

FACULTAD
DE INGENIERIA
Universidad de Buenos Aires

Muestreo e interpolación en tiempo discreto

Dr. Ing. Pedro D. Arini

Señales y Sistemas
Facultad de Ingeniería
Universidad de Buenos Aires

2do. cuatrimestre 2026

Resumen

1 Teorema del muestreo

Resumen

1. Teorema del muestreo

2. Procesamiento digital de señales de tiempo discreto

Resumen

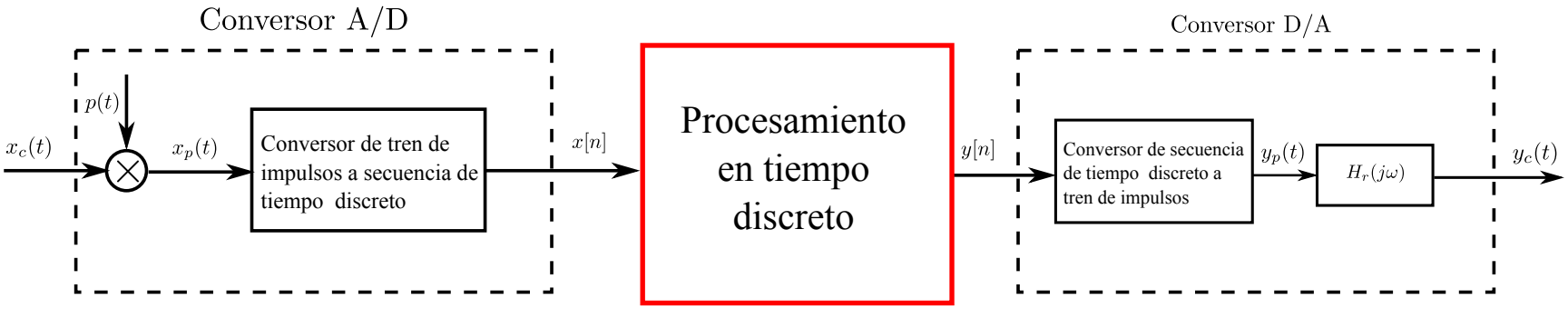
1. Teorema del muestreo
2. Procesamiento digital de señales de tiempo discreto
3. Muestreo e interpolación en tiempo discreto

Resumen

1. Teorema del muestreo
2. Procesamiento digital de señales de tiempo discreto
3. Muestreo e interpolación en tiempo discreto
4. Cuantificación

Procesamiento en tiempo discreto I

La posibilidad de poder muestrear una señal de tiempo continuo y poder volver a recuperar la señal original a partir de sus muestras permite que muchas operaciones de procesamiento de una señal se puedan realizar en tiempo discreto. Esto es:



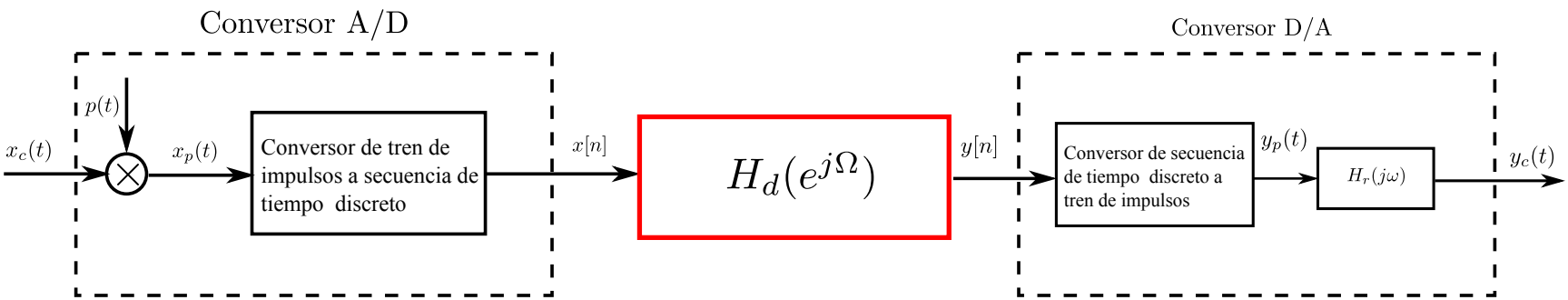
Con el advenimiento de procesadores más potentes, pequeños y baratos esta metodología de trabajo se ha vuelto algo estándar en ingeniería electrónica. De hecho, es muy díficil hoy en día encontrar aplicaciones o dispositivos donde el análisis y la extracción de información de una señal de tiempo continuo no se realice sobre una secuencia de tiempo discreto que resulta del muestreo de la señal de interés original!!

Procesamiento en tiempo discreto II

La etapa de procesamiento en tiempo discreto se puede modelar como un sistema de tiempo discreto. Dependiendo de la naturaleza del procesamiento a realizar dicho sistema puede ser no lineal y variante en el tiempo

Sin embargo, nosotros nos restringiremos al caso en el que el procesamiento en tiempo discreto se puede modelar como un sistema LTI!!

Dado que el sistema será LTI, lo podemos representar con su repuesta al impulso o lo que es lo mismo con su respuesta en frecuencia:



Será interesante analizar este sistema en cascada!!

Procesamiento en tiempo discreto III

Sabemos que la transformada de Fourier de $x[n]$ vale:

$$X\left(e^{j\Omega}\right) = \frac{1}{T} X_c\left(j\frac{\Omega}{T}\right) \quad \Omega \in [-\pi, \pi)$$

para el caso en el que no hay aliasing. De esta forma la señal $y[n]$ a la salida del sistema de procesamiento discreto se puede escribir como:

$$Y\left(e^{j\Omega}\right) = H_d\left(e^{j\Omega}\right) X\left(e^{j\Omega}\right) = \frac{1}{T} H_d\left(e^{j\Omega}\right) X_c\left(j\frac{\Omega}{T}\right) \quad \Omega \in [-\pi, \pi)$$

La señal de tiempo continuo $y_c(t)$ se puede escribir como:

$$Y_c(j\omega) = \frac{1}{T} H_d(e^{j\omega T}) H_r(j\omega) X_c(j\omega), \quad \omega \in \mathbb{R}$$

Vemos que en el caso en que la señal $x_c(t)$ es de banda limitada y el muestreo se realiza de acuerdo al teorema del muestreo para que no exista aliasing, tenemos que la cascada es equivalente a un sistema LTI con respuesta en frecuencia:

$$H_c(j\omega) = \frac{1}{T} H_d(e^{j\omega T}) H_r(j\omega)$$

Procesamiento en tiempo discreto IV

El caso particular en que $H_r(\omega)$ es el filtro interpolador ideal con frecuencia de corte ω_c tenemos que:

$$H_c(j\omega) = \begin{cases} H_d(e^{j\omega T}) & |\omega| < \omega_c \\ 0 & |\omega| > \omega_c \end{cases} \tag{1}$$

Nuevamente es claro que el filtro equivalente en tiempo continuo es el filtro de tiempo discreto $H_d(e^{j\Omega})$ con el típico escalamiento en frecuencia dado por:

$$\omega = \frac{\Omega}{T}$$

De esta forma partiendo de (1) y asumiendo que el filtro interpolador es el ideal y el muestreo no presenta aliasing, podemos diseñar el filtro de tiempo discreto $H_d(e^{j\Omega})$ a partir de la respuesta en frecuencia total del sistema deseada proporcionada por $H_c(j\omega)$!!

Procesamiento en tiempo discreto V

Ejemplo: Supongamos que tenemos un sistema de muestreo donde el filtro $H_r(j\omega)$ corresponde al retenedor de orden cero. Quisiéramos diseñar el filtro $H_d(e^{j\Omega})$ para precompensar al distorsion introducida por dicho retenedor. Vamos a partir de:

$$H_c(j\omega) = \frac{1}{T} H_d(e^{j\omega T}) H_r(j\omega)$$

Buscamos entonces que:

$$H_c(j\omega) = \begin{cases} 1 & |\omega| < \frac{\omega_s}{2} \\ 0 & |\omega| > \frac{\omega_s}{2} \end{cases}$$

Notar que, si $X_c(j\omega)$ es de banda limitada a W podemos pedir que la frecuencia de corte del filtro $H_c(j\omega)$ esté en cualquier valor que satisfaga $W \leq \omega_c \leq \omega_s - W$, asumiendo que $\omega_s > 2W$. En este caso hemos elegido la frecuencia de corte igual a la mitad de la frecuencia de sampling.

Sabemos que:

$$H_r(j\omega) = T \text{sinc}\left(\frac{\omega T}{2\pi}\right) e^{-j\omega T/2}$$

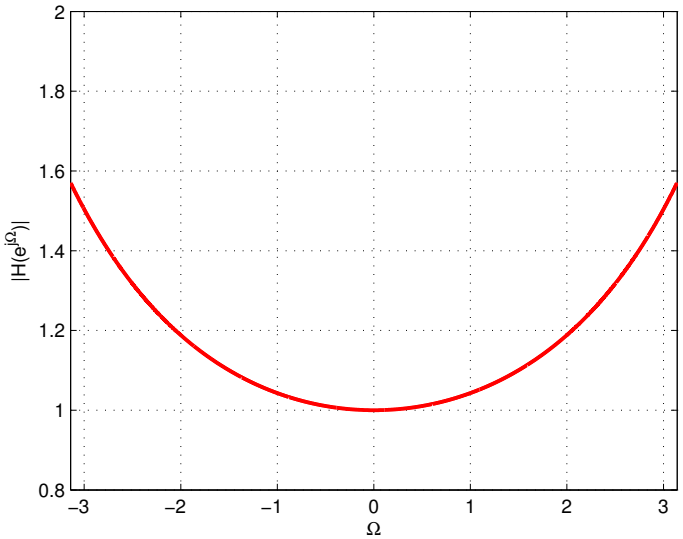
Procesamiento en tiempo discreto VI

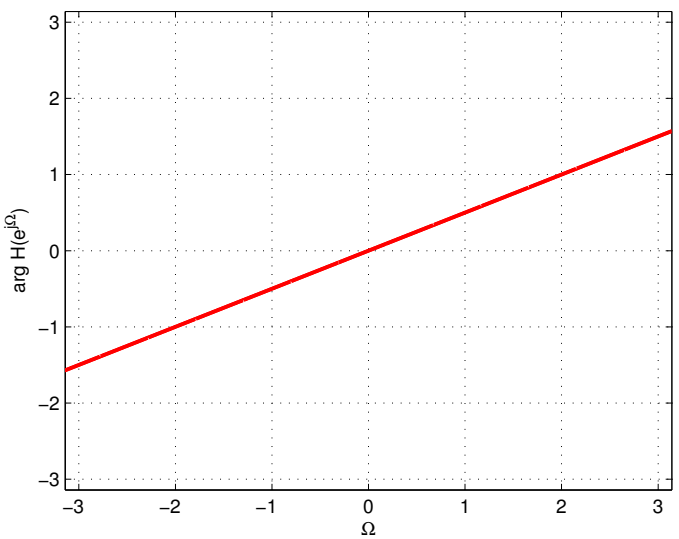
Podemos escribir entonces:

$$H_d(e^{j\omega T}) = \begin{cases} \dfrac{e^{j\omega T/2}}{\text{sinc}(\frac{\omega T}{2\pi})} & |\omega| < \frac{\omega_s}{2} \\ 0 & |\omega| > \frac{\omega_s}{2} \end{cases}$$

Expresando todo en función de la frecuencia de tiempo discreto Ω tenemos que el sistema de tiempo discreto pre-compensandor que hay que ubicar en la cascada tiene respuesta en frecuencia:

$$H_d(e^{j\Omega}) = \frac{e^{j\Omega/2}}{\text{sinc}\left(\frac{\Omega}{2\pi}\right)}, \ \Omega \in [-\pi, \pi)$$





Procesamiento en tiempo discreto VII

Ejemplo: Supongamos que deseamos implementar un sistema que retrase una señal de tiempo continuo $x_c(t)$ en una cantidad Δ. La respuesta en frecuencia del sistema en tiempo continuo a implementar es:

$$H_c(j\omega) = e^{-j\omega\Delta}$$

Dado que la señal a procesar será de banda limitada W y asumiendo que el filtro $H_r(j\omega)$ es un pasabajo ideal con frecuencia de corte $W < \omega_c < \omega_s - W$ y ganancia T tenemos que la respuesta en frecuencia de tiempo continuo deseada la podemos escribir como:

$$H_c(j\omega) = \begin{cases} e^{-j\omega\Delta} & |\omega| < \omega_c \\ 0 & |\omega| > \omega_c \end{cases}$$

Como $H_c(j\omega) = \frac{1}{T} H_d(e^{j\omega T}) H_r(j\omega)$ tenemos que:

$$H_d(e^{j\omega T}) = \begin{cases} e^{-j\omega\Delta} & |\omega| < \omega_c \\ 0 & |\omega| > \omega_c \end{cases}$$

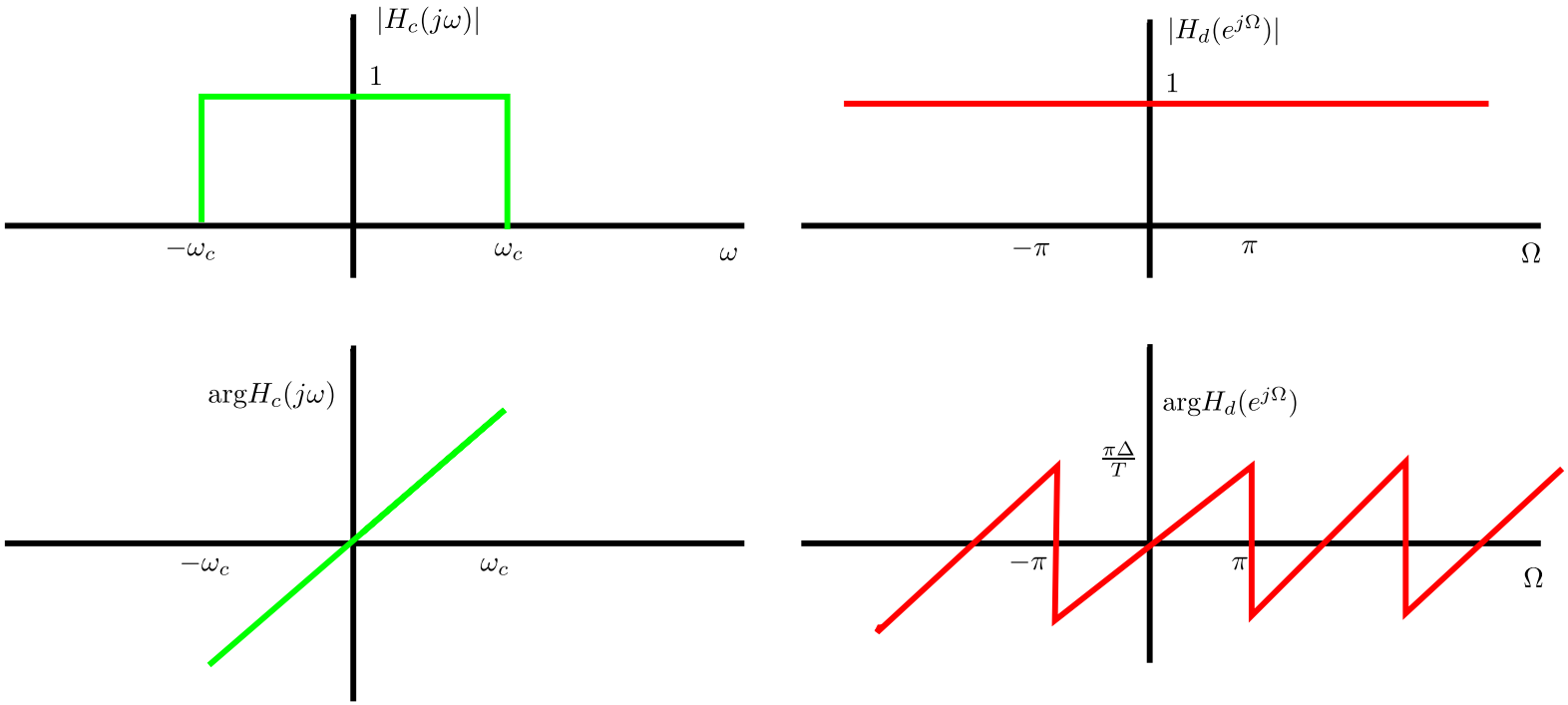
De esta forma el filtro de tiempo discreto a implementar es:

$$H_d(e^{j\Omega}) = \begin{cases} e^{-j\Omega\Delta/T} & |\Omega| < \omega_c T \\ 0 & |\Omega| > \omega_c T \end{cases}$$

Procesamiento en tiempo discreto VIII

En el caso especial de que $\omega_c = \frac{\omega_s}{2}$ tenemos que:

$$H_d(e^{j\Omega}) = e^{-j\Omega\Delta/T} \quad |\Omega| < \pi$$



Vemos las diferencias y simulitudes entre el filtro de tiempo continuo deseado y la implementación del filtro de tiempo discreto equivalente!!

Procesamiento en tiempo discreto IX

De esta forma si la entrada es $x_c(t)$ (y obviamente de banda limitada) la salida al sistema completo será $y_c(t) = x_c(t - \Delta)$. En el caso particular que $x_c(t) = \frac{1}{T}\text{sinc}\left(\frac{t}{T}\right)$ tenemos que $x[n] = \frac{1}{T}\delta[n]$. Además:

$$y_c(t) = \frac{1}{T}\text{sinc}\left(\frac{t-\Delta}{T}\right)$$

Es claro que $y[n]$ son las muestras de $y_c(t)$ tomadas a una tasa T. Entonces:

$$y[n] = \frac{1}{T}\text{sinc}\left(n - \frac{\Delta}{T}\right)$$

Entonces es claro que la respuesta al impulso del sistema $H_d(e^{j\Omega})$ vale:

$$h[n] = \text{sinc}\left(n - \frac{\Delta}{T}\right)$$

Si $\frac{\Delta}{T}$ es un entero es claro que $h[n] = \delta\left[n - \frac{\Delta}{T}\right]$. Entonces la operación de retardo de tiempo continuo deseada se puede implementar con un retardo en tiempo discreto. Sin embargo, si $\frac{\Delta}{T}$ no es un entero, la operación de retardo de tiempo continuo deseada no se traduce en una operación de retardo en tiempo discreto!! Cuál es la interpretación de un retardo $\frac{\Delta}{T}$ no entero??

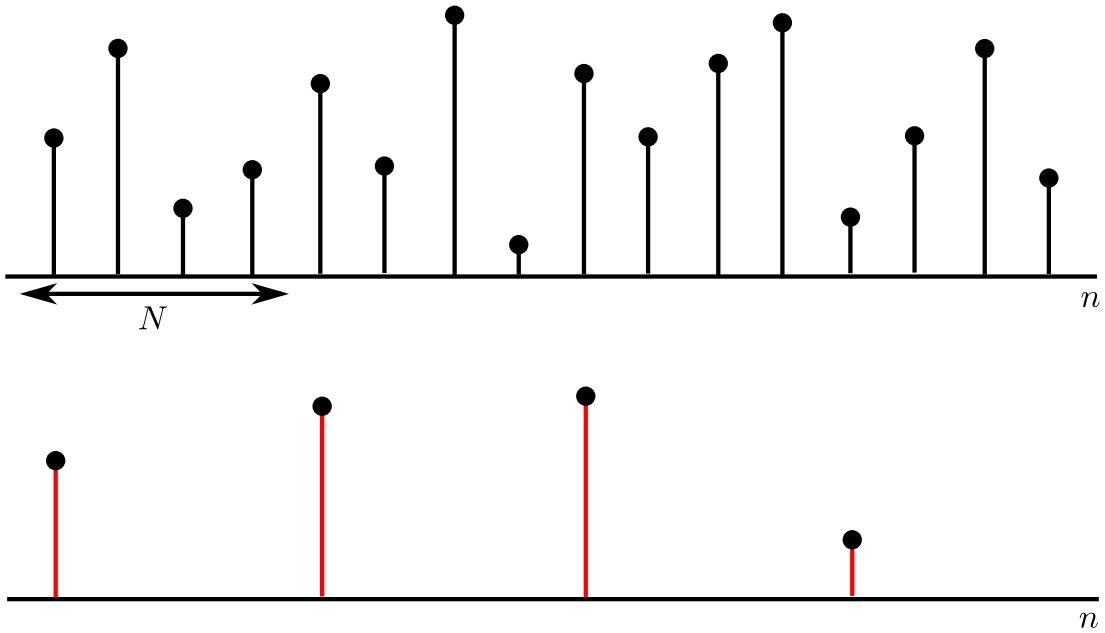
Muestreo en tiempo discreto I

Hasta el momento hemos considerado el muestreo de señales de tiempo continuo. Pero también podemos considerar el muestreo de señales de tiempo discreto.

El muestreo de señales de tiempo discreto obedece a los mismos principios que las señales de tiempo continuo y las ideas de fondo son básicamente las mismas!!

Dada una señal de tiempo discreto $x[n]$ consideramos la siguiente señal:

$$x_p[n] = \begin{cases} x[n] & \text{si } n \text{ es múltiplo de } M \\ 0 & \text{en otro caso} \end{cases}$$



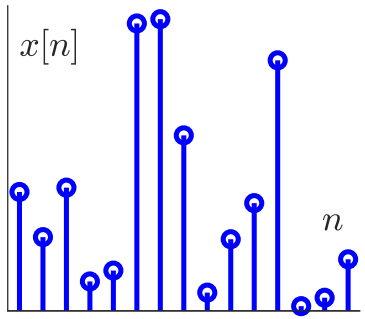
Muestreo en tiempo discreto II

Podemos escribir:

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

$$x_p[n] = x[n]p[n]$$

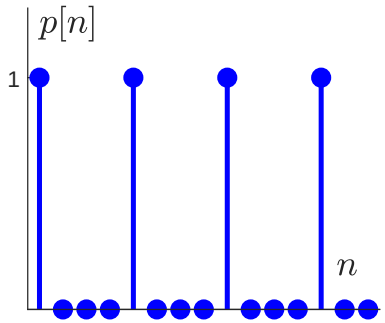
$$= \sum_{k=-\infty}^{\infty} x[kN]\delta[n - kN]$$



En el dominio de la frecuencia:

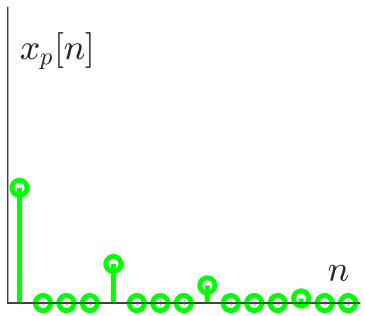
$$X_p(\Omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} P(\theta) X(\Omega - \theta) d\theta$$

$$= \frac{1}{2\pi} \int_{0}^{2\pi} P(\theta) X(\Omega - \theta) d\theta$$



Donde:

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



donde $\Omega_s = \frac{2\pi}{N}$ es la frecuencia de muestreo.

Muestreo en tiempo discreto III

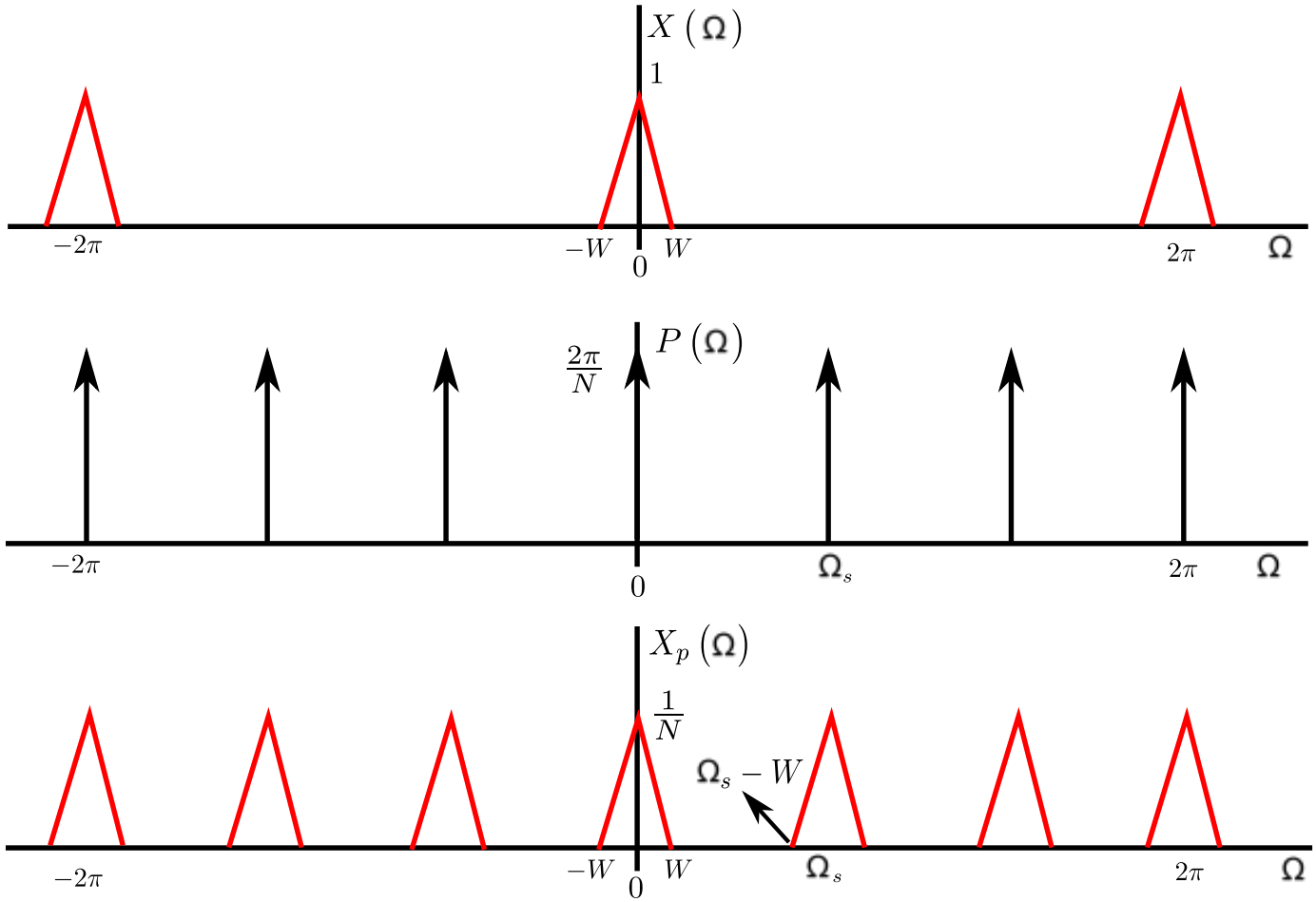
Podemos escribir:

$$\begin{aligned} X_p(\Omega) &= \frac{1}{2\pi}\int_0^{2\pi} P(\theta)\, X(\Omega-\theta)\, d\theta \\ &= \frac{1}{2\pi}\int_0^{2\pi} \frac{2\pi}{N}\sum_{k=-\infty}^{\infty}\delta\left(\theta-\frac{2\pi k}{N}\right) X(\Omega-\theta)\, d\theta \\ &= \frac{1}{N}\sum_{k=-\infty}^{\infty}\int_0^{2\pi}\delta\left(\theta-\frac{2\pi k}{N}\right) X(\Omega-\theta)\, d\theta \end{aligned}$$

Notar que sólo los impulsos con $0 \leq k < N$ quedan dentro del intervalo $[0, 2\pi)$. Entonces:

$$\begin{aligned} X_p(\Omega) &= \frac{1}{N}\sum_{k=0}^{N-1}\int_0^{2\pi}\delta\left(\theta-\frac{2\pi k}{N}\right) X(\Omega-\theta)\, d\theta \\ &= \frac{1}{N}\sum_{k=0}^{N-1} X\left(\Omega-\frac{2\pi k}{N}\right) \\ &= \frac{1}{N}\sum_{k=0}^{N-1} X(\Omega-k\Omega_s), \quad \Omega\in[-\pi,\pi) \end{aligned}$$

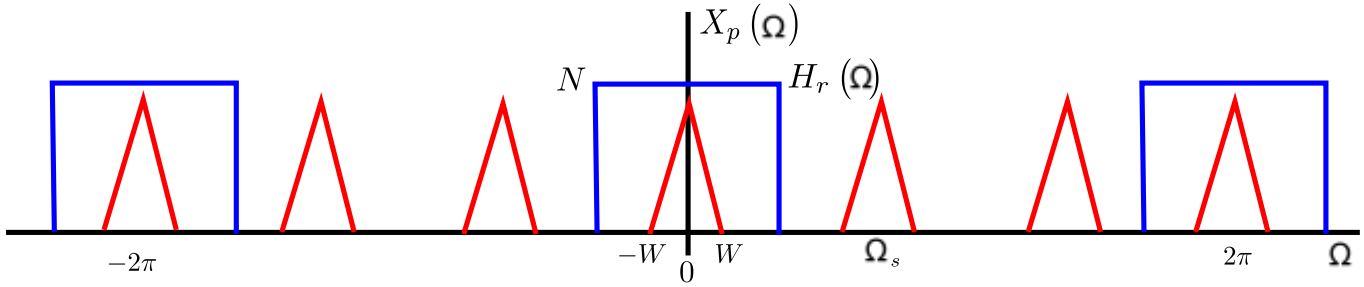
Muestreo en tiempo discreto IV



Vemos que si la señal es banda limitada a W no existe aliasing si $W < \Omega_s - W$, o lo que es lo mismo si $2W < \Omega_s$!! Obviamente si $2W \geq \Omega_s$ tenemos nuevamente aliasing!!

Muestreo en tiempo discreto V

Supongamos que $X(\Omega) = 0$ para $|\Omega| > W$ y si $2W < \Omega_s$. Sea $H_r(\Omega)$ un filtro pasabajos ideal con frecuencia de corte $W \leq \Omega_c \leq \Omega_s - W$ y ganancia N.



Es claro que:

$$X_p(\Omega) H_r(\Omega) = \frac{1}{N} \sum_{k=0}^{N-1} X(\Omega - k\Omega_s) H_r(\Omega) = X(\Omega)$$

Recuperamos la señal original $x[n]$!! En el caso en el que hay solapamiento esto no sería posible, ya que dicho solapamiento altera el espectro original de $x[n]$!!

Muestreo en tiempo discreto VI

Teorema

Sea $x[n]$ una señal de banda limitada que cumple:

$$X(\Omega) = 0 \ |\Omega| \geq W$$

La señal $x[n]$ queda determinada en forma única por las muestras $x[nN], n \in \mathbb{Z}$ si la tasa de muestreo N satisface:

$$\Omega_s = \frac{2\pi}{N} > 2W, \quad N < \frac{\pi}{W}$$

- Notar que si $W > \frac{\pi}{2}$ no será posible muestrear una función de tiempo discreto sin tener aliasing!
- Este teorema revela una condición suficiente para reconstruir una señal de banda limitada a partir de sus muestras. Pero si uno impone más restricciones a las señales $x[n]$ el teorema puede dejar de ser condición necesaria (por supuesto sigue siendo condición suficiente).
- Si $x[n]$ proviene del muestreo de una señal de banda limitada de tiempo continuo con período de muestreo T, el muestrear $x[n]$ con período N es equivalente a haber muestreado la señal $x_c(t)$ con período NT (sin considerar los ceros en $x_p[n]$!!) y no habrá aliasing si el ancho de banda de $x_c(t)$ es menor a $\frac{\pi}{NT}$.

Reconstrucción en tiempo discreto

Podemos analizar la ecuación de reconstrucción para el caso de muestreo de tiempo discreto. Sea $x[nN]$ la señal muestreada. La respuesta al impulso del filtro pasa-bajos ideal $H_r(\Omega)$ con frecuencia de corte Ω_c y ganancia N vale:

$$h_r(n) = \frac{N\Omega_c}{\pi} \frac{\sin(\Omega_c n)}{\Omega_c n}$$

La secuencia reconstruida vale $x_r[n] = x_p[n] * h_r[n]$ o lo que es lo mismo:

$$x_r[n] = \sum_{k=-\infty}^{\infty} x[kN] \frac{N\Omega_c}{\pi} \frac{\sin(\Omega_c(n-kN))}{\Omega_c(n-kN)}$$

Notar que $x_r[nN] = x[nN]\ \forall n \in \mathbb{Z}$. Es más:

Por el análisis en frecuencia que hicimos sabemos que de hecho $x_r[n] = x[n]\ \forall n \in \mathbb{Z}$!!

Cuando en lugar del filtro pasa-bajos ideal usamos otro filtro reconstructor tenemos:

$$x_r[n] = \sum_{k=-\infty}^{\infty} x[kN] h_r[n-kN]$$

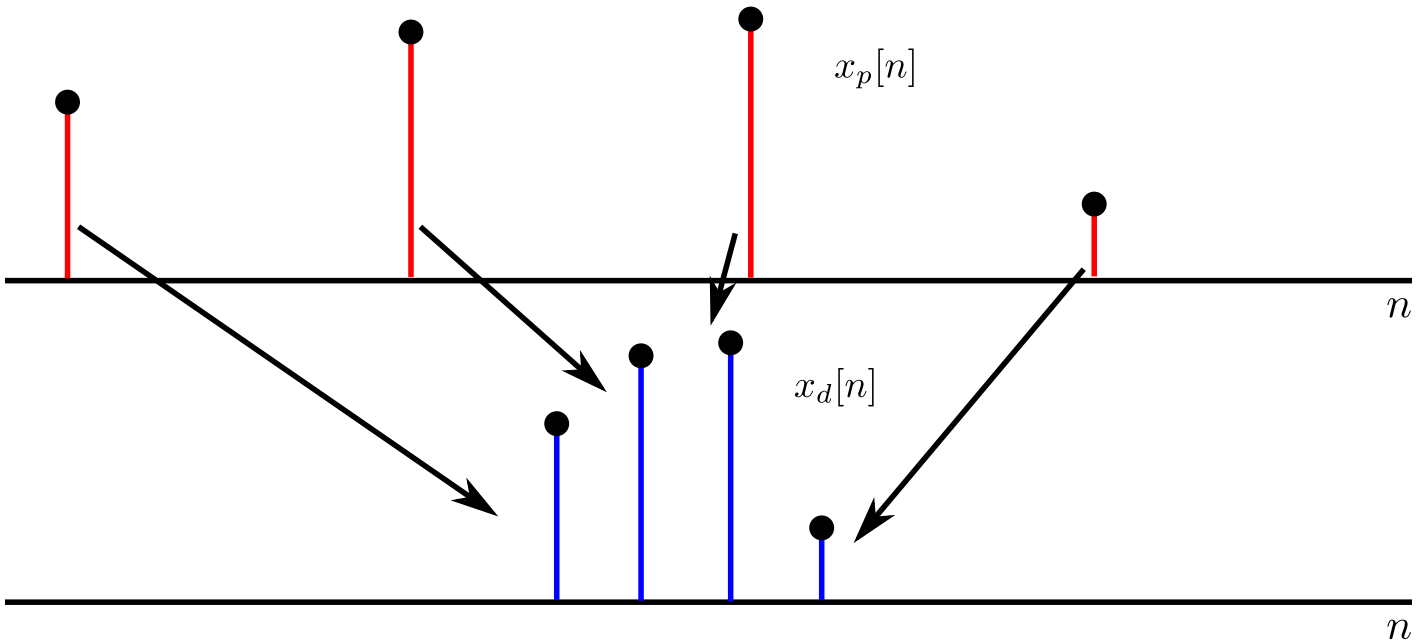
Decimación y expansión en tiempo discreto I

En el caso de la señal muestreada $x_p[n]$ vemos que por cada muestra existen $N-1$ ceros. Es claro que si N está bien elegido, toda la información de la señal original $x[n]$ está contenida sólo en las muestras y no en los ceros.

Transmitir o almacenar toda la secuencia $x_p[n]$ es altamente ineficiente, especialmente si N es grande!!

Entonces, en función de esta observación vamos a considerar la siguiente secuencia, en la cual quitamos los ceros entre muestras:

$$x_d[n] = x_p[nN] = x[nN], \ \forall n \in \mathbb{Z}$$



Decimación y expansión en tiempo discreto II

Podemos escribir:

$$X_d\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} x_d[n]e^{-j\omega n} = \sum_{n=-\infty}^{\infty} x[nN]e^{-j\omega n}$$

Sea $k = nN$, o lo que es lo mismo $n = \frac{k}{N}$ entonces podemos escribir:

$$X_d\left(e^{j\omega}\right) = \sum_{k:\text{múltiplos de } N} x[k]e^{-j\omega k/N} = \sum_{k=-\infty}^{\infty} x_p[k]e^{-j\omega k/N}$$

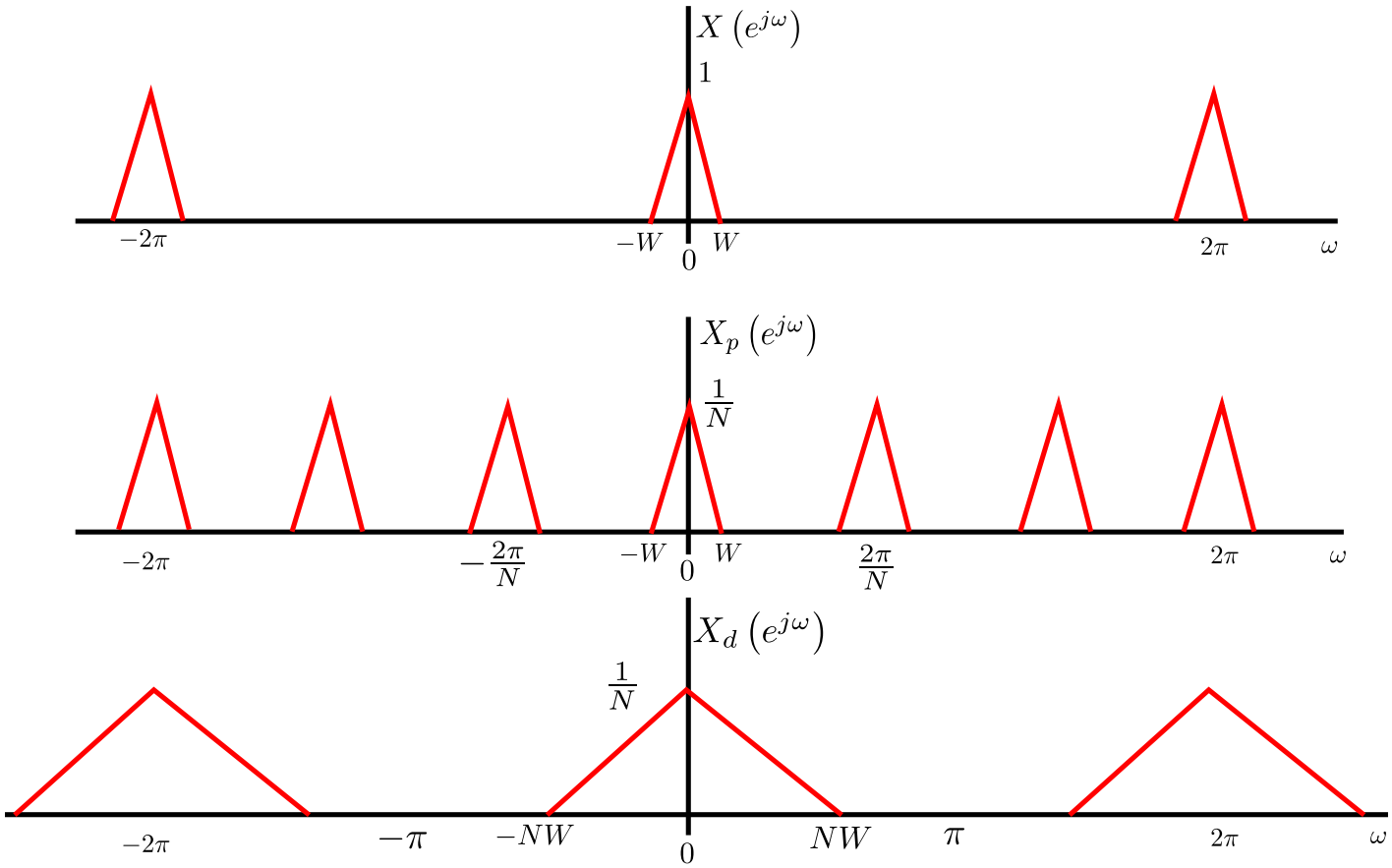
ya que $x_p[k]$ vale cero para valores de k que no son múltiplos de N. Entonces:

$$X_d\left(e^{j\omega}\right) = X_p\left(e^{j\frac{\omega}{N}}\right)$$

Y finalmente:

$$X_d\left(e^{j\omega}\right) = \frac{1}{N}\sum_{k=0}^{N-1} X\left(e^{j\left(\frac{\omega}{N} - \frac{2\pi k}{N}\right)}\right)$$

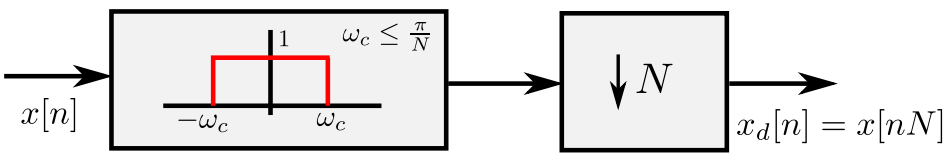
Decimación y expansión en tiempo discreto III



Si el espectro original $X(e^{j\omega})$ es de banda limitada y N es elegido de forma tal que no haya aliasing, el efecto de quitar los ceros (*decimación*), es expandir el espectro de la secuencia original sobre una banda mayor de frecuencias!!

Decimación y expansión en tiempo discreto IV

- Si la secuencia $x[n]$ proviene del muestreo de una señal de tiempo continuo $x_c(t)$, el proceso de decimación, debido a que remueve los ceros de $x_p[n]$, puede verse como la reducción efectiva de la velocidad de muestreo T de la señal continua $x_c(t)$ por un factor N, de forma tal que la velocidad de muestreo total es NT.
- Es claro que para poder decimar por algún factor de N apropiado ($N > 1$) y no tener aliasing, la señal $x_c(t)$ debe haber sido muestreada con una frecuencia $\omega_s > 2NW$ donde W es el ancho de banda de $x_c(t)$ Por qué??
- Si tengo una señal $x[n]$ cuyo ancho de banda no permite la decimación por un factor N de interés sin evitar el aliasing, para poder decimar se necesitará reducir el ancho de banda de $x[n]$. Para ello se puede usar un filtro pasa-bajos ideal de tiempo discreto con ganancia unitaria y frecuencia de corte $\omega_c \leq \frac{\pi}{N}$ que filtre las frecuencias de $x[n]$ por encima de ω_c. Por supuesto que en este caso de $x_d[n]$ no vamos a poder recuperar $x[n]$!! A pesar ello esto puede ser útil en algunas situaciones.

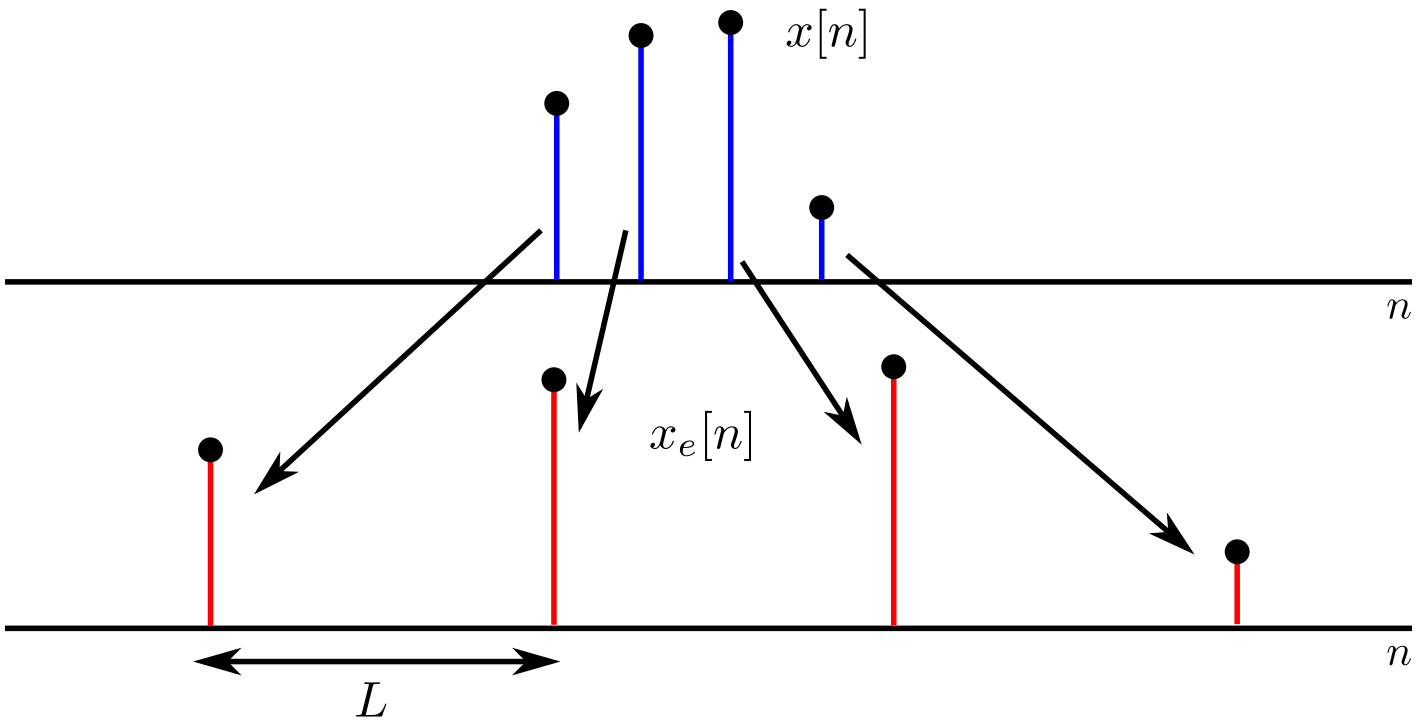


Esta estructura se denomina *decimador*. El mismo decima por un factor de N. En general, aún cuando la señal $x[n]$ tenga banda limitada a $\frac{\pi}{N}$ la etapa de filtrado pasa-bajos estará presente!!

Decimación y expansión en tiempo discreto V

Vimos, entonces, como la decimación de una señal de tiempo discreto es equivalente al muestreo de una señal de tiempo continuo. Consideremos ahora la posibilidad de "aumentar" la frecuencia de muestreo de una señal $x[n]$. Consideremos la siguiente señal:

$$x_e[n] = \begin{cases} x\left[n/L\right] & n \text{ es múltiplo de } L \\ 0 & \text{en otro caso} \end{cases}$$



Decimación y expansión en tiempo discreto VI

Es claro que podemos escribir:

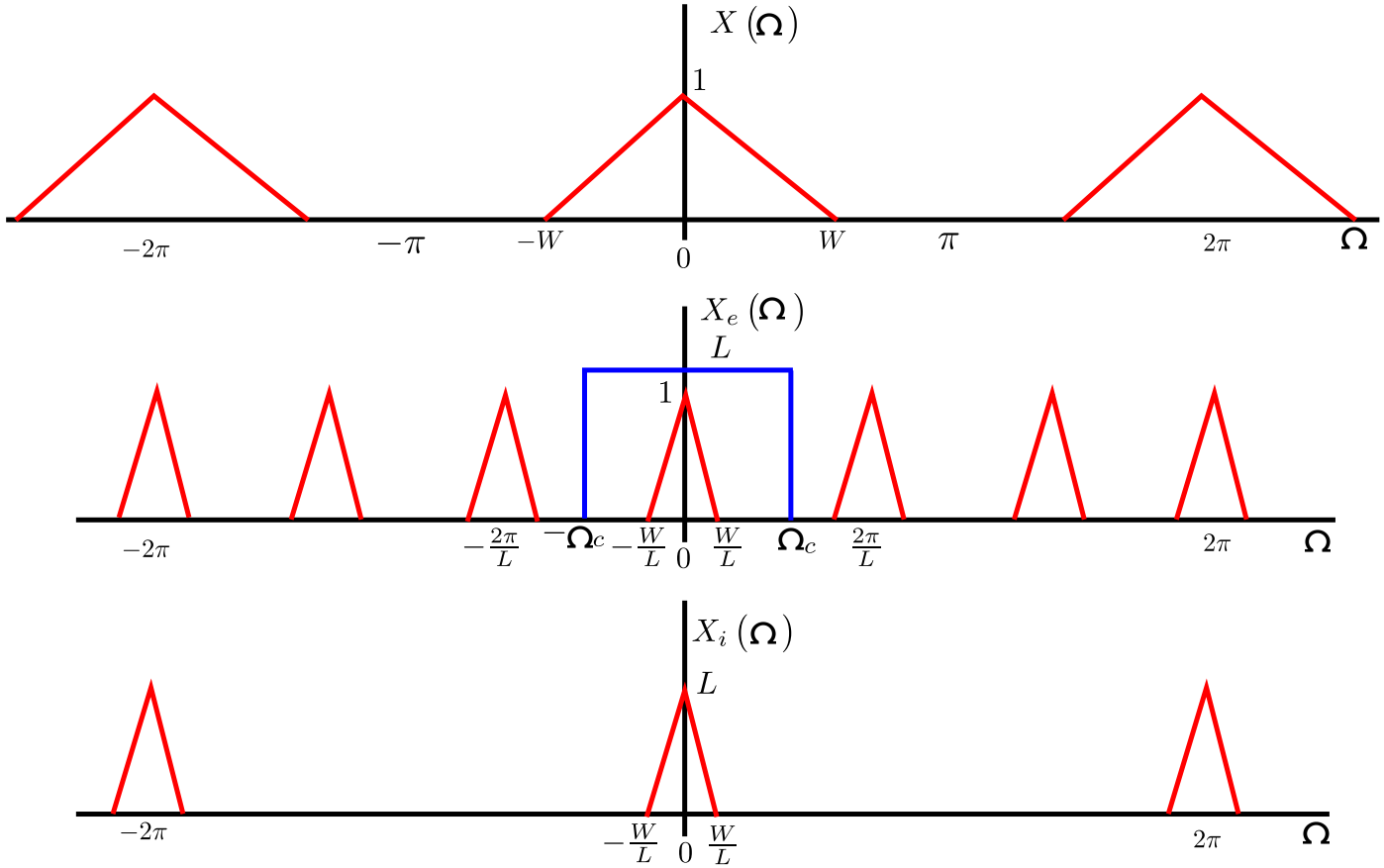
$$x_e[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n - kL]$$

La transformada de Fourier de $x_e[n]$ puede calcularse como:

$$\begin{aligned} X_e(\Omega) &= \sum_{n=-\infty}^{\infty} \left(\sum_{k=-\infty}^{\infty} x[k]\delta[n - kL] \right) e^{-j\Omega n} \\ &= \sum_{k=-\infty}^{\infty} x[k] \sum_{n=-\infty}^{\infty} \delta[n - kL] e^{-j\Omega n} \\ &= \sum_{k=-\infty}^{\infty} x[k] e^{-j\Omega kL} \\ &= X(\Omega L) \end{aligned}$$

La salida del *expansor* (el dispositivo que agrega ceros) es tal que su transformada de Fourier es la transformada de la entrada con una compresión en el eje de la frecuencia por un factor L!!

Decimación y expansión en tiempo discreto VII



Luego de la expansión en el tiempo, a través de un filtrado pasa-bajos, con ganancia L y frecuencia de corte $\omega_c \leq \frac{\pi}{L}$, podemos generar la señal $x_i[n]$!!

Decimación y expansión en tiempo discreto VIII

El filtro pasa-bajos tiene respuesta al impulso:

$$h_r[n] = \frac{\Omega_c L}{\pi} \frac{\sin(\Omega_c n)}{\omega_c n}$$

Entonces la señal $x_i[n]$ puede escribirse:

$$\begin{aligned} x_i[n] =& x_e[n] * h_r[n] \\ &= \left(\sum_{k=-\infty}^{\infty} x[k]\delta[n-kL] \right) * h_r[n] \\ &= \sum_{k=-\infty}^{\infty} x[k] \Big(\delta[n-kL] * h_r[n] \Big) \\ &= \sum_{k=-\infty}^{\infty} x[k] \frac{\Omega_c L}{\pi} \frac{\sin(\Omega_c(n-kL))}{\Omega_c(n-kL)} \end{aligned}$$

Es claro que:

$$\begin{aligned} h_r[0] &= 1 \\ h_r[kL] &= 0, \quad k \in \mathbb{Z} \end{aligned}$$

Decimación y expansión en tiempo discreto IX

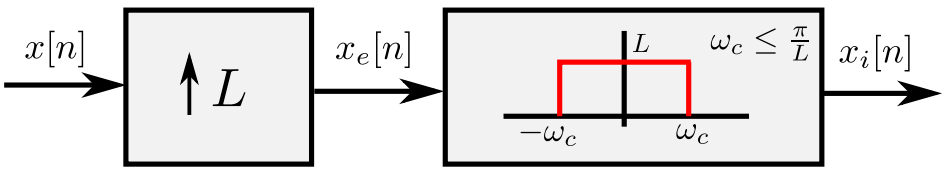
De esta forma tenemos que:

$$x_i[n] = x[n/L], \quad n = 0, \pm L, \pm 2L, \ldots$$

Si $x[n]$ proviene del muestreo (sin aliasing) de una señal de banda limitada en tiempo continuo $x_c(t)$, tenemos que:

$$x_i[n] = x_c\left(n\frac{T}{L}\right), \quad n = 0, \pm L, \pm 2L, \ldots$$

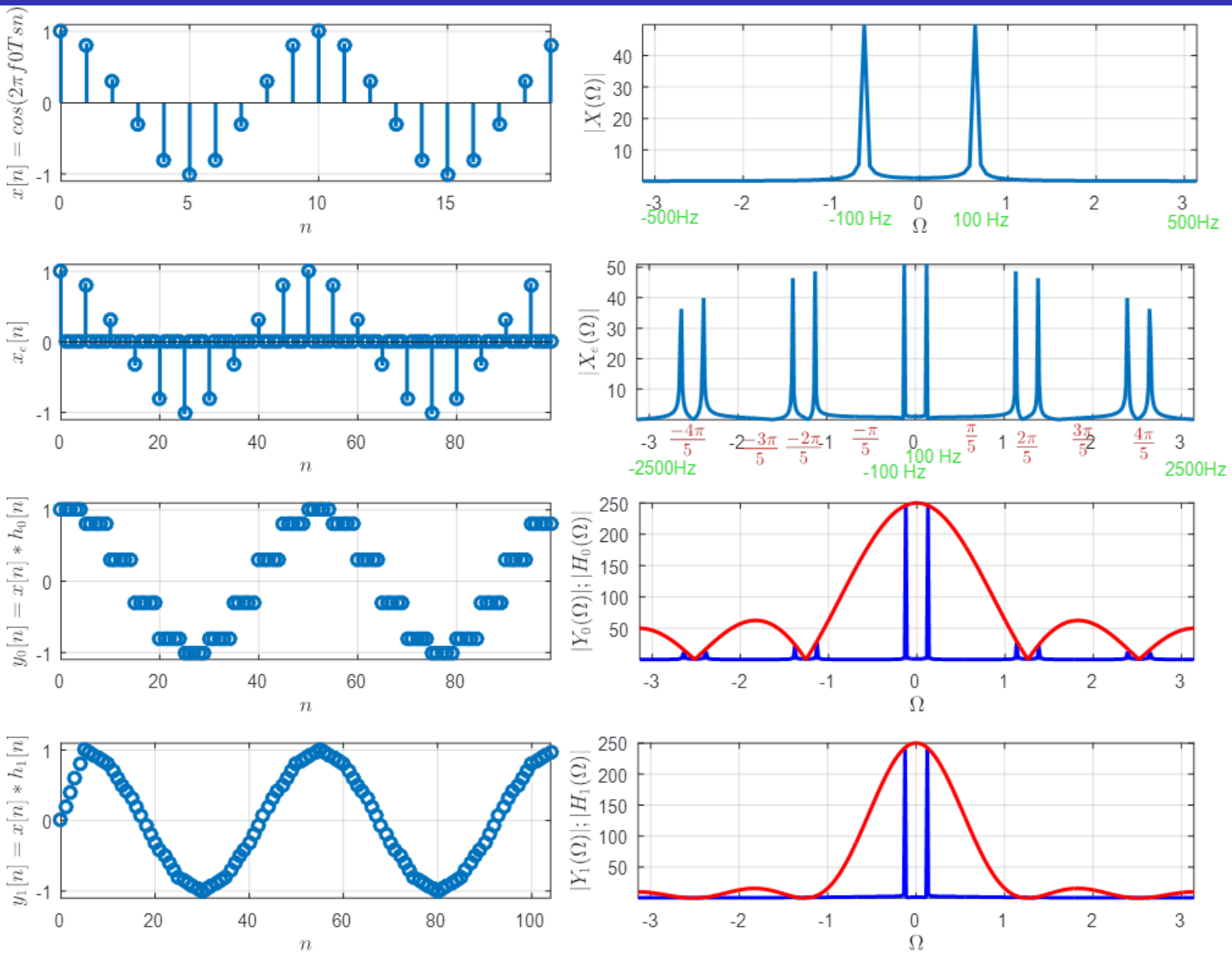
Si $x[n]$ proviene del muestreo sin aliasing de una señal $x_c(t)$ tenemos que la operación de expansión seguida de un filtrado pasa-bajos adecuado es equivalente a un muestreo de la señal $x_c(t)$ a una frecuencia $L\Omega_s$!! Del análisis en la frecuencia hecho tenemos que $x_i[n] = x_c\left(n\frac{T}{L}\right)$ para todo $n \in \mathbb{Z}$!!



Esta estructura se denomina *expansor*. El mismo expande o sobremuestrea por un factor de L.

Así como en un conversor D/A podíamos usar otros filtros interpoladores además del interpolador de banda limitada, aquí también podemos analizar la implementación de otros filtros interpoladores en el expansor!!

Decimación y expansión en tiempo discreto X



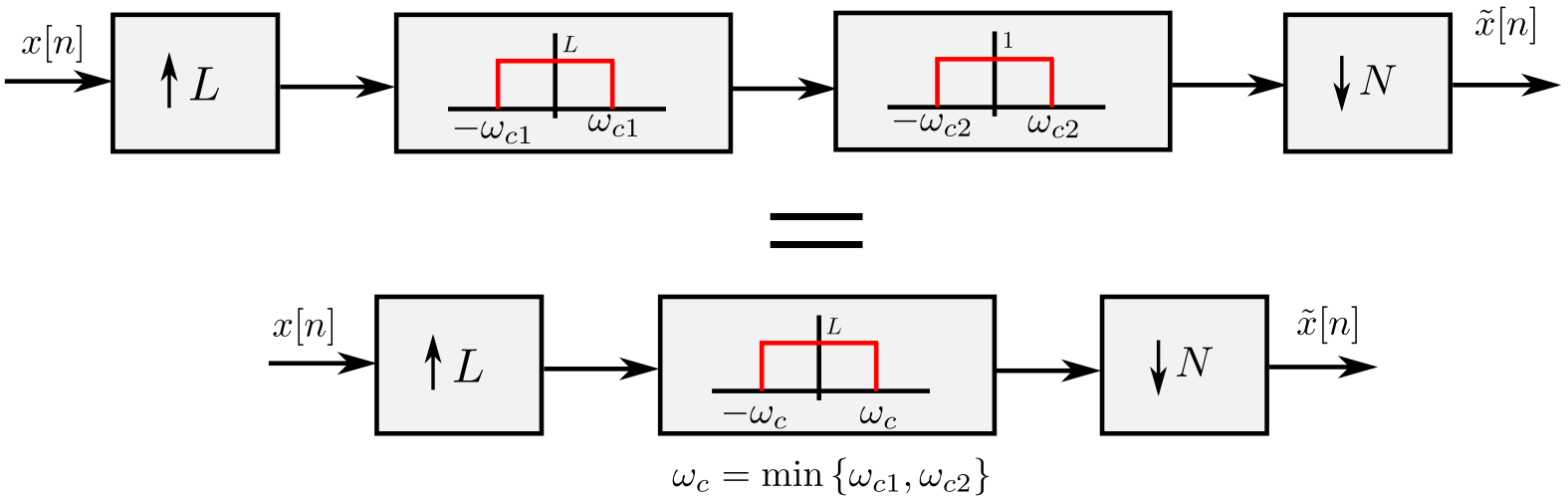
Decimación y expansión en tiempo discreto XI

Si $x[n] = \cos(2\pi f_0 T_s n)$, se obtuvo por muestreo de una señal coseno, cumpliendo con el teorema de Nyquist, responda:

- Observando el primer gráfico, ¿cual es el período N de la señal $x[n]$?
- Observando el segundo gráfico:
  - ¿Cual era la frecuencia del coseno $x(t)$?
  - ¿Cual es la frecuencia del coseno $x[n]$?
  - ¿Cual fue la frecuencia de muestreo?
  - Dado que el módulo de la TF de una señal coseno son dos $\delta(\Omega)$ en Ω_0 ¿ porqué no se observa esto en la gráfica?
- Observando los otros gráficos:
  - ¿Cuál fue el factor de expansión L usado?
  - Al comienzo de la figura 7 se observa una pendiente lineal y no la forma esperada del coseno. Explique.

Decimación y expansión en tiempo discreto X

Lo hecho hasta ahora nos permite también generar una forma de cambiar la frecuencia de muestreo de una señal de banda limitada $x_c(t)$ por cualquier factor racional $\frac{L}{N}$ haciendo procesamiento en tiempo discreto y sin necesidad en ningún momento de volver en tiempo continuo. Si tenemos $x[n] = x_c(nT)$, con $\omega_s = \frac{2\pi}{T}$ podemos generar $\tilde{x}[n] = x_c\left(n\frac{NT}{L}\right)$ usando la siguiente estructura

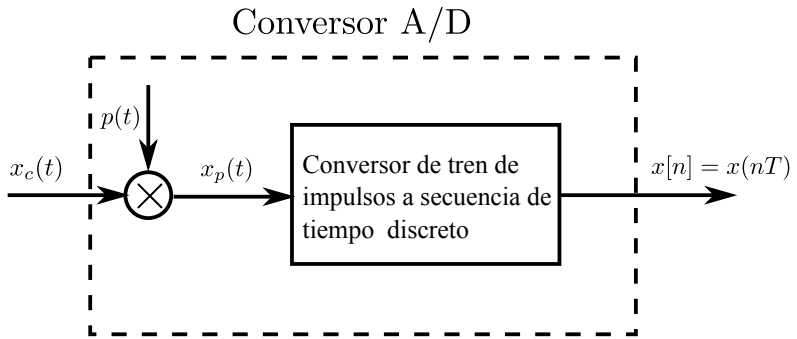


$$\omega_c = \min\{\omega_{c1}, \omega_{c2}\}$$

- Los cambios de la frecuencia de muestreo de una señal continua usando decimadores y expansores se conocen como *resampleos*.
- Estas operaciones pueden ser computacionalmente costosas, especialmente si L y N son grandes. Mediante el concepto de descomposición polifásica se puede reducir el costo computacional en estos casos.
- El orden expansión-decimación no es igual a decimación-expansión.

Cuantificación I

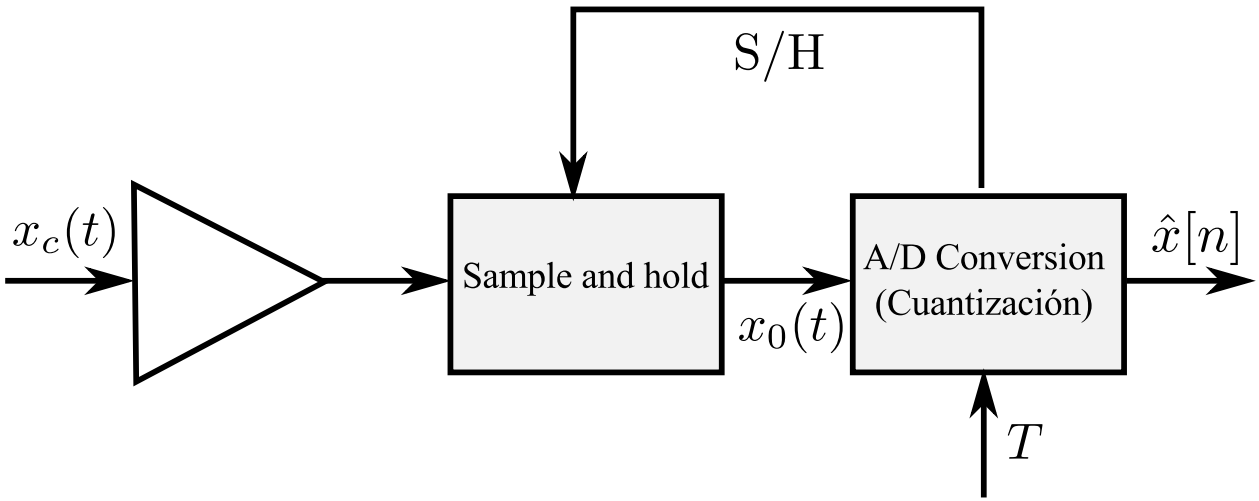
Hasta el momento hemos descripto el muestreo de señales considerando que los conversores A/D tienen la siguiente estructura ideal:



Como mencionamos anteriormente esto es simplemente un modelo matemático de un conversor ideal que es útil para el análisis de las principales características del muestreo en el dominio de la frecuencia.

- Claramente el proceso de tomar tomar la muestra $x_c(nT)$ en la práctica no se hace con un tren de impulsos.
- En general la muestra $x_c(nT)$ es un número real con infinitos decimales. En la práctica, en una computadora, sólo podemos representar números con finitos decimales.

Cuantificación II



- En cada intervalo T, un comando de S/H es enviado desde el módulo de cuantificación.
- En ese punto el circuito de sample and hold mantiene el valor instántaneo de la señal de entrada $x_c(t)$ durante una duración T.
- Gracias a que el valor de la entrada permanece constante, el módulo de cuantificación puede transformar el valor de la entrada en una descripción binaria con b bits.
- Cuando el módulo de cuantificación terminó su conversión (luego de T segundos) puede enviar otro comando S/H al circuito de sample and hold y el proceso vuelve a comenzar.

Cuantificación III

Notar que:

$$x_0(t) = \sum_{n=-\infty}^{\infty} x_c(nT)h_0(t-nT) = h_0(t) * \left(\sum_{n=-\infty}^{\infty} x_c(nT)\delta(t-nT) \right)$$

donde $h_0(t)$ es la respuesta al impulso del retenedor de orden cero.

Vemos entonces que la acción de un conversor A/D real se puede modelizar como la modulación por un tren de impulsos seguida de un filtrado por el retenedor de orden cero!!

El diseño de conversores A/D precisos y rápidos es un área muy importante de la electrónica. Existen muchas variantes de circuitos, técnicas de conversión A/D y problemas asociados con los mismos. Sin embargo, a grandes rasgos el proceso de conversión de un conversor A/D comercial es como el que describimos!!

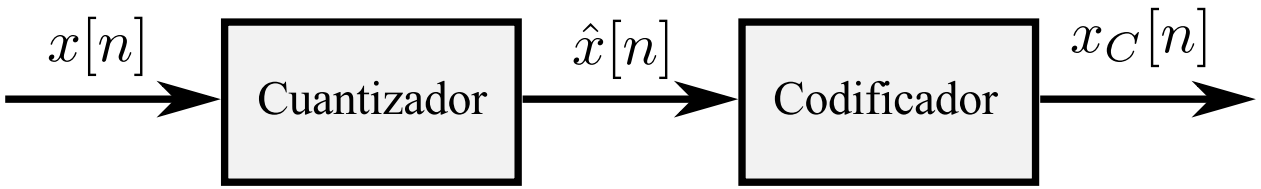
La características más importante de un conversor A/D comercial son su ancho de banda (que tan rápido puede tomar muestras de la señal de entrada) y su precisión (cuántos bits usa para representar las muestras tomadas)!! Estas son básicamente las características que uno "paga" cuando compra un conversor A/D!!

Cuantificación IV

Nos interesará analizar básicamente la precisión del módulo de cuantificación. La acción del dicho módulo se puede resumir en:

$$\hat{x}_C[n] = Q\left(x[n]\right)$$

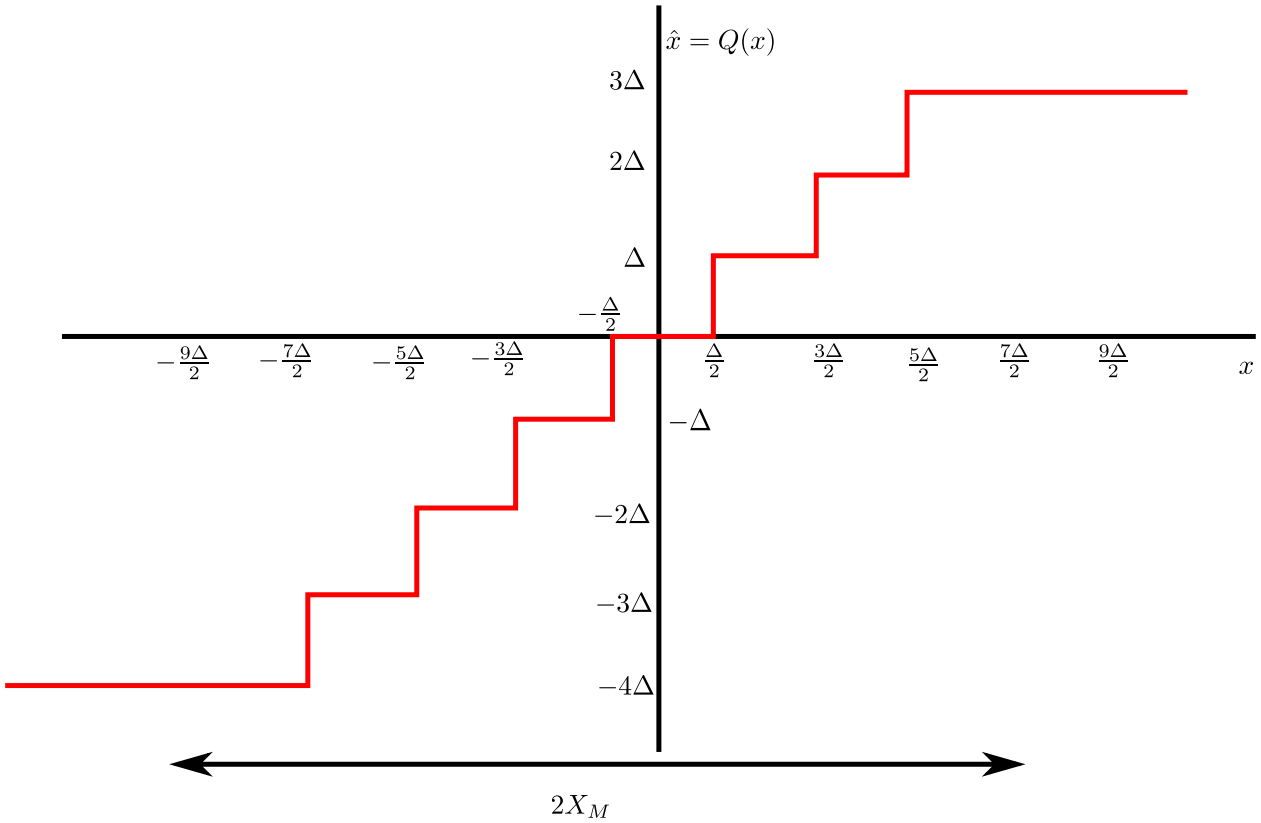
donde $Q : \mathbb{R} \to S$ es un operador *no lineal* y sin memoria, donde S es un conjunto de M elementos. Los M valores a la salida de dicho sistema están predefinidos. La operación de este sistema puede también descomponerse como:



- El cuantizador es el bloque que efectivamente transforma el número real a la entrada en un número con finitos decimales.
- El codificador selecciona para el número entregado por el cuantizador el código binario que lo representará.

Cuantificación V

Un cuantizador típico tiene la siguiente característica:



- El cuantizador tiene rango dinámico igual a $2X_M$ y permite considerar señales que toman valores positivos y negativos. Si una entrada cae fuera del rango dinámico el cuantizador *recorta* a la misma (*clipping*).
- Los pasos de cuantificación son uniformes.
- Los valores de las muestras se redondean hasta el nivel de cuantificación más cercano.

Cuantificación VI

Generalmente el número de niveles de cuantificación es potencia de dos. Es decir, $M = 2^{(B+1)}$. Esto quiere decir, que cada nivel de cuantificación se puede representar usando $B+1$ bits. El bloque de codificación se encargará de asignar las palabras de $B+1$ bits a cada nivel de cuantificación. Existen básicamente 2 esquemas que son los más utilizados:

- Offset binario.
- Complemento a dos.

Nivel de cuantificación	Offset binario	Complemento a dos
-4Δ	000	100
-3Δ	001	101
-2Δ	010	110
$-\Delta$	011	111
0	100	000
Δ	101	001
2Δ	110	010
3Δ	111	011

La mayoría de los procesadores digitales de señales (DSPs) implementan el esquema de codificación de complemento a dos, ya que el mismo permite realizar operaciones aritméticas operando directamente sobre las palabras de código!!

Cuantificación VII

En el sistema de complemento a dos el bit más significativo es el bit de signo, después del cual se considera que existe una coma. Es decir, por convención se considera que los números representados están en $[-1, 1)$. Para $B+1$ bits, tenemos que un número en complemento a dos se representa como:

$$a_0, a_1 a_2 \dots a_B, \quad a_i = \{0, 1\}$$

El valor del número con la representación anterior vale:

$$\hat{x}_C[n] = -a_0 2^0 + a_1 2^{-1} + a_2 2^{-2} + \cdots + a_B 2^{-B}$$

La relación entre las muestras cuantificadas y la palabra de código usada por el codificador se puede escribir:

$$\hat{x}[n] = X_M \hat{x}_C[n]$$

Las muestras codificadas en binario son proporcionales a las muestras cuantificadas!! Por esta razón se pueden usar como una representación numérica válida de las verdaderas muestras cuantificadas!!

Notar que:

$$\Delta = \frac{2X_M}{2^{B+1}} = \frac{X_M}{2^B}$$

Cuantificación VIII

Es importante analizar el error introducido por un bloque cuantizador. Definimos el error de cuantificación $e[n]$ como:

$$e[n] = x[n] - \hat{x}[n]$$

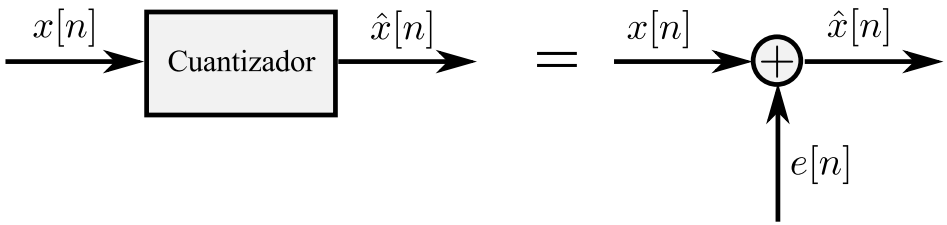
Para un cuantizador que redondea al valor más cercano:

$$-\frac{\Delta}{2} < e[n] \leq \frac{\Delta}{2}$$

siempre y cuando

$$-X_M - \frac{\Delta}{2} < x[n] \leq X_M - \frac{\Delta}{2}$$

Si $x[n]$ queda fuera de este intervalo, el cuantizador *truncará* la salida y el error $e[n]$ no necesariamente cumplirá que $-\frac{\Delta}{2} < e[n] \leq \frac{\Delta}{2}$ Suponiendo que $e[n]$ es conocido, podemos escribir:



Cuantificación IX

Lamentablemente $e[n]$ no será conocido siempre. Por esta razón, nos interesará hacer un análisis de los errores introducidos por el cuantizador. Como en general, no es posible saber con qué tipos de entradas un cuantizador va a ser usado sería conveniente que:

- El análisis debería ser válido para un gran número de señales.
- Debe ser independiente, en la medida de los posible, de las características de las señales $x[n]$ a la entrada del cuantizador.

Por esta razón utilizaremos un modelo estadístico para modelar los errores. El mismo partirá de los siguiente supuestos:

- La secuencia del error $e[n]$ es una secuencia de variables aleatorias, independientes, idénticamente distribuídas y con media cero.
- La secuencia del error $e[n]$ es estadísticamente independiente de la señal de entrada $x[n]$.
- Para cada $n \in \mathbb{Z}$, $e[n]$ está distribuida en forma uniforme sobre el intervalo $[-\frac{\Delta}{2}, \frac{\Delta}{2})$ (no hay clipping o truncamiento).

Aunque hay señales donde estas hipótesis no son ciertas (por ejemplo $x[n] = A \ \forall n \in \mathbb{Z}$). Sin embargo, como las señales del mundo real tienen aleatoriedad estas hipótesis son bastante acertadas (especialmente si $\Delta << 1$)!!

Cuantificación X

Una medida de la degradación que el ruido introduce en la señal de interés es la *relación señal-ruido* (SNR):

$$\text{SNR} = 10 \log_{10} \left(\frac{\sigma_x^2}{\sigma_e^2} \right)$$

Donde σ_x^2 y σ_e^2 son la varianza de la señal $x[n]$ y del error, respectivamente, con:

$$\begin{aligned}
\sigma_e^2 &= \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} e^2 p_e(e) de \\
&= \frac{1}{\Delta} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} e^2 de \\
&= \frac{\Delta^2}{12} \\
&= \frac{X_M^2}{2^{2B} 12}
\end{aligned}$$

Reemplazando:

$$\text{SNR} = 10 \log_{10} \left(\sigma_x^2 \frac{2^{2B} 12}{X_M^2} \right)$$

Cuantificación XI

$$\text{SNR} = 10\log_{10} 12 + 20B\log_{10} 2 - 20\log_{10}\left(\frac{X_M}{\sigma_x}\right) (dB)$$

Observamos lo siguiente:

- Por cada bit extra que nuestro conversor puede entregar tenemos que la SNR aumenta 6 dB.
- Cuanto más pequeña es σ_x con respecto al rango dinámico del conversor X_M mayor es la penalidad en SNR que debemos pagar.
- Sin embargo, tampoco es conveniente aumentar mucho σ_x con respecto a X_M. Por qué??

Cuando se usa un conversor A/D es necesario acondicionar adecuadamente la amplitud de la señal analógica de entrada al rango dinámico del conversor!!

Cuantificación XII

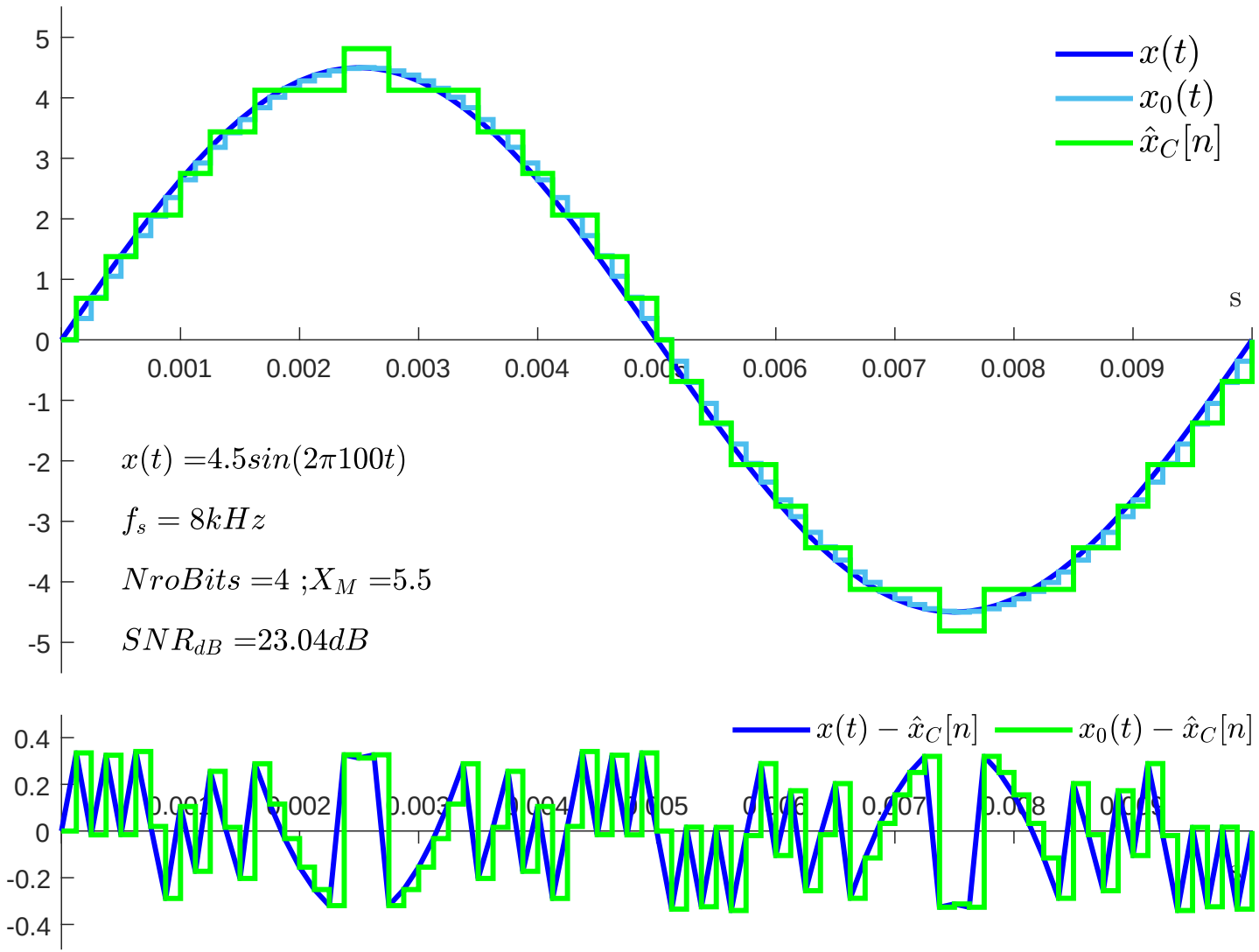
- En algunos casos de interés σ_x se puede aproximar por el valor RMS de la amplitud de la señal.
- Por ejemplo si la señal de tiempo continuo $x_c(t)$ es un seno de amplitud A, tenemos que $\sigma_x = \frac{A}{\sqrt{2}}$.

Veamos algunos ejemplos entonces con funciones senoidales y el efecto de la amplitud de dichas señales en la SNR de a la salida de un conversor A/D.

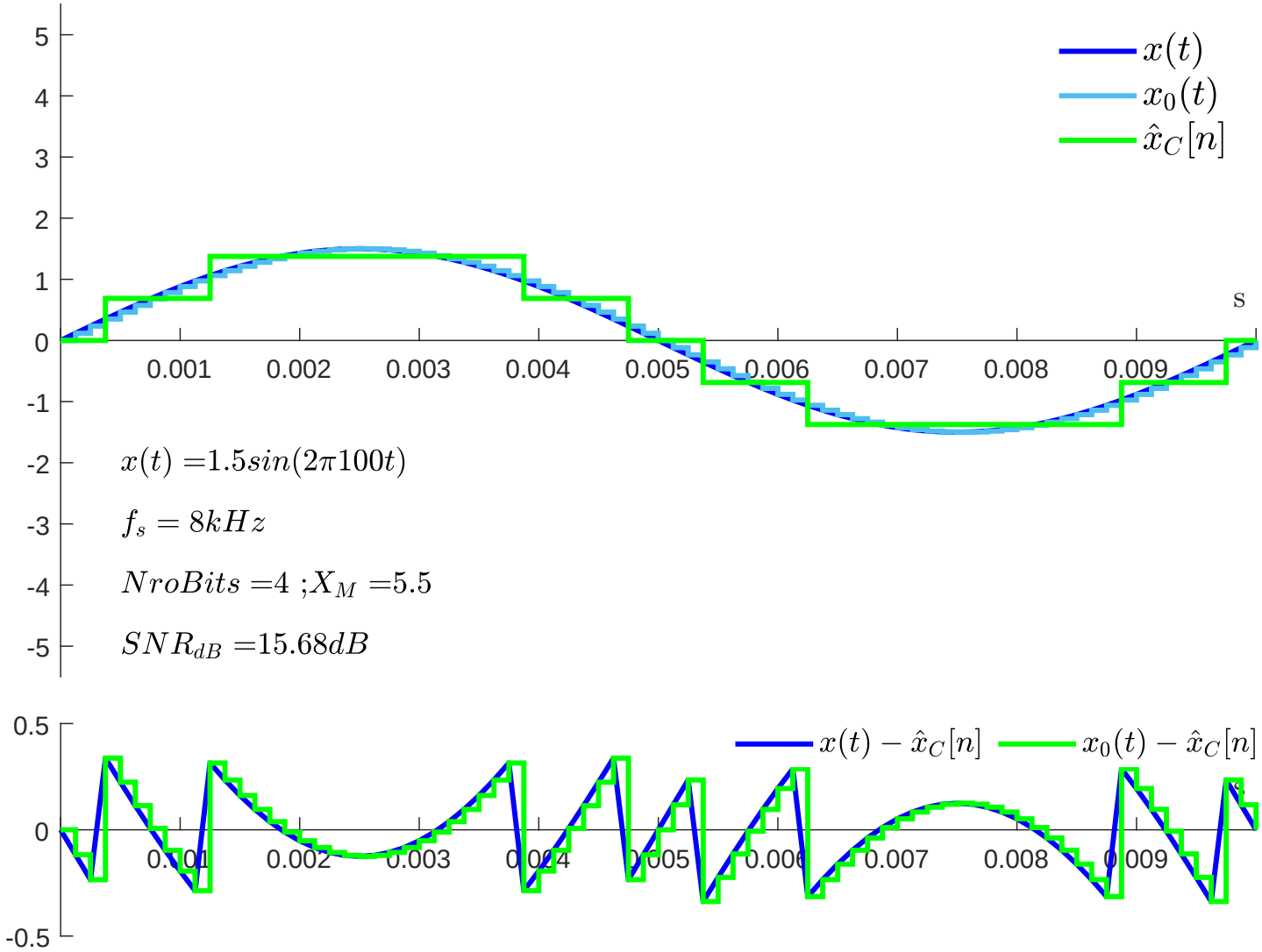
Para calcular el verdadero valor de la SNR vamos a usar la siguiente expresión:

$$\text{SNR} = \frac{\sum_{n=0}^{N-1} x^2[n]}{\sum_{n=0}^{N-1} e^2[n]}$$

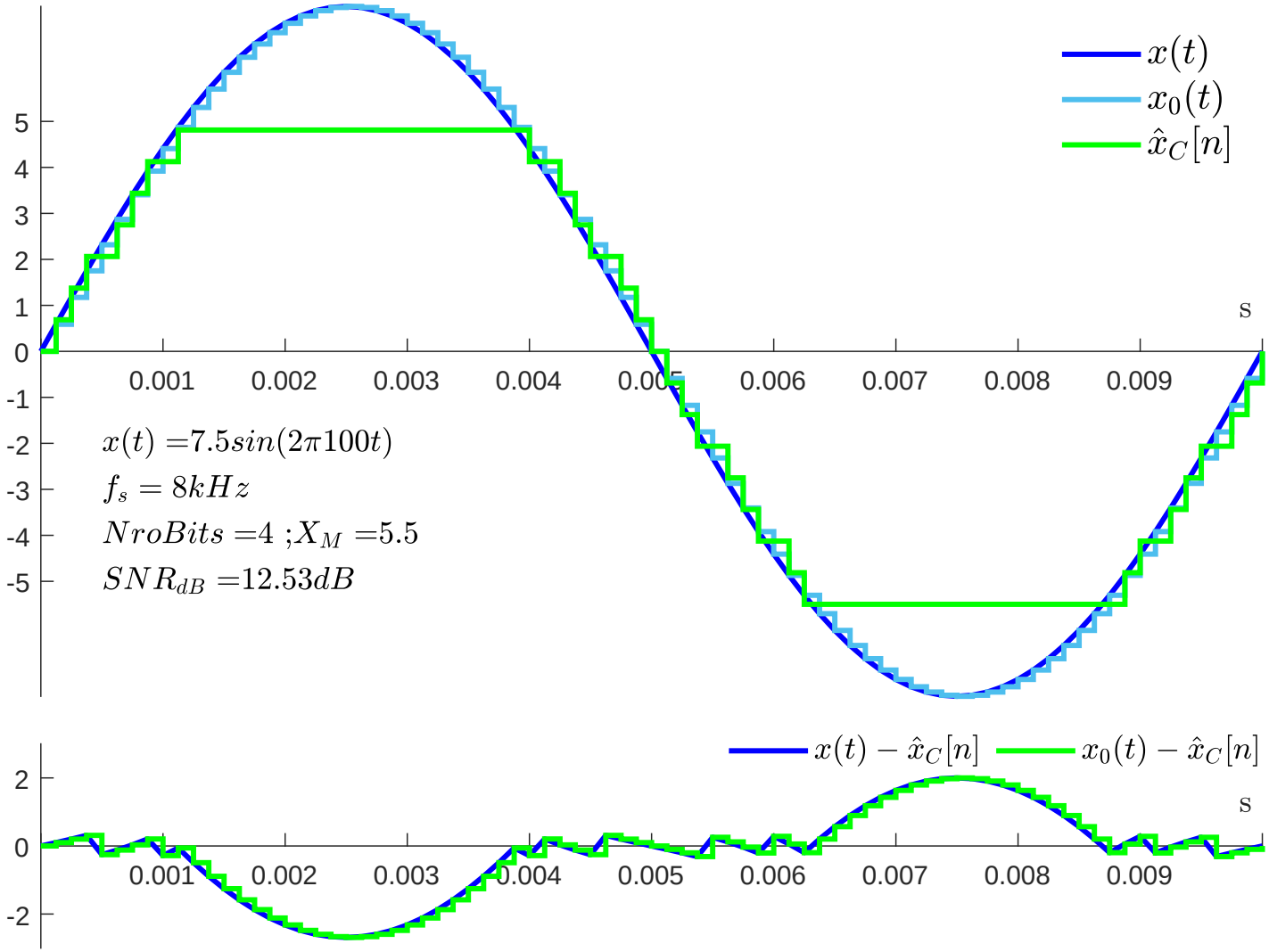
Cuantificación XIII



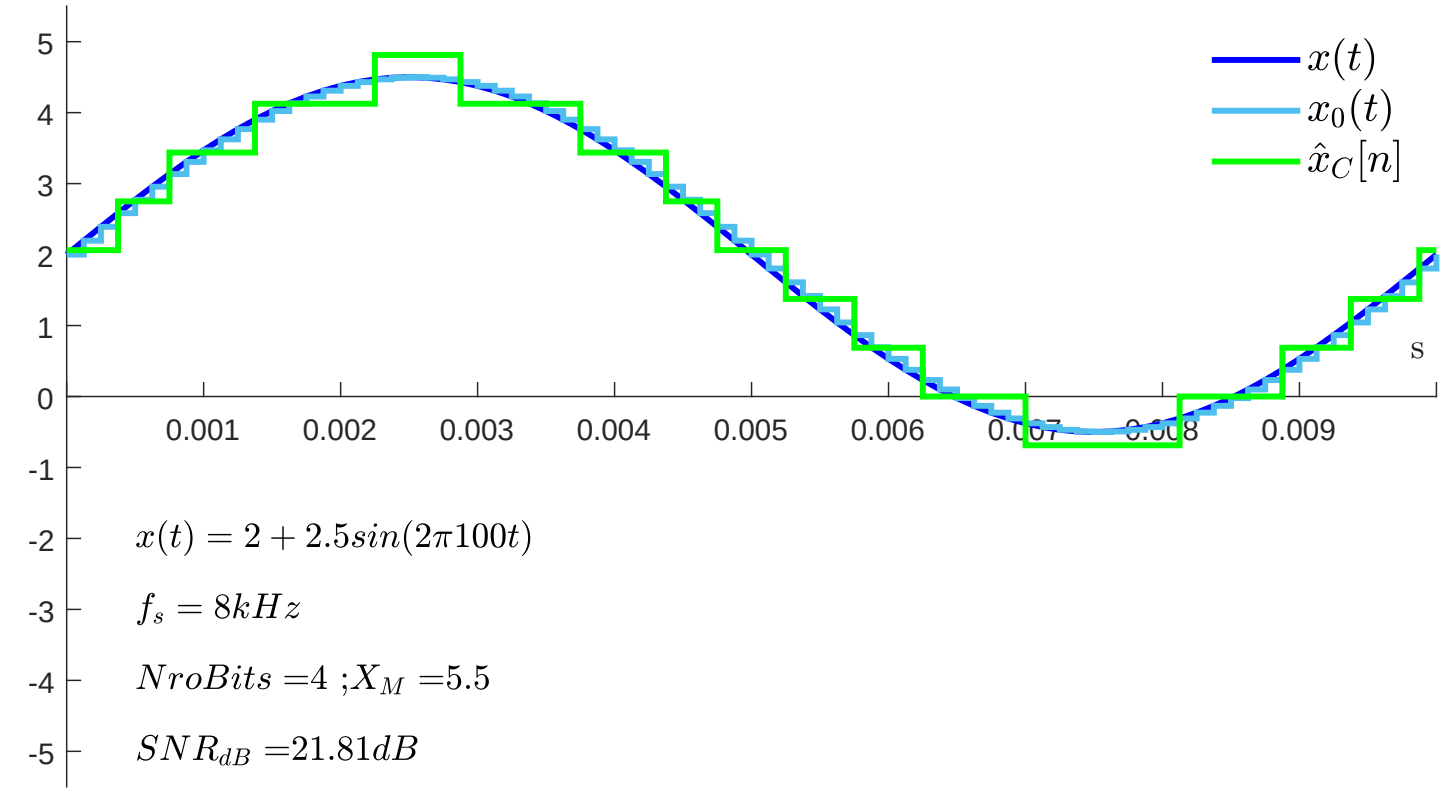
Cuantificación XIV

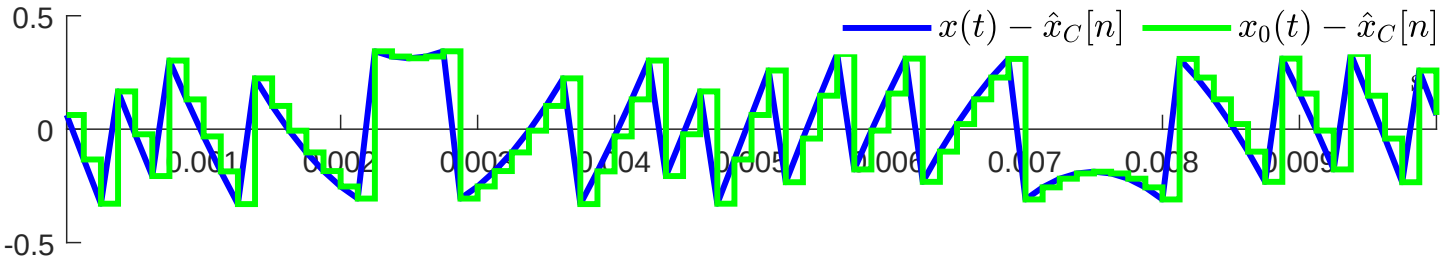


Cuantificación XV



Cuantificación XVI





Cuantificación XVII

Es importante tener una fórmula para el término $20\log_{10}\left(\frac{X_M}{\sigma_x}\right)$ que nos permita en forma general evaluar el número de bits necesarios para una determinada SNR deseada. En general se asume que la señal $x[n]$ tiene una amplitud que se distribuye en forma gaussiana. Esta hipótesis funciona bastante bien para un gran número de señales de interés como voz o música.

Con dicha hipótesis la probabilidad de que la amplitud de la señal supere el valor de $3\sigma_x$ es menor al 0.3 %. Entonces podemos acondicionar la señal analógica de forma tal que $\sigma_x = \frac{X_M}{3}$!!

Entonces obtenemos:

$$\text{SNR} = 6.02B + 1.25 \ (dB)$$

Por ejemplo para obtener una SNR del orden de 100 dB, que es algo usual para audio de alta calidad tenemos que se necesita un conversor que proporcione una cuantificación de 16 bits!! Sin embargo hay que tener en cuenta que dicho nivel de SNR se alcanzará solamente si la amplitud de la señal de entrada se acondiciona adecuadamente al rango dinámico del conversor!!

La clave es siempre aprovechar lo máximo posible el rango dinámico del conversor!!

Temas para leer por cuenta propia

Lectura obligatoria

- Reconstrucción de una señal usando interpolación (Oppenheim and Willsky, Sección 7.2).
- El efecto del submuestreo (Oppenheim and Willsly, Sección 7.3).
- Procesado en tiempo discreto de señales en tiempo continuo (Oppenheim and Schafer, Sección 4.4).
- Procesado digital de señales analógicas (Oppenheim and Schafer, Sección 4.8).

Lectura optativa

- Tratamiento de señales multitasa (Oppenheim and Schafer, Sección 4.7).
- Sobremuestreo y conformación de ruido en la conversión A/D y D/A (Oppenheim and Shafer, Sección 4.9)

Algunos ejercicios I

1. Ejercicios 4.5, 4.8, 4.9, 4.12, 4.15, 4.19, 4.20, 4.22 de Oppenheim and Schafer.
2. Ejercicios 4.31, 4.35, 4.37, 4.38, 4.43, 4.44, 4.56, 4.57, 4.59 de Oppenheim and Schafer.
3. Ejercicio 4.53 de Oppenheim and Schaffer.
4. Ejercicio 7.21, 7.23, 7.24, 7.26, 7.27, 7.28, 7.29, 7.31, 7.37, 7.38 de Oppenheim and Willsky
5. Ejercicio 7.41, 7.42, 7.43, 7.44, 7.50, 7.51, 7.52 de Oppenheim and Willsky.

Tiempo de consultas

¿Preguntas?