

## Diseño de filtros digitales

La mayoría de los textos sobre procesamiento digital de señales dedican un espacio sustancial a la teoría de diseño de filtros –incluyendo especialmente los métodos de aproximación tipo Butterworth, Chebyshev, elíptico, Cauer, etc.– lo que refleja no sólo el formidable esfuerzo de investigación destinado a desarrollar métodos útiles para el cálculo de los coeficientes de los filtros, sino también los avances significativos que se han logrado en este campo. En este capítulo, basado en (Ifeachor y Jarvis, 1993) se presenta un panorama global del proceso de diseño, postergando los aspectos teóricos a los capítulos siguientes. Se describe paso a paso el proceso de diseño de filtros desde las especificaciones hasta la implementación, y se detallan las opciones de que dispone el diseñador, y los factores que pueden influir en sus decisiones.

### 7.1. Filtrado digital de señales

Un filtro es un sistema o una red que cambia selectivamente la forma de onda, o las características amplitud-frecuencia o fase-frecuencia de una manera deseada. Los objetivos comunes del proceso de filtrado son mejorar la calidad de la señal, por ejemplo removiendo o atenuando el nivel de ruido, extrayendo información de dos o más señales previamente combinadas para hacer uso eficiente de un canal de comunicación, etc.

Un filtro digital es un algoritmo implementado en hardware y/o software que opera sobre una señal de entrada digital (discreta en tiempo y cuantizada en amplitud) y genera una señal digital de salida, con la finalidad de efectuar un proceso de filtrado. El término “filtro digital” se refiere al hardware o software específico que ejecuta el algoritmo. Los filtros digitales trabajan sobre valores numéricos asociados a muestras de esas señales analógicas previamente digitalizadas por conversores A/D o simplemente sobre un conjunto de números almacenados en la memoria de una computadora o microprocesador.

En la Fig. 7.1 se muestra un diagrama bloque simplificado de un filtro digital que opera en tiempo real, con entradas y salidas analógicas. La señal de entrada, limitada en banda, se muestrea periódicamente y se convierte en una serie de muestras  $x[n]$ ,  $n = 0, 1, \dots$ . El procesador digital convierte la sucesión de entrada  $x[n]$  en una sucesión de salida  $y[n]$  de acuerdo al algoritmo de cómputo. El conversor digital-analógico convierte la salida



**Fig. 7.1.** Diagrama bloque simplificado de un filtro digital.

digital  $y[n]$  a valores continuos en tiempo que son procesados por un filtro analógico para suavizar la forma de onda y remover componentes no deseadas de alta frecuencia.

Los filtros digitales juegan un papel muy importante en el procesamiento digital de señales. En gran número de aplicaciones, como compresión de datos, procesamiento de señales biomédicas, procesamiento de señales de voz, procesamiento de imágenes, transmisión de datos, audio digital, cancelamiento de ecos telefónicos, se prefieren por sobre los filtros analógicos por uno o más de los siguientes motivos:

- Los filtros digitales pueden tener características que son imposibles de conseguir con filtros analógicos, como por ejemplo una respuesta de fase *exactamente* lineal.
- El desempeño de los filtros digitales no varía con las condiciones ambientales (temperatura, humedad, etc.) como sí ocurre con los filtros analógicos, lo que elimina la necesidad de calibrarlos periódicamente.
- Si el filtro se implementa utilizando un procesador programable la respuesta en frecuencia de un filtro digital puede ajustarse a voluntad (filtrado adaptivo).
- El mismo filtro puede procesar varias señales o canales de entrada sin necesidad de replicar el hardware.
- Las señales filtradas y sin filtrar pueden almacenarse para uso o análisis posterior.
- Los avances en las técnicas de integración VLSI hacen que sea posible fabricar filtros digitales pequeños, de bajo consumo, y de muy bajo costo.
- La precisión con que un filtro digital verifica las especificaciones de diseño está limitada solamente por la longitud de palabra (bits) utilizada para representar los coeficientes del filtro y ejecutar las operaciones aritméticas; con los filtros analógicos es difícil lograr atenuaciones que excedan los 60 o 70 dB en la banda de rechazo (utilizando componentes convencionales).
- El desempeño de los filtros digitales es repetible de unidad a unidad.
- Los filtros digitales pueden utilizarse a muy bajas frecuencias, como las que se encuentran en aplicaciones biomédicas, donde el empleo de filtros analógicos es poco práctico por los valores muy elevados de los componentes pasivos involucrados (capacitores, inductancias). Además, los filtros digitales pueden trabajar sobre un amplio rango de frecuencias simplemente cambiando la frecuencia de muestreo.

Sin embargo, los filtros digitales también presentan una serie de desventajas respecto a los filtros analógicos:

- *Limitación de frecuencia.* La frecuencia de Nyquist –que fija el ancho de banda útil que el filtro puede procesar– queda definida por el proceso de conversión (tiempos

de conversión del conversor A/D y D/A), velocidad del procesador, cantidad de operaciones a ejecutar por unidad de tiempo, etc. Este último término se incrementa a medida que aumenta la exigencia de las características de respuesta del filtro (filtros muy abruptos).

- *Efectos de longitud finita de palabra.* En general, los coeficientes del filtro implementado serán distintos de los calculados teóricamente si la representación numérica que se utiliza para implementar el filtro no es de *precisión infinita* (punto flotante). No sólo influye la cuantización de los coeficientes del filtro, sino también el redondeo de las operaciones numéricas, la cuantización del conversor A/D y D/A, la truncación que ocurre al almacenar los contenidos del acumulador en memoria, etc. Estos efectos, que se modelan como fuentes de ruido de distribución uniforme, pueden limitar seriamente el desempeño de los filtros digitales: variaciones de ganancia en la banda de paso, menor atenuación en la banda de rechazo, y hasta pueden conducir a la inestabilidad en filtros recursivos de orden elevado.
- *Tiempos de diseño y desarrollo prolongados.* Los tiempos de diseño y desarrollo de un filtro digital, en especial el diseño del hardware puede ser muy superior al necesario para el diseño de un filtro analógico. Sin embargo, una vez que se dispone del hardware o el software necesario, la misma plataforma puede utilizarse para muchas otras tareas de filtrado o procesamiento digital de señales con poca o ninguna modificación. Además, el desarrollo de herramientas de CAD avanzadas hacen que el diseño de filtros sea una tarea agradable y sencilla, aunque aún así es necesaria cierta experiencia para aprovecharlas íntegramente.

## 7.2. Filtros FIR e IIR

Los filtros digitales lineales e invariantes en el tiempo pueden clasificarse de acuerdo a la longitud de su respuesta impulsiva  $h[n]$  como IIR, cuando la respuesta tiene duración infinita o FIR, si su duración es finita. Esta respuesta impulsiva  $h[n]$ ,  $n = 0, 1, 2, \dots$  caracteriza completamente el filtro, a punto tal que las señales de entrada y salida están relacionadas por la suma de convolución, que para filtros IIR toma la forma

$$y[n] = \sum_{k=0}^{\infty} h[k]x[n-k], \quad (7.1)$$

y para filtros FIR es

$$y[n] = \sum_{k=0}^{N-1} h[k]x[n-k]. \quad (7.2)$$

Del análisis de estas dos ecuaciones es evidente que, mientras que la suma convolución puede ser una forma apropiada para implementar un filtro FIR, no es adecuada para los filtros IIR debido a que la respuesta impulsiva es muy larga (en teoría, infinitamente larga). Por ello, los filtros IIR se implementan con ecuaciones a diferencia que permiten calcular las muestras de salida en forma recursiva

$$y[n] = - \sum_{k=1}^N a_k y[n-k] + \sum_{k=0}^M b_k x[n-k], \quad (7.3)$$

donde  $n = 0, 1, 2, \dots, y[-1] = 0$ . El número  $N$  es el orden del filtro, y fija la cantidad de modos de la respuesta impulsiva. La relación entre los coeficientes  $a_i$  y  $b_i$  se obtiene aplicando la transformada  $\mathcal{Z}$  a (7.3) y antitransformando. En la ecuación (7.3) la salida  $y[n]$  es función de los valores actuales y pasados de la entrada, y de valores pasados de la salida (de ahí el nombre *recursivo*): el filtro IIR es sistema realimentado. En cambio, en el filtro FIR la salida  $y[n]$  sólo depende de los valores pasados de la entrada  $x[n]$ . Es evidente que, si en la ecuación (7.3) de los filtros IIR los coeficientes  $a_i$  son nulos, se obtiene la suma convolución (7.2) de los filtros FIR, pues hacer  $a_i = 0$  anula los efectos de realimentación de las salidas pasadas. Las ecuaciones (7.2) y (7.3) son las ecuaciones a diferencias que se utilizan para implementar filtros FIR e IIR, respectivamente. Estas ecuaciones, y en particular el valor de los coeficientes  $h[n]$  para los filtros FIR y  $a_i, b_i$  para los filtros IIR, son los objetivos del problema de diseño de filtros digitales.

Las funciones de sistema están dadas por

$$H_{\text{FIR}}(z) = \sum_{k=0}^{N-1} h[k]z^{-k} \quad \text{y} \quad H_{\text{IIR}}(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{\sum_{k=0}^N a_k z^{-k}}$$

para filtros FIR e IIR, respectivamente, y resultan convenientes para el análisis de estabilidad, respuesta en frecuencia, etc.

Los factores que influyen en la elección entre distintas alternativas en cada etapa del diseño de filtro tienen mucho que ver con que el filtro sea FIR o IIR. Por ello resulta importante apreciar las diferencias entre ambos tipos de filtros, las características peculiares de cada uno, y, lo más importante, cómo elegir entre uno y otro.

### 7.3. Comparación entre filtros FIR e IIR

La elección entre una implementación FIR e IIR depende de las ventajas relativas de cada uno de estos dos tipos de filtros.

1. Los filtros FIR se pueden diseñar para tener una respuesta de fase estrictamente lineal (distorsión de fase nula), lo que es importante en muchas aplicaciones, como transmisión de datos, audio digital y procesamiento de imágenes. La respuesta de fase de filtros IIR no es lineal, en especial en cercanías de la zona de transición.
2. Los filtros FIR implementados de forma no recursiva, por ejemplo aplicando (7.2), son inherentemente estables. En cambio, la estabilidad de los filtros IIR siempre debe comprobarse, ya que son sistemas realimentados.
3. Los efectos causados por la implementación con aritmética de punto fijo, tales como los errores de cuantización de los coeficientes y los errores por redondeo en las operaciones aritméticas, son mucho más severos en los filtros IIR que en los FIR.
4. Para satisfacer unas especificaciones dadas los filtros FIR necesitan un mayor número de coeficientes que los filtros IIR, sobre todo si las bandas de transición son estrechas. En consecuencia, los requerimientos de memoria, el número de operaciones y los tiempos de procesamiento son mayores para los FIR que para los IIR. Sin embargo, la posibilidad de implementar los FIR mediante la técnica de convolución rápida usando FFT y también el empleo de técnicas *multirate* permiten aumentar significativamente la eficiencia de las implementaciones.

5. Un filtro analógico convencional puede convertirse en un filtro digital IIR equivalente que satisfaga las especificaciones de diseño de manera sencilla. Esto no es posible con filtros FIR pues no tienen una contraparte analógica. Sin embargo es más sencillo sintetizar filtros con respuestas en frecuencia arbitrarias utilizando filtros FIR.

De las características detalladas arriba puede esbozarse una guía tentativa para elegir entre una implementación FIR o IIR:

- Si los únicos requerimientos importantes son bandas de transición estrechas (filtros con cortes muy abruptos) y eficiencia de cómputo, se prefieren filtros IIR pues necesitan un número de coeficientes mucho menor que un filtro FIR equivalente (especialmente si se eligen características frecuenciales elípticas o de Cauer).
- Si el número de coeficientes del filtro no es muy elevado (por ejemplo, si las bandas de transición no son muy abruptas), y en particular, si se desea muy poca o ninguna distorsión de fase, se suele elegir filtros FIR. Los procesadores digitales modernos (DSP) están optimizados para implementar este tipo de filtros, y algunos se han diseñado específicamente con esa finalidad (por ejemplo, el DSP56200 de Motorola, o el INMOS A100). Sin embargo, en un campo tan dinámico como éste la capacidad y el desempeño de los componentes varía rápidamente.

#### EJEMPLO 7.1. Comparación de las características de filtros FIR e IIR

Se desea diseñar un filtro que cumpla con las siguientes especificaciones:

banda de paso:	$0 \text{ a } 3\pi/5$ ,
banda de rechazo:	$4\pi/5 \text{ a } \pi$ ,
ganancia en la banda de paso:	$1 \pm 0,06$ ,
ganancia en la banda de rechazo:	$< 0,2$ .

Los requisitos de diseño se pueden satisfacer tanto como un filtro IIR o FIR:

##### ■ Filtro 1 (IIR):

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}, \quad (7.4)$$

con  $a_1 = -0,6744878$ ,  $a_2 = -0,3633482$ ,  $b_0 = 0,4981819$ ,  $b_1 = 0,9274777$ ,  $b_2 = 0,4981819$ . La Fig. 7.2 muestra el diagrama bloque de una posible implementación del filtro, cuyas ecuaciones a diferencia son

$$\begin{aligned} w[n] &= x[n] - a_1 w[n-1] - a_2 w[n-2], \\ y[n] &= b_0 w[n] + b_1 w[n-1] + b_2 w[n-2]. \end{aligned}$$

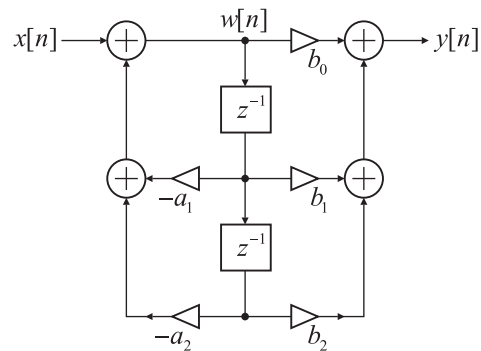
En la Fig. 7.3 se grafica la respuesta en frecuencia (en módulo y fase), y el retardo de grupo, comprobándose que se satisfacen las especificaciones de diseño.

##### ■ Filtro 2 (FIR):

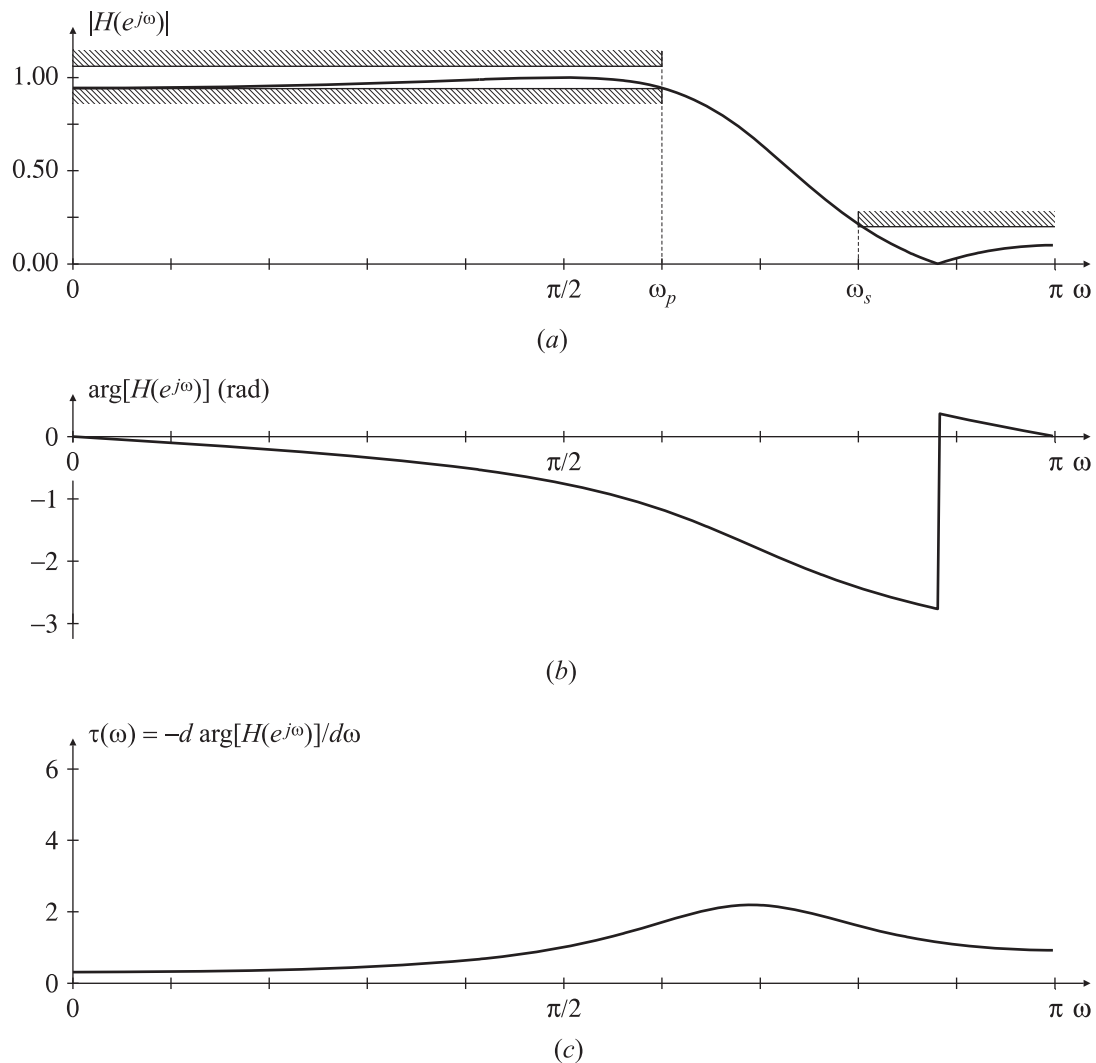
$$H(z) = \sum_{k=0}^{11} h[k] z^{-k}$$

con

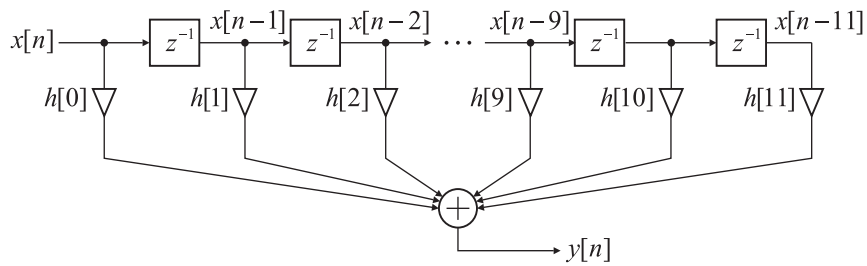
$$\begin{aligned} h[0] &= h[11] = +0,5460328 \times 10^{-2}, \\ h[1] &= h[10] = -0,4506875 \times 10^{-1}, \\ h[2] &= h[9] = +0,6916942 \times 10^{-1}, \\ h[3] &= h[8] = -0,5538437 \times 10^{-1}, \\ h[4] &= h[7] = -0,6342841 \times 10^{-1}, \\ h[5] &= h[6] = +0,5789240 \times 10^0, \end{aligned}$$



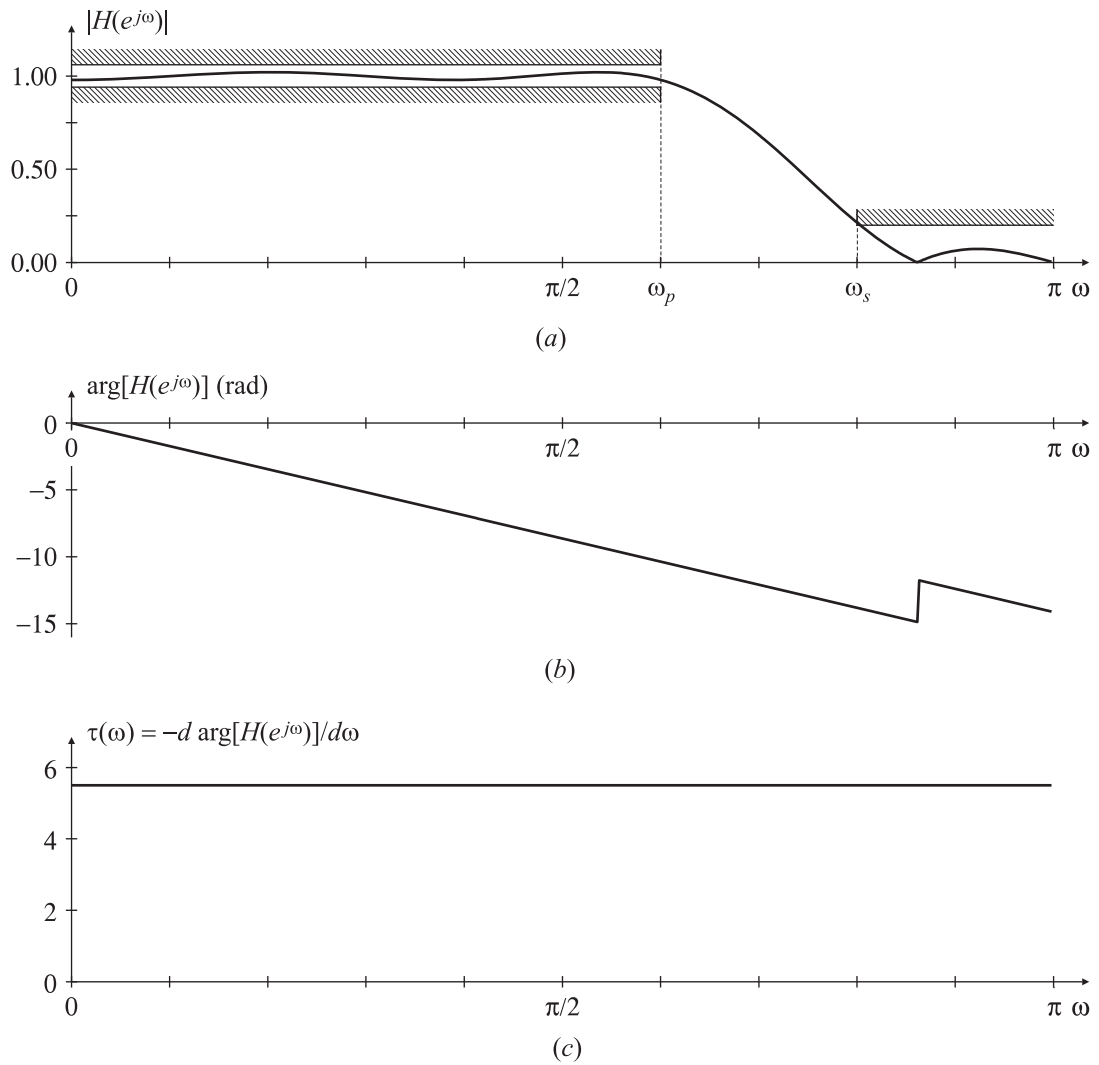
**Fig. 7.2.** Representación en diagrama bloque del filtro IIR del Ejemplo 7.1.



**Fig. 7.3.** Respuesta en frecuencia del filtro IIR (7.4): módulo (a), fase (b) y retardo de grupo (c).



**Fig. 7.4.** Representación en diagrama bloque del filtro FIR del Ejemplo 7.1.



**Fig. 7.5.** Respuesta en frecuencia del filtro FIR (7.5): módulo (a) , fase (b) y retardo de grupo (c) .

La Fig. 7.4 muestra el esquema de una posible implementación, cuya ecuación a diferencias es

$$y[n] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + \cdots + h[10]x[n-10] + h[11]x[n-11]. \quad (7.5)$$

La respuesta en frecuencia (módulo, fase y retardo de grupo) se ilustra en la Fig. 7.5.  $\square$

Del examen de ambas implementaciones se observa que los requerimientos de cómputo y de lugares de almacenamiento para los dos filtros son:

	FIR	IIR
Número de multiplicaciones:	12	5
Número de sumas:	11	4
Lugares de memoria (coeficientes y datos):	24	8

Es evidente que el filtro IIR es más “económico” en la cantidad de operaciones y de lugares de almacenamiento necesarios que el filtro FIR. Se podría haber explotado la simetría de los coeficientes del filtro FIR para reducir a la mitad la cantidad de memoria necesaria, a costa de dificultar ligeramente la implementación. En este caso el número de coeficientes del filtro FIR ( $N = 12$ ) es aproximadamente 6 veces el orden (la mayor potencia de  $z^{-1}$  en el denominador) de la función transferencia del IIR ( $N = 2$ ).

En los dos casos el módulo de la respuesta en frecuencia satisface las especificaciones, pero las respuestas de fase de los dos filtros son muy diferentes. Mientras que la respuesta de fase del filtro IIR no es lineal (Fig. 7.3), la del filtro FIR (Fig. 7.5) es perfectamente lineal, excepto por un salto de  $\pm\pi$  debido al cero en la banda de rechazo. El retardo de fase, que es la derivada negativa de la curva de fase ( $\phi = -d \arg [H(e^{j\omega})] / d\omega$ ), es constante para todo el rango de frecuencia en el caso del filtro FIR. Para el filtro IIR varía en función de la frecuencia de la señal: la respuesta de fase es razonablemente lineal para bajas frecuencias, pero se aparta de la linealidad para la frecuencias próximas a la de corte. El retardo del filtro IIR es mucho menor que el del FIR, como se observa en la Fig. 7.3 y en la Fig. 7.5 (1/2 muestra vs. 5.5 muestras, respectivamente).

## 7.4. La comparación FIR-IIR en el 2007

Las características generales de los filtros recursivos (IIR) y no recursivos (FIR) han variado desde la década del 70 hasta la actualidad, como muestra la Tabla 7.1. En 1970 la aplicación fundamental era el filtrado selectivo en frecuencia y los filtros IIR ofrecían mayores ventajas, sobre todo porque se habían obtenido métodos de diseño analíticos, óptimos, en forma cerrada.

Cuando la aplicación de los filtros digitales se extendió a otros campos (filtros acoplados y filtros adaptivos) el diseño óptimo de los IIR dejó de ser práctico, y los FIR se volvieron una elección natural, en especial porque la teoría y las técnicas de diseño de los filtros adaptivos están bien desarrolladas solamente para filtros FIR. También aparecieron nuevas aplicaciones en las que era necesario preservar la forma de onda de la señal de salida, lo que impone que el filtro tenga una respuesta de fase lineal. Este requisito se satisface más fácilmente usando filtros FIR. En general, los factores que contribuyeron a la mayor utilización de filtros no recursivos fueron:

- nuevas aplicaciones de filtrado;



**Tabla 7.1:** Comparación entre filtros recursivos (IIR) y no recursivos (FIR) a lo largo del tiempo. Las zonas grisadas indican las áreas donde una de las implementaciones es superior a la otra.

Propiedad	1970		1980		1990	
	FIR	IIR	FIR	IIR	FIR	IIR
métodos de diseño	subóptimo usando ventanas	óptimo, analítico, forma cerrada	óptimo usando métodos iterativos	óptimo, analítico, forma cerrada	óptimo usando métodos iterativos	óptimo, analítico, forma cerrada
lugares de memoria-multiplicaciones necesarias	muchos	pocos	más	menos	más	menos
¿pasatodos exactos?	no	sí	no	sí	no	sí
¿inestables?	no	si polos $p_i$ ( $ p_i  > 1$ )	no	si polos $p_i$ ( $ p_i  > 1$ )	no	no
¿zona muerta?	no	sí	no	sí	no	no
¿fase lineal?	sí	no	sí	no	sí	sí
¿transformador de Hilbert?	poco eficiente	no causal	poco eficiente	no causal	poco eficiente	eficiente
¿adaptivo?	sí	no	sí	difícil o imposible	sí	difícil o imposible
¿posibilidad de paralelismo?			muchas	pocas	muchas	muchas

- requerimientos de fase lineal;
- la disponibilidad de los primeros métodos de optimización: los FIR seguían necesitando más operaciones por muestra que los IIR, pero la diferencia de eficiencia entre ambos era menor;
- aparición de técnicas para trabajar con filtros FIR con respuesta impulsiva de gran longitud (método de convolución rápida, propuesto por Stockham en 1966);
- la disminución del costo de la memoria y la disponibilidad de multiplicadores por hardware;
- la posibilidad de efectuar cálculos en paralelo: si bien ambos tipos de filtros se pueden adaptar para el cálculo en paralelo, el paralelismo es más limitado para los IIR si no se puede efectuar un producto en el intervalo de tiempo entre dos muestras.

Como resultado de esta evolución, para 1980 las preferencias se habían volcado hacia los filtros no recursivos (FIR). Pero como muestra la Tabla 7.1, algunas de las características que favorecieron a los FIR durante 1980 se neutralizaron en la década del 90. Lo que cambió para los IIR es que en la actualidad se pueden diseñar para ser siempre estables y también para tener una respuesta de fase lineal. Las implementaciones recientes son

más adecuadas para la paralelización. Según algunos autores (Rader, 2006) muchos de los “problemas” de los IIR en la actualidad están sobrestimados.

### 7.4.1. Los filtros recursivos en la actualidad

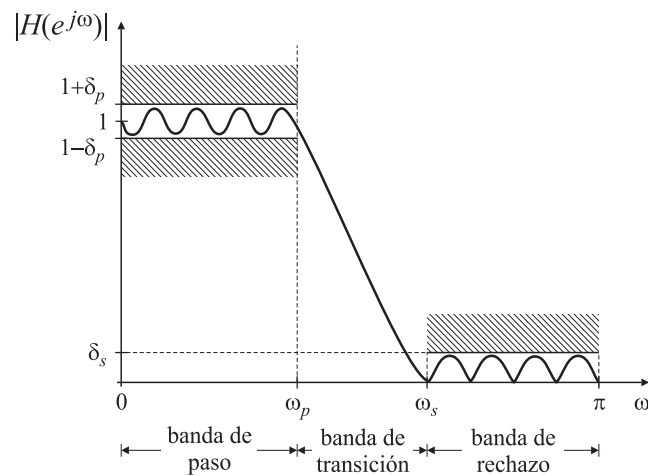
En alguna literatura se considera que los filtros IIR tienen problemas de inestabilidad, respuesta de fase no lineal, paralelismo limitado, y que son inapropiados para ciertas aplicaciones. Sin embargo, la mayoría de estas limitaciones pueden salvarse si se efectúa procesamiento por bloques, y se acepta el error despreciable de truncar la respuesta impulsiva. Un filtro recursivo puede aproximar una respuesta impulsiva deseada con error arbitrariamente pequeño. En teoría, la respuesta impulsiva de un filtro recursivo es infinitamente larga (de ahí el nombre IIR), pero en la práctica la respuesta decae exponencialmente rápido a cero y se la puede truncar después de algunos cientos de muestras cuando ha decaído a valores insignificantes, manteniendo la cota de error. El comportamiento es equivalente a filtrar la señal con un filtro FIR de orden elevado, pero con muchas menos operaciones por muestra que las que serían necesarias para un filtro FIR de la misma longitud.

La señal de entrada se parte en bloques de longitud  $L$ , mucho mayor que la longitud  $P$  de la respuesta impulsiva (truncada) del filtro. Cada bloque de entrada se filtra usando la ecuación a diferencias recursiva, y el resultado se trunca a longitud  $L + P - 1$ . Los bloques de salida sucesivos se concatenan como en el método *overlap-add* solapando las primeras  $P - 1$  muestras de la etapa actual con las últimas  $P - 1$  muestras de la etapa anterior. La eficiencia computacional aumenta, pues mientras que el cálculo de cada muestra de salida necesita  $P$  operaciones en un FIR de longitud  $P$ , la aproximación de su respuesta impulsiva con un filtro IIR de orden  $N$  se consigue con  $N \ll P$ . De esta forma el cálculo de cada muestra de salida necesita a lo sumo  $2N \ll P$  operaciones.

El procesamiento por bloques permite además solucionar otro tipo de problemas:

1. **Estabilidad:** El procesamiento por bloques permite que los filtros no sean necesariamente causales. En los filtros causales, los polos fuera del círculo unitario representan inestabilidad, pero si cada bloque de salida se computa iterando la ecuación a diferencias hacia atrás en el tiempo, los polos que están fuera del círculo no implican inestabilidad. De esta forma se pueden tener filtros recursivos con polos en cualquier lugar del plano complejo, excepto sobre el círculo unitario.
2. **Fase lineal:** Si  $H(e^{j\omega})$  es la respuesta en frecuencia de un filtro cualquiera, se puede obtener un filtro de fase lineal (de hecho, de fase nula) con respuesta en frecuencia  $|H(e^{j\omega})|^2$  si el bloque de entrada se filtra dos veces, una de ellas hacia atrás en el tiempo, como se estudió en el Capítulo anterior. Esta idea fue propuesta por Powell y Chau en 1990, pero en su momento no recibió mucha atención.
3. **Paralelismo:** cada bloque puede procesarse en diferentes instancias de hardware.

El procesamiento por bloques permite implementar transformadores de Hilbert muy eficientes (Rader, 1984). Siguiendo la idea de los filtros de fase nula, se utiliza un par de filtros IIR cuya respuesta de fase difiere en  $\pi/2$  radianes. Esta implementación es mucho más eficiente que un diseño FIR óptimo, necesitando de 5 a 10 veces menos multiplicaciones para las mismas especificaciones. Otros filtros, como los adaptivos, todavía no se



**Fig. 7.6.** Bandas de tolerancia para un filtro pasabajos.

pueden implementar fácilmente como IIR aún usando el esquema de filtrado por bloques.

## 7.5. Diseño de filtros

El diseño de un filtro digital involucra los siguientes pasos:

- especificación de los requerimientos del filtro;
- elección de una aproximación conveniente y cálculo de los coeficientes;
- representación del filtro utilizando una estructura adecuada (realización);
- análisis de los efectos de la longitud finita de palabra en el desempeño;
- implementación en hardware o software.

Estos cinco pasos no son necesariamente independientes, ni necesitan seguirse en el orden descrito; actualmente, las técnicas de diseño disponibles combinan el segundo y parte del tercero y cuarto. Para lograr un filtro eficiente es necesario iterar entre las distintas etapas, especialmente si, como es habitual, las especificaciones de diseño dejan cierto grado de libertad al diseñador, o si se desean explorar otras alternativas de diseño.

### 7.5.1. Especificaciones de diseño

Los requerimientos incluyen la especificación de

1. las características de las señales: tipo de fuente de señal, interfaz de entrada-salida, velocidad de procesamiento, ancho de palabra, la mayor frecuencia de interés;
2. las características del filtro: la respuesta en módulo y/o fase deseadas y sus tolerancias, la velocidad de operación, el modo de filtrado (en línea o fuera de línea);

3. la forma de implementación: como una rutina de alto nivel en una computadora, o un programa específico para un DSP;
4. otras restricciones al diseño, como por ejemplo el costo del filtro.

Es posible que inicialmente el diseñador no cuente con toda la información necesaria para especificar completamente el filtro, pero cuanto más detalles se conozcan más sencillo será el proceso de diseño.

Aunque algunos de los requerimientos discutidos arriba son dependientes de la aplicación, es necesario resaltar los aspectos referidos a las *características* del filtro. Frecuentemente, los requisitos del filtro digital se especifican en el dominio frecuencial, y para el caso de los filtros selectivos en frecuencia, estas especificaciones toman la forma de bandas de tolerancia, como muestra la Fig. 7.6 para el caso de un filtro pasabajos. Las zonas grisadas indican los límites de tolerancia. En la banda de paso, el módulo de la respuesta admite una variación máxima de  $\pm\delta_p$ , y en la banda de rechazo se pretende que la ganancia no exceda  $\delta_s$ . Es frecuente especificar estas cotas en dB.

El ancho de la zona de transición determina qué tan abrupto es el filtro. En esta región se espera que el módulo de la respuesta en frecuencia decrezca monótonamente desde la banda de paso a la banda de rechazo. Los principales parámetros de interés son

desviación en la banda de paso:	$\delta_p$ ,
desviación en la banda de rechazo:	$\delta_s$ ,
frecuencia de corte de la banda de paso:	$\omega_p$ ,
frecuencia de corte de la banda de rechazo:	$\omega_s$ .

Las frecuencias esquina se suelen normalizar a la frecuencia de muestreo ( $\omega_i = 2\pi f_i / f_s$ ), pero también son frecuentes las especificaciones en unidades típicas como Hz, kHz, y a menudo son mucho más significativas. Las desviaciones con respecto a la respuesta deseada tanto en la banda de paso como en la banda de rechazo pueden expresarse en valores absolutos o en decibeles (dB), indicando la ondulación máxima ("ripple") tolerada en la banda de paso, y la atenuación mínima exigida en la banda de rechazo:

$$\begin{aligned} A_s \text{ (atenuación en la banda de rechazo):} & \quad -20 \log_{10} \delta_s, \\ A_p \text{ (ondulación en la banda de paso):} & \quad 20 \log_{10} (1 + \delta_p). \end{aligned}$$

En general, la respuesta de *fase* un filtro no se especifica tan detalladamente como el módulo de la respuesta en frecuencia. En muchos casos es suficiente indicar que importa la distorsión de fase, o que se pretende una respuesta de fase lineal. Sólo es necesario detallar la respuesta en fase deseada en aquellas aplicaciones donde los filtros se utilizan para compensar o ecualizar la respuesta en fase de un sistema.

#### EJEMPLO 7.2. Especificaciones de diseño de un filtro pasabanda

Se desea diseñar un filtro FIR pasabanda cuya respuesta en frecuencia satisfaga las siguientes especificaciones:

banda de paso:	$0,18 \times 2\pi$ a $0,33 \times 2\pi$ ,
ancho de la zona de transición:	$0,04 \times 2\pi$ ,
ganancia en la banda de paso:	$1 \pm 0,05$ ( $A_p = 0,42$ dB),
ganancia en la banda de rechazo:	$< 0,001$ ( $A_s = 60$ dB).

En la Fig. 7.7 se muestra el esquema de bandas de tolerancias del filtro. La atenuación en la banda de rechazo debe ser mayor que  $A_s = -20 \log_{10} \delta_s = 60$  dB, y la ondulación máxima tolerada en la banda de paso es de  $A_p = 20 \log_{10} (1 + \delta_p) = 0,42$  dB.  $\square$

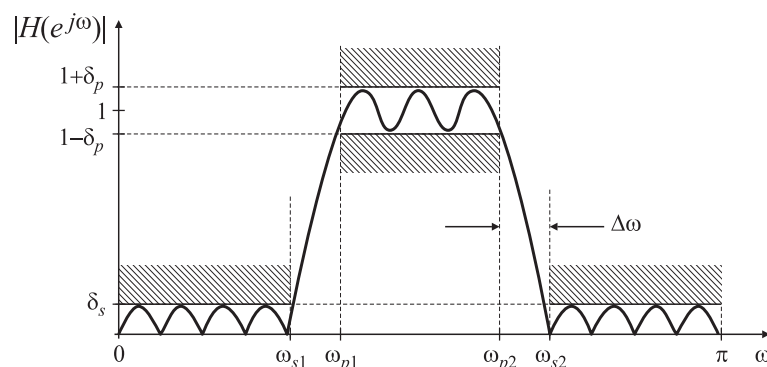


Fig. 7.7. Bandas de tolerancia para el filtro pasabanda del Ejemplo 7.2.

### 7.5.2. Elección de la aproximación y cálculo de los coeficientes

En esta etapa se elige alguno de los métodos de aproximación (Butterworth, Chebyshev, elíptico, etc. si el filtro es IIR, o equiripple, óptimo, con ventanas, etc. si es FIR) y se calculan los valores de los coeficientes  $h[n]$  del filtro FIR, o  $a_k$  y  $b_k$  del filtro IIR, de modo de satisfacer las especificaciones (Sección 7.5.1). El método empleado para determinar los coeficientes es distinto según el filtro sea IIR o FIR.

Tradicionalmente, el cálculo de los coeficientes de los filtros IIR se basa en la transformación de las funciones de sistema de filtros analógicos en filtros discretos. Los dos métodos clásicos son el de *invariación al impulso* y el de *transformación bilineal*. Con la técnica de la invariación al impulso la respuesta impulsiva del filtro digital está formada por las muestras de la respuesta impulsiva del filtro analógico. Sin embargo, el módulo de la respuesta en frecuencia del filtro discreto es diferente, en general, de la del filtro continuo. Debido a fenómenos de aliasing, el método no es apropiado para el diseño de filtros pasaaltos o filtros eliminabanda, o cualquier tipo de filtro cuya respuesta en frecuencia no sea aproximadamente nula por encima de cierta frecuencia  $f_H$ .

Por otra parte, el método de la transformada bilineal produce filtros muy eficientes, y es apropiado para el diseño de cualquier tipo de filtro con ganancia constante en la banda de paso. Permite el diseño de filtros con características frecuenciales clásicas como Butterworth, Chebyshev, elípticos, etc. Los filtros digitales calculados en base a la transformada bilineal preservan la característica de respuesta en frecuencia, pero no las propiedades temporales. En la actualidad se cuenta con programas de cálculo que permiten obtener los coeficientes del filtro digital directamente a partir de las especificaciones usando el método de la transformada bilineal o la técnica analítica desarrollada por Rader y Gold (1967) para filtros recursivos equiripple.

Los filtros con respuestas sencillas, como los resonadores, "notch", filtros peine, osciladores digitales, etc., se pueden diseñar ubicando directamente los polos y los ceros en el plano complejo; sin embargo, si las especificaciones son exigentes, es conveniente utilizar las técnicas de diseño clásicas.

Los coeficientes de los filtros FIR pueden calcularse con varios métodos. Entre ellos merecen destacarse el diseño por ventanas, la técnica de suavizado de las bandas de transición, el método de muestreo en frecuencia, y los métodos óptimos (el algoritmo de Parks y McClellan o el algoritmo de Remez). El método de diseño utilizando ventanas es sen-

**Tabla 7.2:** Métodos típicos para el diseño de filtros IIR y FIR.

IIR	FIR
invariación al impulso	ventanas
transformada bilineal	bandas de transición suaves
ubicación directa de polos y ceros	muestreo en frecuencia
óptimos	óptimos

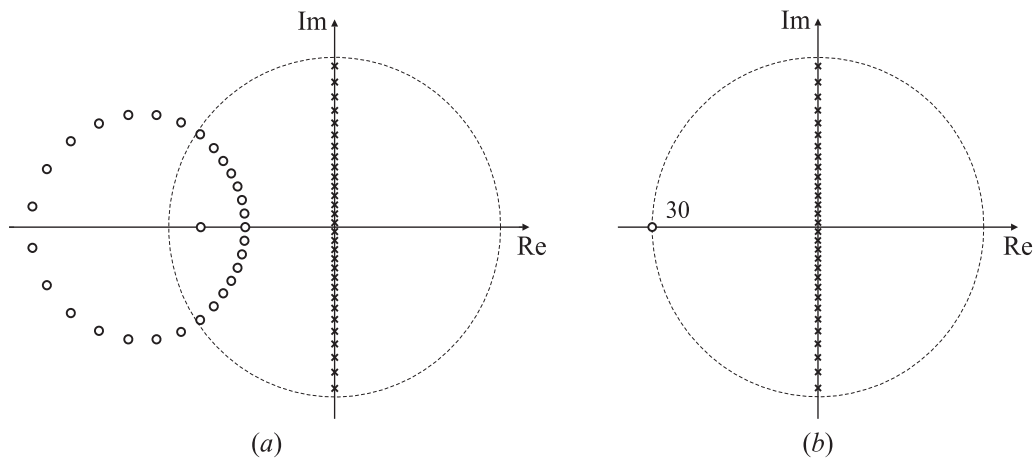
cillo y flexible pero no permite un control muy preciso sobre los parámetros del filtro. La técnica de las bandas de transición suaves permite atenuar el efecto Gibbs, a costa de una pequeña complejidad adicional. El método de muestreo en frecuencia permite una implementación particularmente eficiente (y recursiva) de filtros FIR, pero puede tener problemas cuando se usa aritmética de punto fijo. Con la disponibilidad de programas de diseño eficientes y con buena interfaz con el usuario, los métodos óptimos son ampliamente utilizados hoy en día, y para la mayoría de las aplicaciones permiten obtener rápidamente el filtro FIR deseado.

Los métodos típicos para el cálculo de los coeficientes de los filtros se resumen en la Tabla 7.2. El método de diseño se elige de acuerdo a la aplicación en particular. Si bien influyen varios factores, el más importante es qué tan críticas son las especificaciones. La decisión “difícil” es optar entre FIR o IIR. En aquellos casos en que las propiedades de los FIR (respuesta de fase estrictamente lineal, estabilidad inherente) son imprescindibles, la mejor elección puede ser el diseño por métodos óptimos, o usando ventanas (generalmente la de Kaiser). Si, en cambio, son deseables las características de los IIR (menor cantidad de coeficientes para especificaciones similares) el método de la transformada bilineal es apropiado para la mayoría de los casos.

### 7.5.3. Realización

Por *realización* se entiende convertir una función de sistema  $H(z)$  en un conjunto de ecuaciones a diferencias que se ejecutarán como un algoritmo en un procesador. El conjunto de ecuaciones también se conoce como la *estructura* del filtro. Existen infinitas formas de escribir la ecuación a diferencias, de modo que existen infinitos tipos de estructuras. Matemáticamente las distintas realizaciones producen el mismo resultado sobre la señal a filtrar; sin embargo en el momento de la implementación la sensibilidad a la variación de los coeficientes, propagación del ruido de redondeo o de truncación, organización y/o requerimientos de memoria en un procesador en particular, etc., suelen decidir la elección de una estructura particular. Cada realización se asocia a un diagrama bloque como los de las Figs. 7.2 y 7.4, que revela la cantidad de lugares de memoria necesarios para el almacenamiento de los datos y los coeficientes, número de operaciones requeridas, etc.

Es frecuente utilizar distintas estructuras para implementar filtros IIR o FIR. Para filtros IIR, las tres estructuras típicas son las *formas directa, cascada y paralelo*; el diagrama bloque de una *forma directa*, denominada de “tipo II” es la que se muestra en la Fig. 7.2 para un sistema de segundo orden. En este caso, la función transferencia (9.4) se implementa con el conjunto de ecuaciones a diferencia (9.5); la Fig. 7.2 muestra que se necesitan dos lugares de memoria para almacenar la historia del filtro (los cuadrados señalados con  $z^{-1}$ )



**Fig. 7.8.** Comparación de los patrones de polos y ceros de una implementación en *forma directa I* (a) y en *secciones de segundo orden*, cada una de ellas en *forma directa I* (b).

y cinco lugares de memoria para almacenar los coeficientes  $(a_1, a_2, b_0, b_1, b_2)$ . Sin embargo, las estructuras cascada y paralelo, que se estudiarán en el Capítulo 13, son las más utilizadas para la implementación de filtros IIR pues son mucho más insensibles a los efectos de cuantización de coeficientes que la estructura directa. Ésta presenta una alta sensibilidad a la variación de los coeficientes, especialmente para filtros de alto orden, como se muestra en el siguiente EJEMPLO.

### EJEMPLO 7.3. Influencia de la estructura con coeficientes de precisión “infinita”

Aún cuando los coeficientes se implementen con muy alta precisión, como cuando se usa MATLAB, la estructura directa es inadecuada para realizar filtros de alto orden. Por ejemplo, un filtro tipo Butterworth discreto de orden  $N = 30$  con frecuencia de corte en  $\omega_c = \pi/2$  se caracteriza por tener 30 ceros en  $z = -1$ . La implementación en forma directa, utilizando los comandos

```
[n, d] = butter(30, 0.5);
zplane(n, d);
```

da resultados muy diferentes, como se puede ver en la Fig. 7.8(a). Por efecto de los (pequeños) errores numéricos, los 30 ceros que deberían estar ubicados en  $z = -1$ , se esparcen en un entorno, no necesariamente pequeño, de este punto. La implementación como secciones de segundo orden, que se diseña con los comandos

```
[z, p, k] = butter(30, 0.5);
[sos, g] = zp2sos(z, p, k);
hsos = dfilt.df1sos(sos, g);
zplane(hsos);
```

tiene la distribución de polos y ceros prevista por la teoría, como se observa en la Fig. 7.8(b). □

Para los filtros FIR (7.2) la forma más utilizada es la directa, Fig. 7.4, porque es sencilla de implementar y aprovecha al máximo la arquitectura del hardware de la mayoría de los procesadores digitales de señales (DSP). Esta estructura también se conoce como filtro transversal o “línea de retardo” (*tapped delay line*) pues almacena los valores pasados de las muestras de la señal de entrada. Como se estudiará más adelante, esta implementación es mucho menos sensible a la cuantización de los coeficientes que la forma directa de los filtros IIR. Otras estructuras de uso frecuente son las de muestreo en frecuencia y la de convolución rápida, que se estudiarán en el Capítulo 13.



Sintetizando, para un filtro dado la elección de la estructura depende:

- del tipo de filtro (IIR o FIR);
- de la facilidad de implementación;
- de la sensibilidad de la estructura a los efectos de la longitud finita de palabra.

#### 7.5.4. Efectos de la longitud de palabra finita

Los pasos de diseño referidos a la aproximación y a la realización suponen que las operaciones que especifican los algoritmos se realizan con precisión infinita, o al menos muy alta. Sin embargo, en muchas aplicaciones donde el bajo costo es decisivo, es necesario representar los coeficientes del filtro utilizando un número finito de bits (típicamente 8, 16 o más), y realizar las operaciones indicadas por las ecuaciones a diferencia utilizando aritmética de precisión finita.

El uso de un número finito de bits degrada el desempeño del filtro, y en algunos casos puede inutilizarlo por completo. El diseñador debe tener en cuenta estos efectos, y elegir longitudes de palabra apropiados para los coeficientes del filtro, las variables de entrada y salida, y las operaciones aritméticas. En general, el ancho en bits de las palabras de entrada y salida (que se leen o se escriben en los conversores A/D y D/A, respectivamente) son mucho menores que la longitud de palabra utilizado para los coeficientes y los cálculos (10 o 16 bits, frente a 25 o 32 bits).

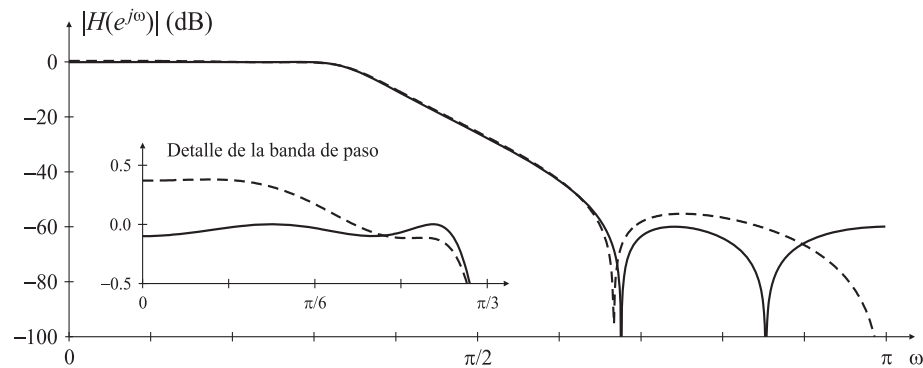
Las principales causas de la disminución del desempeño en los filtros digitales debido a la longitud finita de palabra son:

- *Cuantización de las señales de entrada y salida*, debido a la resolución finita de los conversores A/D y D/A. Este efecto se suele modelar como un fenómeno aleatorio (ruido), tal como se estudió en el Capítulo 7.
- *Cuantización de los coeficientes*, lo que conduce a variaciones de la respuesta en frecuencia de filtros FIR e IIR, y posiblemente a inestabilidades en los filtros IIR.
- *Errores aritméticos por redondeo*. Los procesadores con longitud de palabra finita tienen un acumulador con bits adicionales para mantener la precisión (por ejemplo, el resultado de la multiplicación de dos números de  $B$  bits es un número de  $2B$  bits). Cuando estos resultados se almacenan en memoria se truncan a la longitud de palabra adoptada. El error resultante, denominado *ruido de redondeo*, altera la respuesta del filtro. En los filtros IIR puede causar inestabilidad, y también dar lugar oscilaciones de pequeña amplitud de la salida aún con entrada nula.
- *Desborde (overflow)*, que ocurre cuando el resultado de una suma excede el ancho de palabra adoptado, resultando en errores de las muestras de salida, que pueden causar inestabilidad, u oscilaciones de gran amplitud en los filtros IIR.

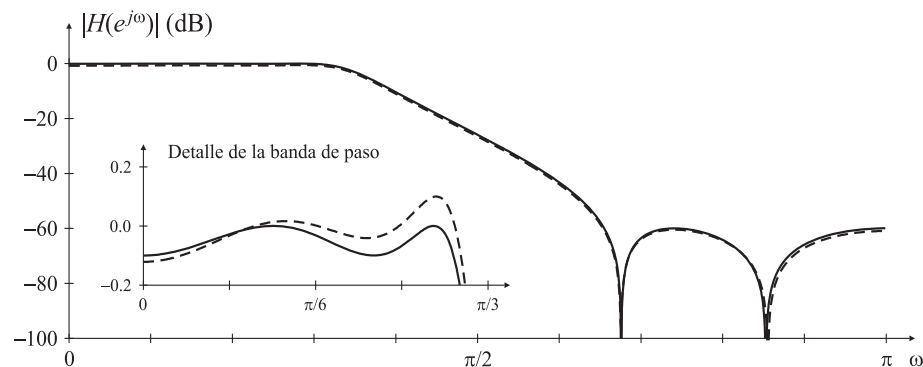
La degradación del desempeño depende de:

- la longitud de palabra y tipo de aritmética utilizada para efectuar los cálculos;
- el método adoptado para cuantizar las variables y los coeficientes del filtro a la longitud de palabra elegida;





**Fig. 7.9.** Respuesta en frecuencia del filtro del EJEMPLO 7.4 implementado con una estructura tipo forma directa II. (—) filtro prototipo; (---) filtro con coeficientes cuantizados.



**Fig. 7.10.** Respuesta en frecuencia del filtro del EJEMPLO 7.4 implementado con una cascada de dos bloques de segundo orden. (—) filtro prototipo; (---) filtro con coeficientes cuantizados.

- la estructura del filtro.

Conociendo estos factores, el diseñador puede analizar los efectos de la longitud de palabra finita en el desempeño del filtro, y efectuar las correcciones que sean necesarias.

#### EJEMPLO 7.4. Efectos de cuantización en un filtro IIR

Un filtro IIR elíptico de cuarto orden con frecuencia de corte en  $\omega_c = 0,3\pi$ , 0,1 dB de ondulación en la banda de paso, y 60 dB de atenuación en la banda de rechazo está caracterizado por la función de sistema

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}},$$

donde

$$\begin{aligned} b_0 &= 0,01588232577583, & a_0 &= 0,57816706294199, \\ b_1 &= 0,04509870267465, & a_1 &= -1,00000000000000, \\ b_2 &= 0,06160127928264, & a_2 &= 0,98785160189871, \\ b_3 &= 0,04509870267465, & a_3 &= -0,49030269551500, \\ b_4 &= 0,01588232577583, & a_4 &= 0,10997293011303. \end{aligned}$$

El módulo de la respuesta en frecuencia del filtro se muestra con línea de trazo continuo en la Fig. 7.9. Si el filtro se implementa en un procesador de 8 bits, utilizando una estructura tipo forma directa II se deben truncar los coeficientes, que resultan

$$\begin{aligned} b_0 &= 0,0156250, & a_0 &= 0,5781250, \\ b_1 &= 0,0468750, & a_1 &= -1,0000000, \\ b_2 &= 0,0625000, & a_2 &= 0,9843750, \\ b_3 &= 0,0468750, & a_3 &= -0,4921875, \\ b_4 &= 0,0156250, & a_4 &= 0,1093750. \end{aligned}$$

El módulo de la respuesta en frecuencia del filtro con los coeficientes cuantizados, graficada con línea de trazos en Fig. 7.9, revela que se violan las especificaciones de diseño. Sin embargo, la elección de una estructura diferente puede solucionar el problema. En la Fig. 7.10 se muestra el módulo de la respuesta en frecuencia del mismo filtro, implementado como una cascada de dos secciones de segundo orden, con ganancia  $C = 0,0274701$ , y cuyos coeficientes cuantizados a 8 bits son

$$\begin{aligned} b_{01} &= 0,5000000, & a_{01} &= 0,5000000, \\ b_{11} &= 0,8984375, & a_{11} &= -0,4531250, \\ b_{21} &= 0,5000000, & a_{21} &= 0,1406250, \\ b_{02} &= 0,5000000, & a_{02} &= 0,5000000, \\ b_{12} &= 0,5234375, & a_{12} &= -0,4140625, \\ b_{22} &= 0,5000000, & a_{22} &= 0,3437500. \end{aligned}$$

La figura revela que los requisitos de diseño se satisfacen sin necesidad de cambiar la resolución en bits de los coeficientes.  $\square$

El ejemplo anterior muestra que los efectos de la cuantización de los coeficientes pueden atenuarse cambiando la estructura del filtro. Dependiendo de la implementación, algunos de los errores mencionados pueden ser insignificantes. Por ejemplo, si el filtro se calcula en un lenguaje de alto nivel que se ejecuta en computadoras más o menos modernas, por ejemplo con MATLAB en una PC, los errores de cuantización de coeficientes y redondeo no siempre son importantes. En aplicaciones de tiempo real las señales de entrada y salida se cuantizan a 8, 12 o 16 bits, según el tipo de conversor A/D y D/A utilizado. Si el ancho de palabra elegido para efectuar la operaciones y el redondeo es de esta magnitud, es casi obligatorio analizar los efectos de la cuantización en las características del filtro.

### 7.5.5. Implementación

Una vez que se han calculado los coeficientes del filtro, elegido una estructura adecuada, y verificado que las especificaciones se satisfacen después de considerar la degradación causada por la cuantización de coeficientes y de las variables a la longitud de palabra elegida, la ecuación a diferencias debe implementarse como una rutina de software o en hardware. Cualquiera sea el método de implementación, cada muestra de la salida  $y[n]$  del filtro debe calcularse de acuerdo a la ecuación a diferencias que involucra multiplicaciones, sumas y restas, y retardos. De modo que para implementar un filtro se necesitan los siguientes bloques básicos:

- memoria (ROM o RAM) para almacenar los coeficientes del filtro;

- memoria RAM para almacenar las muestras actuales y pasadas de la entrada  $x[n]$ ,  $x[n-1]$ , ...,  $x[n-M]$ , y la salida  $y[n]$ ,  $y[n-1]$ , ...,  $y[n-N]$ ;
- multiplicadores en hardware o rutinas de multiplicación (bibliotecas de software);
- sumador o unidad aritmético-lógica.

La manera en que se configuran estos bloques depende fuertemente según el procesamiento sea en tiempo real o fuera de línea. Cuando se procesa fuera de línea, los datos o señal de entrada están disponibles en algún medio de almacenamiento (cinta, CD, memoria, etc.) Tal es el caso del registro de datos experimentales para el análisis posterior. En estos casos el filtro se implementa utilizando un lenguaje de alto nivel, y se ejecuta en una computadora de propósito general. Por ello las implementaciones fuera de línea suelen ser programas. En algunas aplicaciones, por ejemplo el procesamiento de imágenes, se agrega hardware adicional dedicado para acelerar la velocidad de procesamiento.

En el procesamiento en tiempo real se requiere que el filtro genere una muestra de la señal de salida ante cada muestra de la señal de entrada, lo que implica que el cómputo completo de la ecuación a diferencias debe completarse en el intervalo de tiempo entre una y otra muestra de la señal de entrada, o bien operar sobre *bloques* de datos utilizando las técnicas tipo *overlap-add* u *overlap-save*. El procesamiento en tiempo real necesita de hardware dedicado cuando la frecuencia de muestreo es muy alta, o cuando el filtro es de alto orden. Este hardware se nuclea alrededor de un procesador digital de señales, o DSP, que en la actualidad es producido por varios fabricantes, como Motorola, Texas Instruments, Analog Devices, Burr Brown, etc. Cada DSP incluye memoria RAM y ROM o FLASH, multiplicadores rápidos, unidades aritmético lógicas flexibles, métodos de direccionamiento poderosos, etc. y algunas unidades incluyen los conversores A/D y D/A necesarios para procesar señales del mundo exterior.

## 7.6. Referencias

- Ifeachor, E., Jervis, B. (1993) *Digital Signal Processing: a practical approach*, Addison-Wesley Longman Ltd, Edingburg Gate, Harlow, England.
- Powell, S. R., Chau, P. M. (1990) "The reversed filtering in real time", *Proc. IEEE Int. Symp. Circuits Systems*, May 1990, Vol. 2, pp. 1239-1243.
- Rader, C. M., Gold, B. (1967) "Digital filter design in the frequency domain", *Proc. IEEE*, Vol. 55, No. 2, pp. 149-171.
- Rader, C. M. (1984) "A simple method for sampling in-phase and quadrature components", *IEEE Trans. Aerospace Electr. Syst.*, Vol. AES 20, No. 6, pp. 821-824.
- Rader, C. M. (2006) "The rise and fall of recursive digital filters", *IEEE Signal Proc. Magazine*, Vol. 23, No. 6, pp. 46-49.
- Stockham, T. G. (1966) "High speed convolution and correlation", *Proc. AFIPS*, No. 28, pp. 229-233.