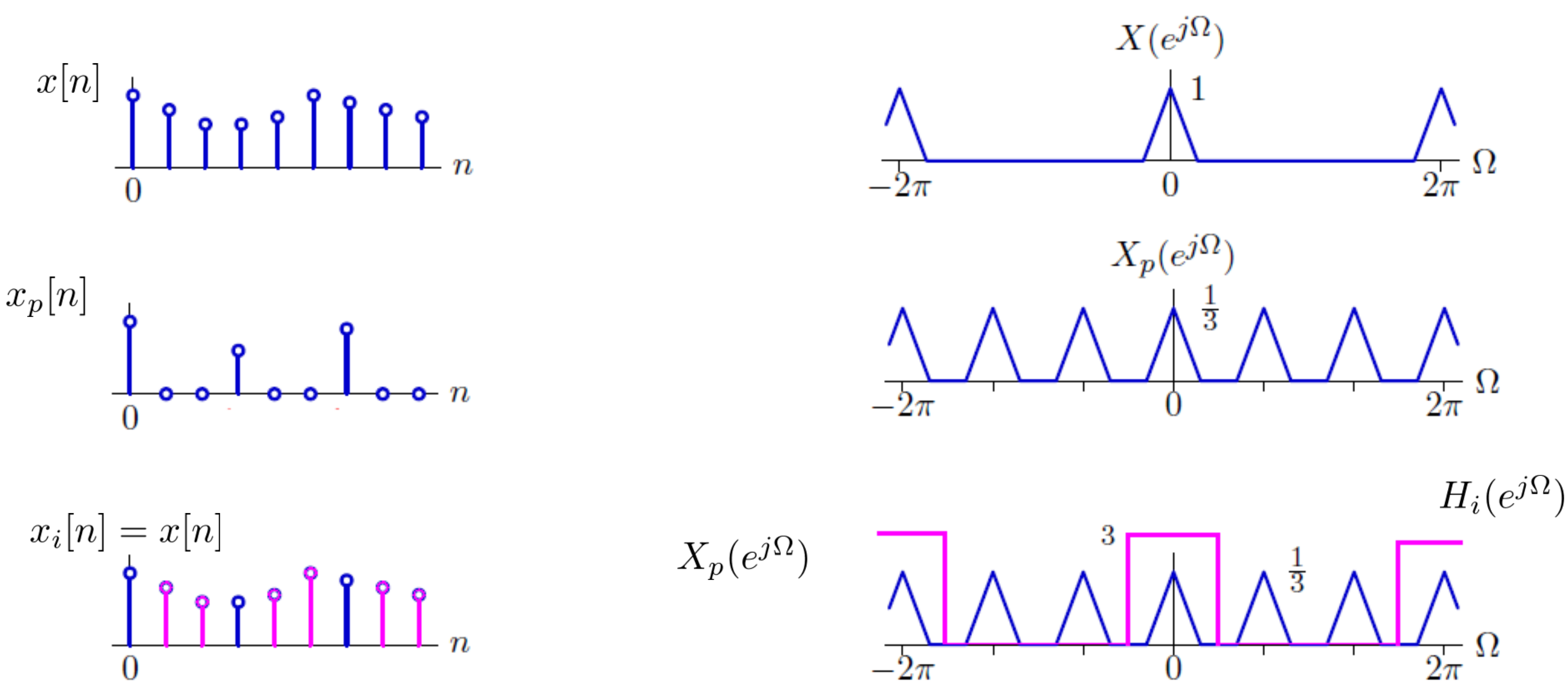
  - Asumiremos que el bloque a diseñar es LTI y por lo tanto nos bastará con especificar su RI o su RF ➔ Lo haremos en frecuencia porque es más fácil

Interpolación de SD: dominio de la frecuencia

- Lo analizamos y diseñamos en el dominio de la frecuencia

$$H_i(e^{j\Omega}) \Rightarrow h_i[n]$$

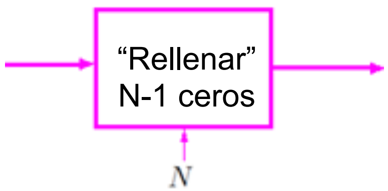


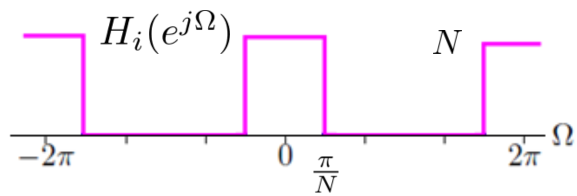


$$X_i(e^{j\Omega}) = H_i(e^{j\Omega})X_p(e^{j\Omega}) = X(e^{j\Omega})$$

Interpolador ideal de SD

- En frecuencia el interpolador ideal es un filtro paso bajo de ganancia N





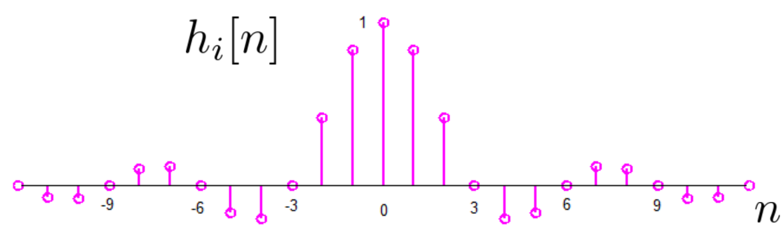
$$H_i(e^{j\Omega}) = \begin{cases} N, & |\Omega| \le \pi/N \\ 0, & \pi/N \le |\Omega| \le \pi \end{cases}$$

$$\text{periódica } 2\pi$$

- ¿En tiempo? ➔ $h_i[n] = TF^{-1}\left\{H_i(e^{j\Omega})\right\}$

$$h_i[n] = \frac{1}{2\pi}\int_{-\pi/N}^{\pi/N} N \cdot e^{j\Omega n} d\Omega = \frac{N}{2\pi}\frac{2j sin(\pi n/N)}{jn} = \frac{sin(\pi n/N)}{\pi n/N} = sinc(n/N)$$

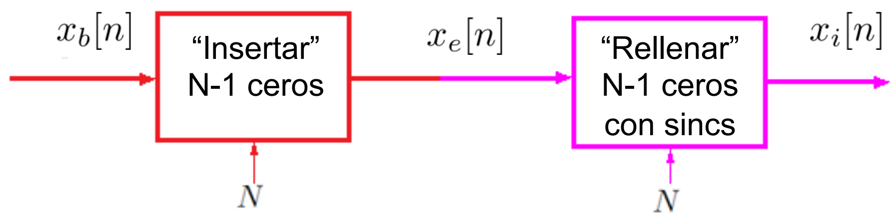
- ¡¡Vuelve a ser una sinc!!

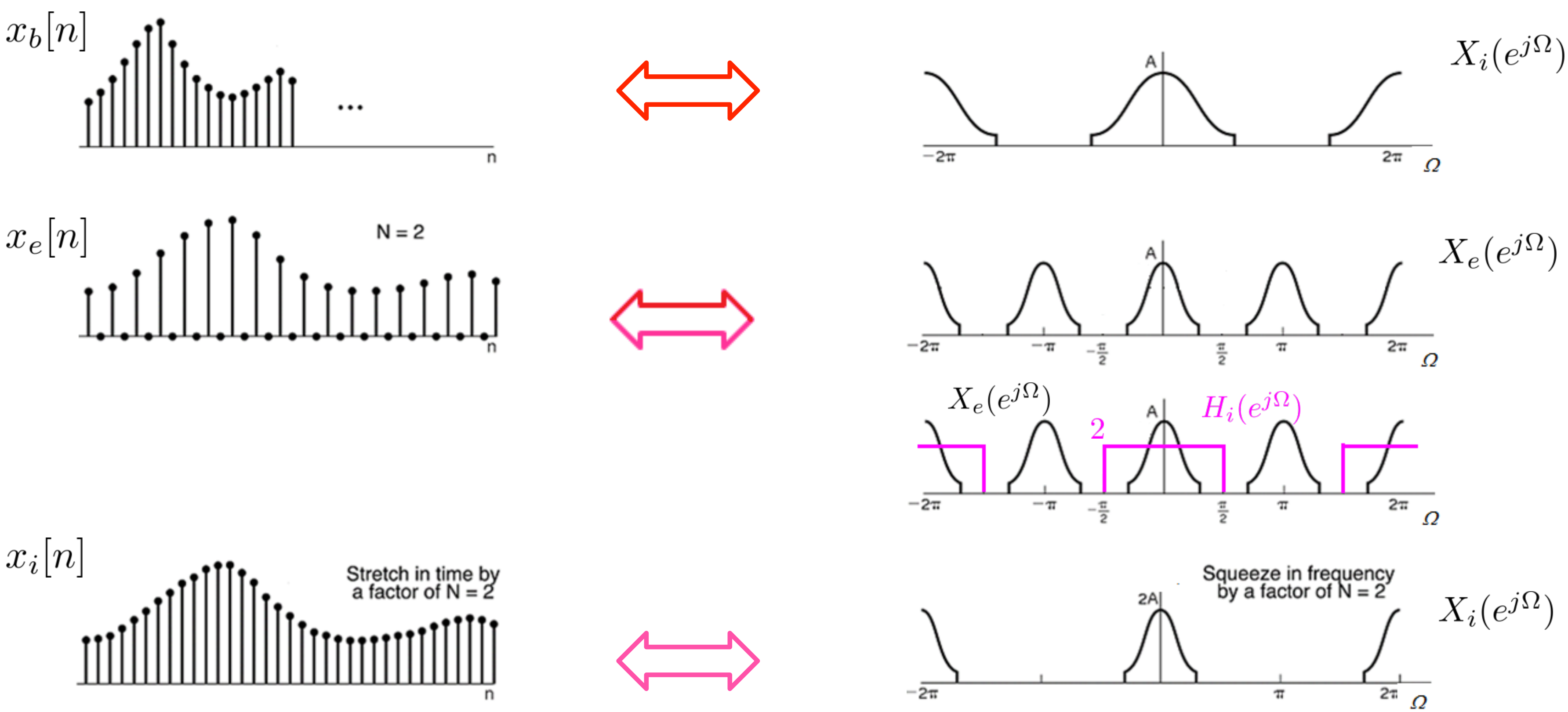


Vale cero en los múltiplos de N

- Podemos volver a utilizar interpoladores subóptimos (orden cero, lineal, sincs truncadas) ➔ Todo igual que en el caso continuo

Interpolación SD: gráficamente



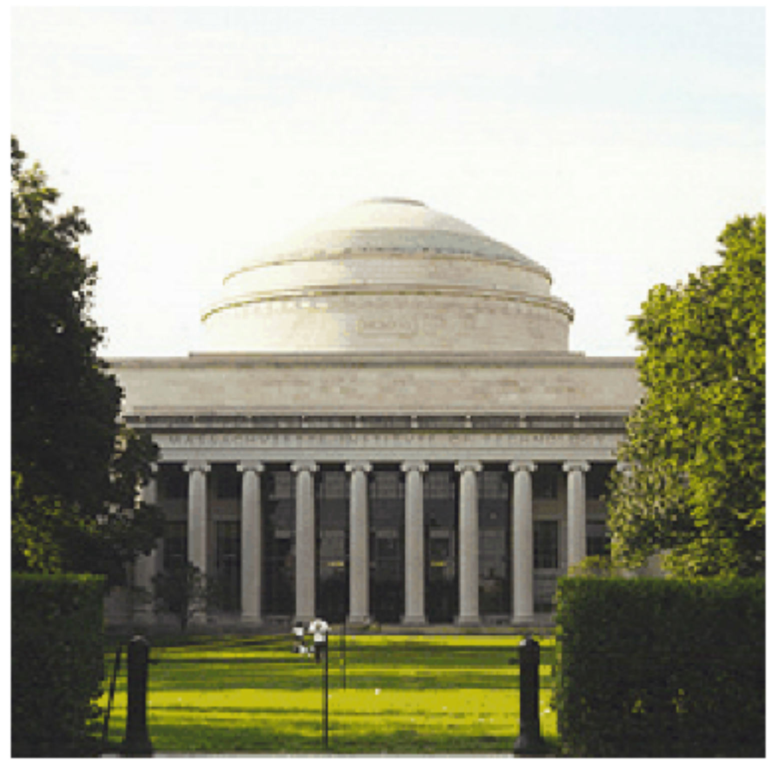


TIEMPO **FRECUENCIA**

Ejemplo de interpolación en imágenes

$\uparrow 2$

$\uparrow 2$



¿Qué estamos viendo en esta imagen? ¿Dónde está este edificio?