Práctica 1

Introducción básica al procesado de sonido en Matlab

1.1. Transformada de Fourier de señales continuas

El espectro de una señal analógica $x(t)$ se define como

$$X(\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\Omega t}dt \tag{1.1}$$

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt \tag{1.2}$$

siendo $\Omega = 2\pi f$ la frecuencia en radianes por segundo (rad/s) y f la frecuencia en Hz.

La transformada inversa se define como

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\Omega)e^{j\Omega t}d\omega \tag{1.3}$$

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{j2\pi ft}df \tag{1.4}$$

La tabla 1.1 muestra algunos ejemplos de pares de transformadas frecuentemente utilizados.

$x(t)$	$X(\Omega)$	$X(f)$
$\Pi(t/T_0)$	$T_0 sinc(\Omega T_0/2)$	$T_0 sinc(T_0 f)$
$\cos(\Omega_0 t)$	$\pi\left[\delta(\Omega-\Omega_0)+\delta(\Omega+\Omega_0)\right]$	$\frac{1}{2}\left[\delta(f-f_0)+\delta(f+f_0)\right]$
$\sin(\Omega_0 t)$	$\frac{\pi}{j}\left[\delta(\Omega-\Omega_0)-\delta(\Omega+\Omega_0)\right]$	$\frac{1}{2j}\left[\delta(f-f_0)-\delta(f+f_0)\right]$
$\sum_{k=-\infty}^{\infty}\delta(t-nT_0)$	$\frac{2\pi}{T_0}\sum_{n=-\infty}^{\infty}\delta(\Omega-n\frac{2\pi}{T_0})$	$\frac{1}{T_0}\sum_{n=-\infty}^{\infty}\delta(f-\frac{n}{T_0})$

Tabla 1.1: Ejemplos de pares de transformadas

1.2. Muestreo uniforme de señales

El teorema de muestreo nos dice que una señal continua de ancho de banda B Hz puede ser reconstruida a partir de sus muestras si la frecuencia de muestreo, f_s, verifica que $f_s \geq 2B$.

En figura 1.1 se ilustra este proceso, donde se puede comprobar que si $f_s \geq 2B$ la señal original se puede recuperar mediante un filtro paso bajo ideal, mientras que si $f_s < 2B$ se produce un solapamiento espectral conocido, por el término inglés “aliasing", que nos impide dicha recuperación.

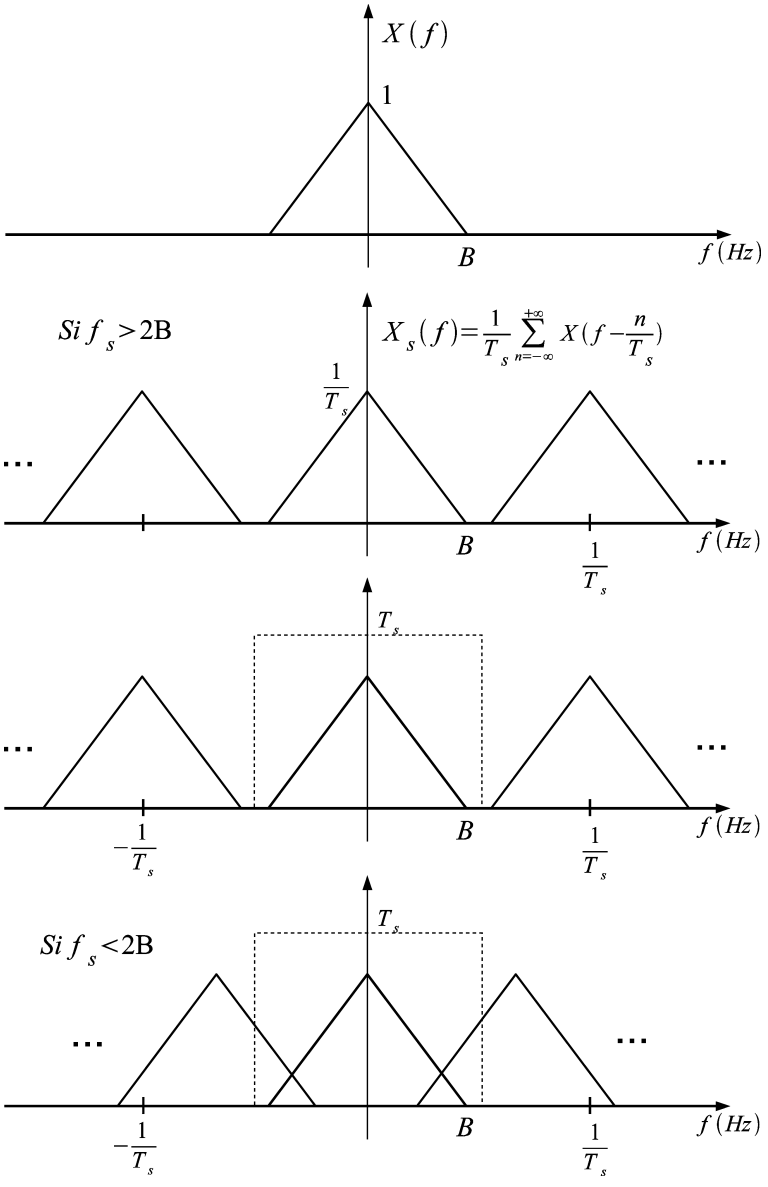

Figura 1.1: Ilustración del teorema del muestreo

Dado que en la práctica no existen las señales estrictamente limitadas en banda, antes de realizar el muestreo es siempre necesario realizar un filtrado previo que nos limite la señal al ancho de banda de interés (que dependerá de la aplicación).

Como resultado del muestreo de una señal $x(t)$ vamos a obtener una secuencia discreta de valores $x(n) = x(nT_s)$. Como es bien sabido el espectro de una secuencia discreta viene dado por

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \tag{1.5}$$

y es periódico en ω, que ahora representa la frecuencia discreta en rad. La relación entre Ω (rad/s) y ω se establece a través de $f_s = 1/T_s$, de forma que $\omega = \Omega T_s$. Esta relación se ilustra en la figura 1.2, donde se comprueba que $fs/2$ se corresponde con la frecuencia discreta $\omega = \pi$.

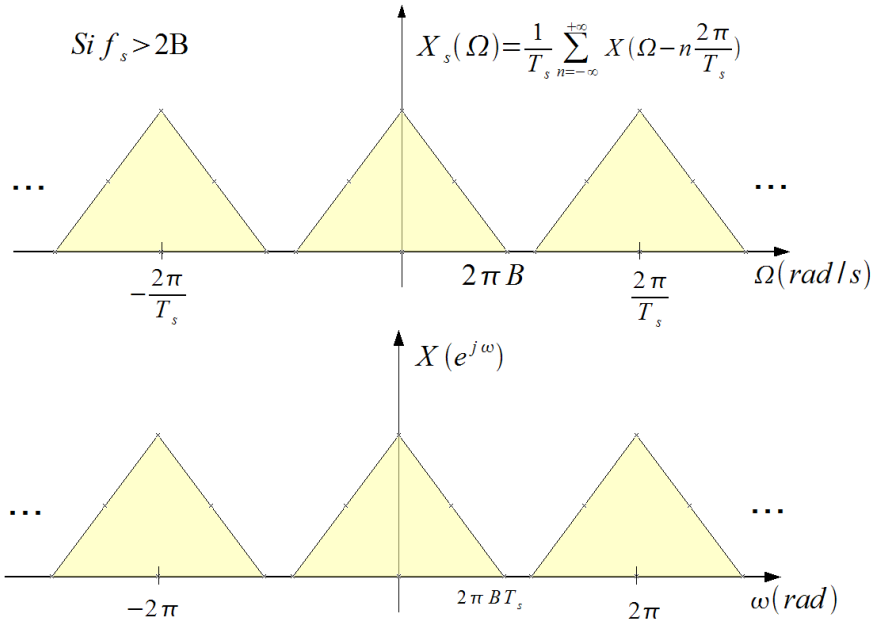

Figura 1.2: Relación entre la frecuencia continua y discreta

1.3. Cuantificación

Teóricamente la dimensión de conjunto de posibles valores a la salida de un muestreador es infinita (incluso con entrada acotada). Sin embargo, en la práctica es necesario representar dichos valores con una determinada precisión (número de bits) lo que implica reducir a un número finito los posibles valores de salida. Este proceso se conoce como cuantificación y se ilustra en la figura 1.3. Evidentemente el proceso de cuantificación introduce una distorsión (error) que dependerá de la precisión utilizada.

1.4. Modificación de la frecuencia de muestreo

Un parámetro inherente a cualquier secuencia discreta obtenida a través del muestreo de una señal continua es la frecuencia de muestreo. En la práctica sucede que muy frecuentemente hemos de trabajar con señales adquiridas por otros y no siempre a la frecuencia de muestreo que necesitamos o bien cierto algoritmo nos exige trabajar con una frecuencia de muestreo

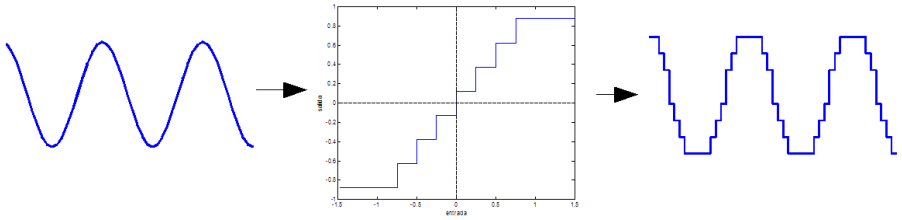

Figura 1.3: Cuantificación uniforme

determinada. Existen técnicas que permiten la modificación de la frecuencia de muestreo sin necesidad de remuestrear la señal original, aunque como veremos existe alguna limitación. Así la técnica denominada interpolación nos va a permitir incrementar la frecuencia de muestreo a un múltiplo de la frecuencia de muestreo de partida, mientras que el diezmado nos posibilitará reducirla a un submúltiplo de la frecuencia original.

Conviene tener presente que si hemos realizado el muestreo de la señal continua a una frecuencia adecuada f_{s0} dispondremos, como máximo, de las componentes espectrales de la señal original hasta $f_{s0}/2$. Por tanto, aunque incrementemos la frecuencia de muestreo hasta $f_{s1} = Lf_{s0}$ (L entero) mediante interpolación no se van a generar nuevas componentes espectrales y simplemente tendremos un mayor rango de frecuencias con valor nulo o despreciable. Otra cosa muy distinta sería muestrear la señal original directamente a la frecuencia f_{s1} habiéndola previamente limitado en banda a $f_{s1}/2$ Hz.

Por lo que respecta al diezmado, la reducción de la frecuencia de muestreo implica necesariamente una reducción del ancho de banda, por lo que si deseamos que la nueva frecuencia de muestreo sea $f_{s1} = f_{s0}/M$ hemos de eliminar previamente aquellas componentes espectrales por encima de $f_{s1}/2$. En este caso el proceso sí es equivalente a remuestrear la señal continua original a la nueva frecuencia f_{s1}.

En la figura 1.4 se ilustran los diagramas de bloques típicos de las técnicas de diezmado e interpolación, y en la figura 1.5 la combinación de ambas para modificar la frecuencia de muestreo por un factor racional L/M.

El efecto sobre el espectro de diezmado e interpolación se ejemplifica en la figura 1.6.

1.5. Filtros digitales

Los filtros digitales son herramientas básicas del procesado de señal. Los filtros digitales pueden ser FIR (Finite Impulse Response) o IIR (Infinite Impulse Response).

La función de transferencia de un filtro FIR es de la forma

$$H(z) = \sum_{k=0}^{L} h[k] z^{-k} \tag{1.6}$$

siendo $h[n]$ su respuesta impulsional o, equivalentemente, el conjunto de coeficientes del filtro. La salida de este filtro ante una entrada $x[n]$ se puede obtener fácilmente como

$$y[n] = \sum_{k=0}^{L} h[k] x[n-k] \tag{1.7}$$

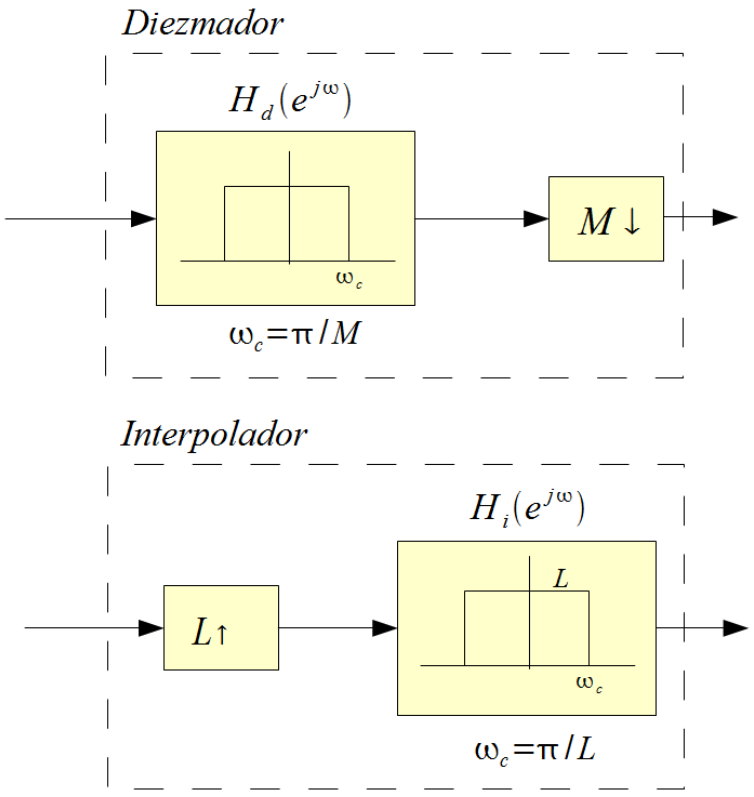

Figura 1.4: Diagramas de bloques de un diezmador y un interpolador.

La función de transferencia genérica de un filtro IIR es

$$H(z) = \frac{\sum_{k=0}^{M} b[k]z^{-k}}{\sum_{k=0}^{N} a[k]z^{-k}} \tag{1.8}$$

y su salida se puede calcular de acuerdo con

$$y[n] = \frac{1}{a[0]}\left[\sum_{k=0}^{M} b[k]x[n-k] - \sum_{k=1}^{M} a[k]y[n-k]\right] \tag{1.9}$$

La elección de un tipo u otro de filtro depende de la aplicación ya que cada uno tiene sus ventajas e inconvenientes. La ventaja más notable de los filtros IIR es que con un menor número de coeficientes que un FIR logran una respuesta en frecuencia más selectiva. Además el retardo a su salida es mucho menor que en el caso del filtros FIR. Sin embargo los filtros IIR no tienen fase lineal, mientras que el diseño de filtros FIR de fase lineal es sencillo.

A lo largo de esta asignatura utilizaremos fundamentalmente filtros FIR de fase lineal.

Ejercicio 1 *Mediante la función Matlab "fir1" diseña algunos filtros FIR paso bajo, pasa banda y paso alto para distintas frecuencias de corte y órdenes (por ejemplo N=20, 100, 200). Observa sus respuestas impulsionales y sus respuestas en frecuencia con la herramienta "fvtool".*

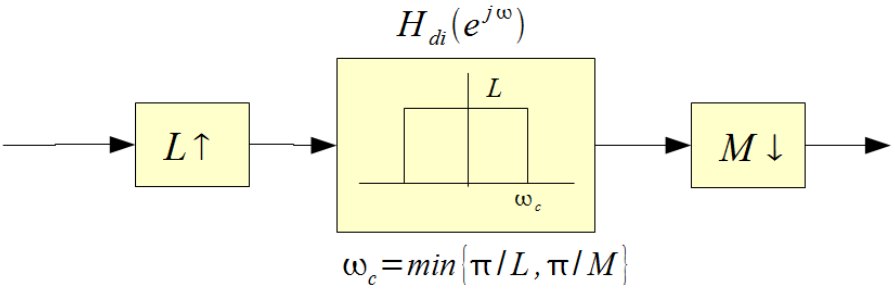

Figura 1.5: Modificación de la frecuencia de muestreo por un factor racional.

Ejercicio 2 *Genera 200 muestras de una sinusoide discreta de frecuencia $\pi/2$ y fíltrala con un filtro FIR paso bajo de frecuencia $3\pi/4$ utilizando la función "filter". Compara las formas de onda de la salida y la entrada.*

Ejercicio 3 *Considera la señal musical almacenada en el fichero "audio1.wav". Lee el fichero con la función "audioread" y reproduce el vector leído con la frecuencia de muestreo adecuada utilizando "soundsc". Vuelve a reproducirlo con frecuencias de muestreo mayores y menores. Comenta lo que escuchas.*

Ejercicio 4 *Utilizando las funciones "fft" y "plot", representa el módulo del espectro de un segmento de un segundo del fichero "audio2.wav". ¿Qué ancho de banda tiene? Filtra la señal de audio completa con diferentes filtros paso bajo con frecuencias de corte 3500, 7500 y 15000 Hz y orden N=200. Escucha las señales obtenidas.*

Ejercicio 5 *Crea sendas funciones Matlab que implementen un diezmador y un interpolador por un factor L. La cabecera de la llamada a la función debe ser similar a:*

```
function [y fsd]= diezmado(x,M,fs,orden_filtro)
```

1.6. Bancos de filtros

Son una herramienta ampliamente utilizada para análisis y/o codificación de señales. En la figura 1.7 se representa el esquema de un banco de filtros que divide la señal de entrada en dos bandas de frecuencias. En las figuras 1.8, 1.9 y 1.10 se ilustra su funcionamiento.

Ejercicio 6 *Comencemos por ver la típica respuesta en frecuencia de los filtros de un banco de filtros. Edita el script Matlab 'filtrosqmf.m'. Observa su contenido y ejecútalo.*

Ejercicio 7 *Analiza ahora el contenido del script "bancofiltros.m". Relaciona los distintos vectores con el diagrama de bloque de la figura 1.7. Representa el espectro de las distintas señales a la entrada y salida de los distintos bloques.*

Ejercicio 8 *Repite el ejercicio anterior para la entrada alternativa que se propone en "bancofiltros.m". Relaciona los distitnos espectros con los de las figuras 1.8, 1.9 y 1.10.*

Ejercicio 9 *Si quisieras ahora implentar un banco de filtros de cuatro bandas de frecuencia,¿cómo lo harías?*

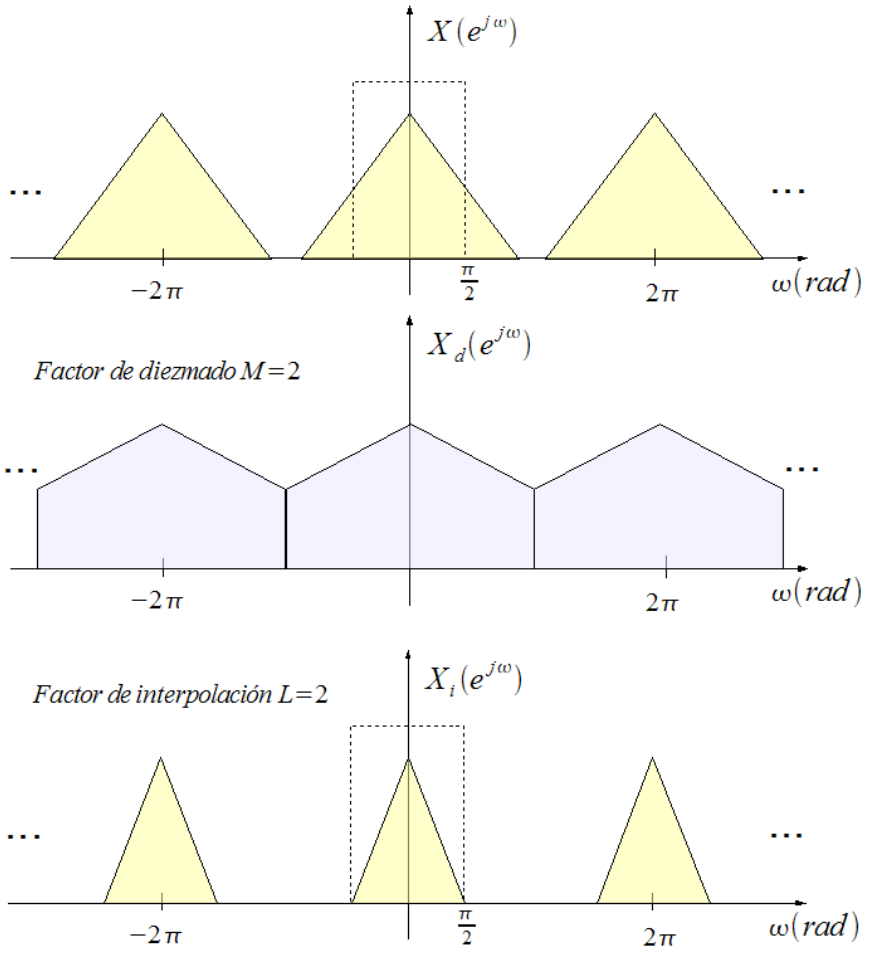

Figura 1.6: Modificación de la frecuencia de muestreo por un factor de 2.

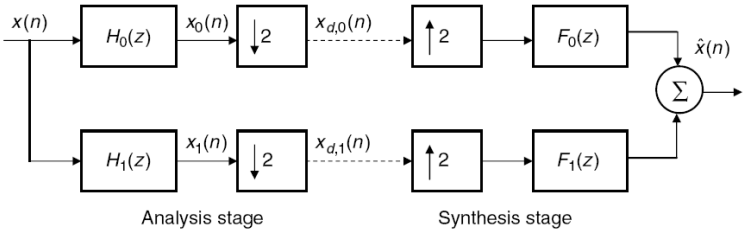

Figura 1.7: Ejemplo de análisis y síntesis mediante un banco de filtros

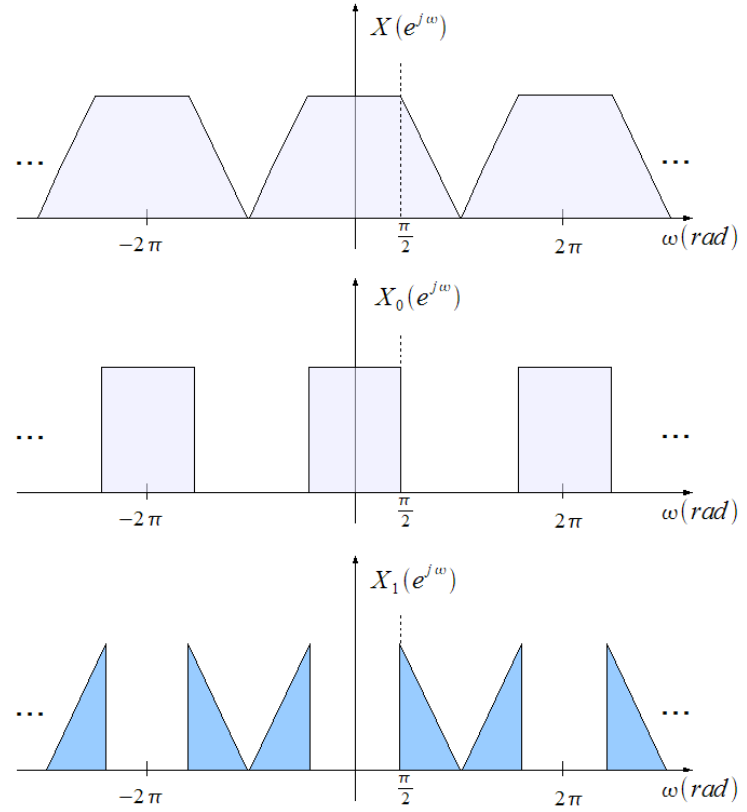

Figura 1.8: Salidas de los filtros de los diezmadores

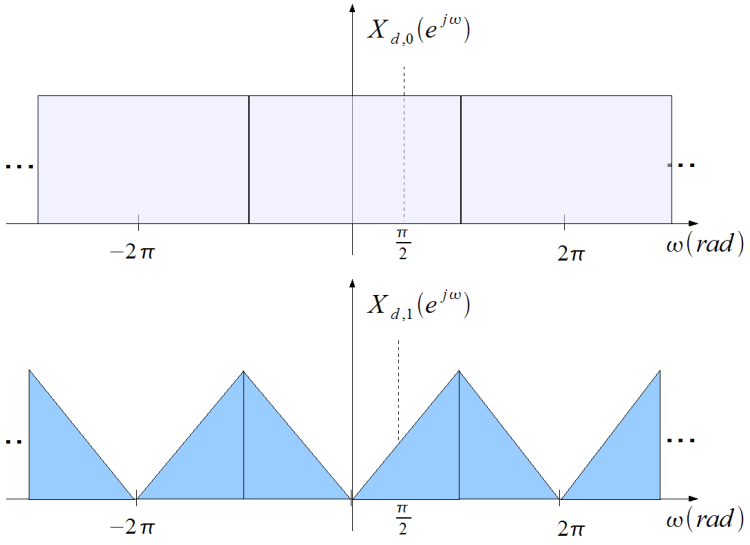

Figura 1.9: Salidas de los diezmadores

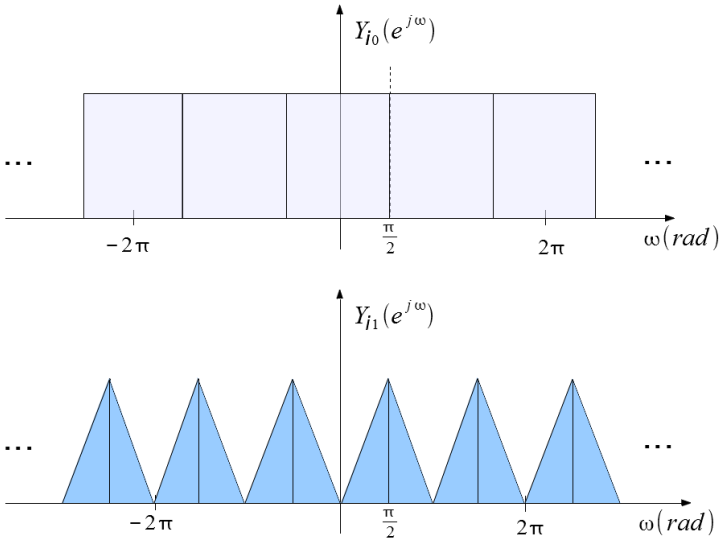

Figura 1.10: Entradas a los filtros interpoladores