



**FACULTAD
DE INGENIERIA**
Universidad de Buenos Aires

Diseño de filtros en tiempo discreto

Dr. Ing. Leonardo Rey Vega

Señales y Sistemas
Facultad de Ingeniería
Universidad de Buenos Aires

Marzo 2024

- 1 Filtros en tiempo discreto
- 2 Diseño de filtros IIR
- 3 Diseño de filtros FIR

Las aplicaciones de filtros en tiempo discreto son muchísimas:

- Sistemas de comunicaciones.
- Aplicaciones de ingeniería biomédicas.
- Procesamiento de habla.
- Procesamiento digital de audio.
- Procesamiento de imágenes.
- Técnicas de procesamiento de señales avanzadas: bancos de filtros, wavelets, etc.
- Técnicas de estimación de procesos estocásticos.
- Técnicas de beamforming y antenas inteligentes.

Las etapas de diseño de un filtro en tiempo discreto son:

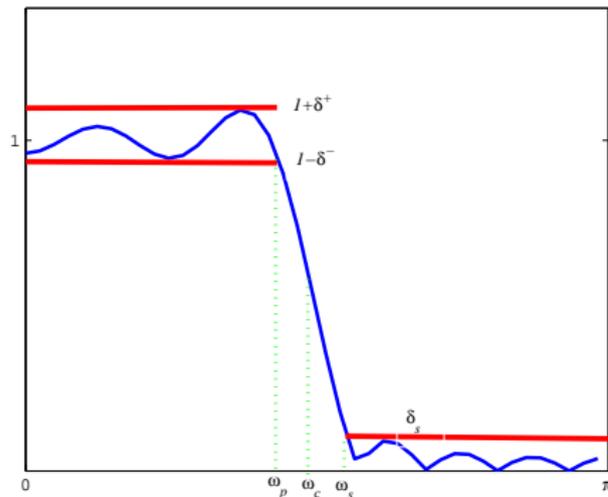
- 1 Especificación de las propiedades deseadas (dependiente de la aplicación).
- 2 Aproximación de las especificaciones mediante un sistema en tiempo discreto.
- 3 Realización del sistema.

Nos ocuparemos de la etapa número 2

La aproximación se hará mediante funciones transferencias racionales, estables y causales:

- Filtros IIR.
- Filtros FIR.

Especificaciones de filtros en tiempo discreto



Se especifican:

- Frecuencia de paso: ω_p .
- Frecuencia de atenuación: ω_s
- Ripple en la banda de paso

$$\delta_p = \max \{ \delta^+, \delta^- \}$$

ó

$$A_p = 20 \log_{10} (1 \pm \delta_p)$$

- Ripple en la banda de atenuación: δ_s ó

$$A_s = -20 \log_{10} \delta_s$$

Características de filtros IIR

Ventajas:

- Las técnicas más populares de diseño usan filtros en tiempo continuo como prototipos.
- Fórmulas que dan los coeficientes de los filtros en forma cerrada.
- Especificaciones exigentes llevan a filtros con un número pequeño de coeficientes lo que implica bajo costo computacional.

Ventajas:

- Las técnicas de diseño usan filtros en tiempo continuo como prototipos.
- Fórmulas que dan los coeficientes de los filtros en forma cerrada.
- Especificaciones exigentes llevan a filtros con un número pequeño de coeficientes lo que implica bajo costo computacional.

Desventajas:

- Es necesario tener especial cuidado si se trabaja en precisión finita.
- No es posible tener filtros con fase lineal en forma exacta.

Dado que la idea los métodos de diseño de filtros IIR en tiempo discreto es usar prototipos en tiempo continuo debemos tener en cuenta los siguiente:

- Dado que nuestras especificaciones son para el filtro en tiempo discreto debemos ser capaces de trasladar dichas especificaciones a tiempo continuo para diseñar el prototipo.
- Al pasar del filtro de tiempo continuo al filtro en tiempo discreto debemos mantener la causalidad y estabilidad.

Los métodos usuales de diseño son:

- Invarianza al impulso.
- Transformación bilineal.

Invarianza al impulso I

La idea básica es obtener la respuesta al impulso en tiempo discreto muestreando la respuesta al impulso del prototipo en tiempo continuo:

$$h(n) = Th_c(nT)$$

En el campo transformado:

$$H(e^{j\omega}) = \sum_{k=-\infty}^{\infty} H_c\left(\frac{\omega}{T} + \frac{2\pi k}{T}\right)$$

Si el filtro en tiempo continuo es de banda limitada:

$$H_c(\Omega) = 0, \quad |\Omega| \geq \frac{\pi}{T}$$

$$H(e^{j\omega}) = H_c\left(\frac{\omega}{T}\right), \quad |\omega| \leq \pi$$

Invarianza al impulso II

Debido al aliasing inherente de la técnica y a la imposibilidad de obtener filtros en tiempo continuo de banda limitada se debe tener especial cuidado en el diseño.

Muchas veces se tiene que $H_c(\Omega) \approx 0$ si $|\Omega| \geq \pi/T$ con lo se puede asumir sin muchos problemas que:

$$H(e^{j\omega}) \approx H_c\left(\frac{\omega}{T}\right), \quad |\omega| \leq \pi$$

Es importante notar que:

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



Esta ecuación básica nos permite obtener las especificaciones del filtro prototipo en tiempo continuo a partir de las correspondientes al filtro en tiempo discreto.

Las especificaciones relacionadas con los ripples en la banda de paso y de atenuación permanecen invariantes (sólo si el aliasing es despreciable!).

Invarianza al impulso III

Método de diseño:

- 1 Sean las especificaciones del filtro en tiempo discreto. Para un simple pasabajos: ω_s , ω_p , A_s y A_p . Elegimos T (ej: $T = 1$.)
- 2 Obtenemos las especificaciones en tiempo discreto: $\Omega_p = \omega_p/T$, $\Omega_s = \omega_s/T$ (suponemos que el aliasing es despreciable.)
- 3 Generamos la transferencia racional del filtro de tiempo continuo prototipo $H_c(s)$ y la escribimos como $H_c(s) = \sum_{k=1}^N \frac{A_k}{s-a_k}$.
- 4 Dado que $h_c(t) = \sum_{k=1}^N A_k e^{a_k t} u(t)$ ($\Re\{a_k\} < 0$ ya que $H_c(s)$ es estable):

$$h(n) = Th_c(nT) = \sum_{k=1}^N TA_k e^{a_k T n} u(n) \longrightarrow H(z) = \sum_{k=1}^N \frac{TA_k}{1 - e^{a_k T} z^{-1}}$$

Es claro que si $H_c(t)$ es estable $H(z)$ también lo será.

Transformación bilineal I

Esta técnica elimina el problema del aliasing introduciendo una compresión del eje de frecuencias. Para ello se usa la siguiente transformación:

$$s = \frac{2}{T} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right)$$


Esta es una transformación conforme:

- El semiplano izquierdo en s se transforma en el interior de $|z| = 1$ y viceversa.
- El eje $j\Omega$ se mapea en el círculo $|z| = 1$.

Se conservan la causalidad y la estabilidad!

El filtro en tiempo discreto:

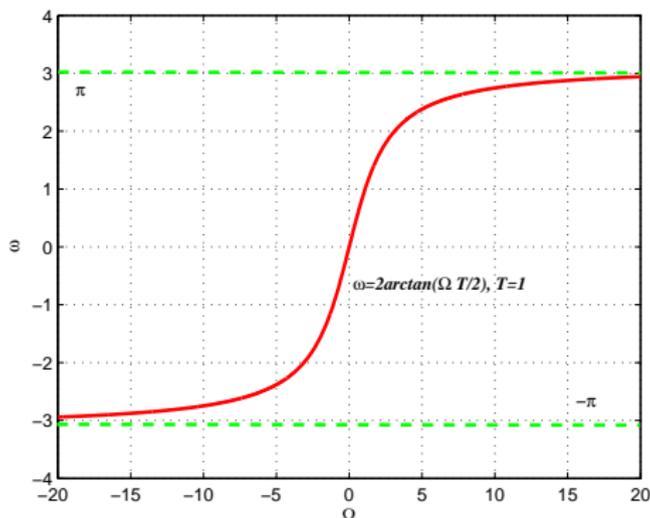
$$H(z) = H_c \left[\frac{2}{T} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) \right]$$


Se ve que no existe el problema del aliasing! Se puede usar este método para diseñar filtros pasabanda y pasaaltos!

Transformación bilineal II

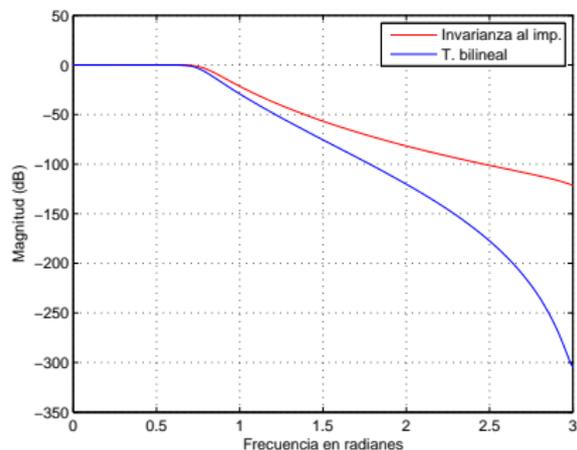
Es posible probar que:

$$\Omega = \frac{2}{T} \tan\left(\frac{\omega}{2}\right), \quad \omega = 2 \arctan\left(\frac{\Omega T}{2}\right)$$



La compresión del eje de frecuencias puede traer problemas para algunos diseños particulares. Sin embargo para diseños que requieren bandas con ganancias y atenuaciones constantes no suele haber problemas.

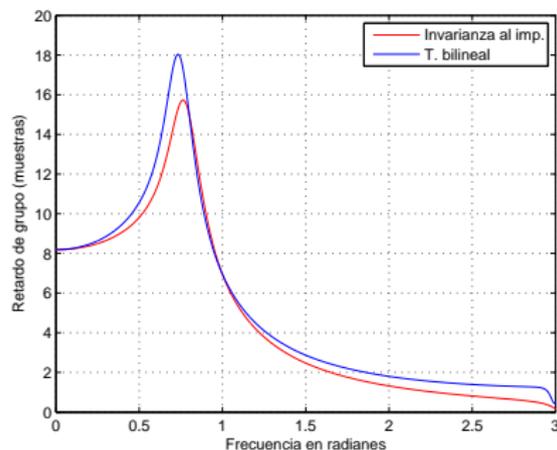
Ejemplo



Filtro pasabajo con especificaciones:

$$0,95 \leq |H(e^{j\omega})| \leq 1 \quad 0 \leq \omega \leq 0,25\pi$$

$$|H(e^{j\omega})| \leq 0,05 \quad 0,35\pi \leq \omega \leq \pi$$



Diseño usando $T = 1$ con un filtro prototipo tipo Butterworth con $N = 9$ y $\Omega_c = 0,7808$. Para ambos diseños el número de coeficientes en numerador y denominador es menor o igual a 11.

Ventajas:

- Filtros que son inherentemente estables.
- Es posible diseñarlos con fase lineal.
- Generalmente son suficientemente robustos cuando son implementados en precisión finita.

Ventajas:

- Filtros que son inherentemente estables.
- Es posible diseñarlos con fase lineal.
- Generalmente son suficientemente robustos cuando son implementados en precisión finita.

Desventajas:

- Son computacionalmente más complejos que un filtro IIR con las mismas especificaciones.
- No siempre es posible obtener los coeficientes en forma cerrada.

Teorema

Un filtro $h(n)$ real, causal, estable y con transferencia racional es de fase lineal sí y sólo sí es FIR y

$$h(n) = \pm h(N - n), \quad H(z) = \sum_{n=0}^N h(n)z^{-n}.$$

Existen 4 clases de sistemas con fase lineal:

- Tipo I: $h(n) = h(N - n)$ con N par.
- Tipo II: $h(n) = h(N - n)$ con N impar.
- Tipo III: $h(n) = -h(N - n)$ con N impar.
- Tipo IV: $h(n) = -h(N - n)$ con N par.

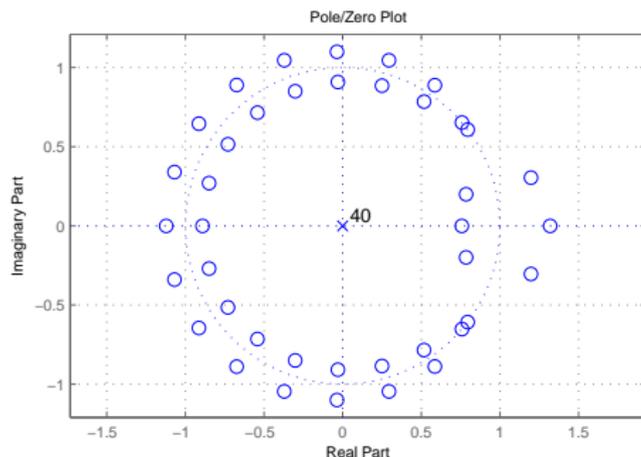
Filtros FIR de fase lineal II

Es posible probar que para los sistemas FIR de fase lineal:

$$H(z) = \pm z^{-N} H(z^{-1})$$



Si z_0 es cero entonces $1/z_0$ es cero!!



- Para $z = -1$ y sistemas tipo I y II $H(-1) = (-1)^N H(-1)$.

- Métodos de ventaneo.
- Diseño por cuadrados mínimos.
- Diseños equirriple.
- Eigenfilters.

Todas estas técnicas permiten obtener diseños con fase lineal.

Estas técnicas pueden ser modificadas para obtener respuestas de filtros multibanda y otras especificaciones en lugar de fase lineal (ej: filtros de fase mínima).

El filtro pasabajo ideal:

$$H_I(e^{j\omega}) = \begin{cases} 1 & 0 \leq |\omega| \leq \omega_c \\ 0 & \omega_c \leq |\omega| \leq \pi \end{cases}$$

$$h_I(n) = \frac{\sin \omega_c n}{\pi n}, \quad -\infty < n < \infty$$

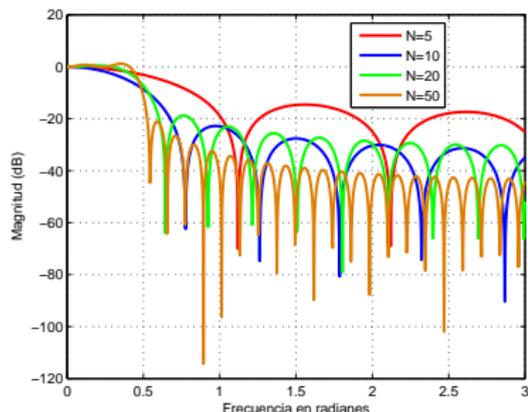
Este filtro es IIR y no causal!!

Este filtro se podría aproximar como:

$$h(n) = \frac{\sin \omega_c (n - N)}{\pi n}, \quad 0 \leq n \leq 2N$$

$h(n)$ es de fase lineal y es equivalente a multiplicar $\frac{\sin \omega_c (n - N)}{\pi n}$ por una ventana rectangular de tamaño $2N$

Métodos de ventaneo II



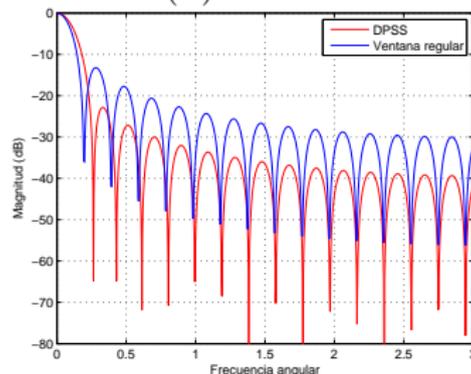
El compromiso entre ancho del lóbulo principal, altura de lóbulos secundarios y orden del filtro no es muy bueno!!

La idea es buscar ventanas que mejoren estos compromisos!!

Métodos de ventaneo III

Es de interés conocer como es la “mejor” ventana $v(n)$ de tamaño N y frecuencia de corte σ :

$$\min_{v(n)} \frac{\frac{1}{2\pi} \int_{|\omega| \geq \sigma} |V(e^{j\omega})|^2 d\omega}{\frac{1}{2\pi} \int_{-\pi}^{\pi} |V(e^{j\omega})|^2 d\omega}$$

La solución a este problema es lo que se conoce como *Discrete prolate spheroidal sequence* (DPSS). Esto es muy complejo de calcular y por eso se buscan aproximaciones a esta ventana óptima.

Una muy buena aproximación a esta ventana lo constituye la ventana de Kaiser:

$$v(n) = \begin{cases} \frac{I_0[\beta(1-[2n/N-1]^2)^{1/2}]}{I_0(\beta)} & 0 \leq n \leq N \\ 0 & \text{en otro caso} \end{cases}, \quad I_0(x) = 1 + \sum_{k=1}^{\infty} \left[\frac{(0,5x)^k}{k!} \right]^2$$

La elección de los parámetros en función de los requerimientos de un filtro pasabajo ($\delta_1 = \delta_2 = \delta$, $A_s = -20 \log_{10} \delta$, $\Delta\omega = \omega_s - \omega_p$):

$$\beta = \begin{cases} 0,1102(A_s - 8,7) & A_s > 50 \\ 0,5842(A_s - 21)^{0,4} + 0,078(A_s - 21) & 21 \leq A_s \leq 50 \\ 0 & A_s < 21 \end{cases}, \quad N = \frac{A_s - 7,95}{2,285\Delta\omega}$$

Esta ventana es muy versátil y es la preferida para la mayor parte de los diseños usando ventaneo!

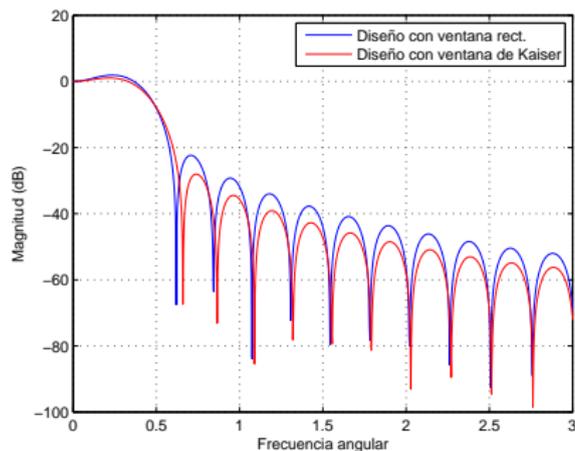
Métodos de ventaneo V

Se tienen las siguientes especificaciones:

$$0,95 \leq |H(e^{j\omega})| \leq 1,05 \quad 0 \leq \omega \leq 0,125\pi$$

$$|H(e^{j\omega})| \leq 0,05 \quad 0,175\pi \leq \omega \leq \pi$$

En ambos casos el número de coeficientes es de 26.



Filtros equirriple I

Si se dispone de especificaciones:

$$A_i^d(e^{j\omega}) - \delta_i \leq |H(e^{j\omega})| \leq A_i^d(e^{j\omega}) + \delta_i, \quad \omega_{i-1} \leq \omega \leq \omega_i, \quad i = 1, \dots, B.$$

Sea $C_i = [\omega_{i-1}, \omega_i]$, $C = \bigcup_{i=1}^B C_i$. La idea de los filtros equirriple encontrar el filtro $H(e^{j\omega})$ tal que:

$$\min_{h(n): 0 \leq n \leq N} \max_{\omega \in C} |E(e^{j\omega})|, \quad E(e^{j\omega}) = W(e^{j\omega}) \left[H(e^{j\omega}) - A^d(e^{j\omega}) \right]$$

- Este es un problema de aproximación por polinomios trigonométricos de Chebyshev.
- El algoritmo para resolverlo es el algoritmo de Remez.
- La implementación de dicho algoritmo para diseño de filtros digitales más popular es el algoritmo de Parks-McClellan.
- Para unas dadas especificaciones los filtros equirriple dan el menor orden posible!

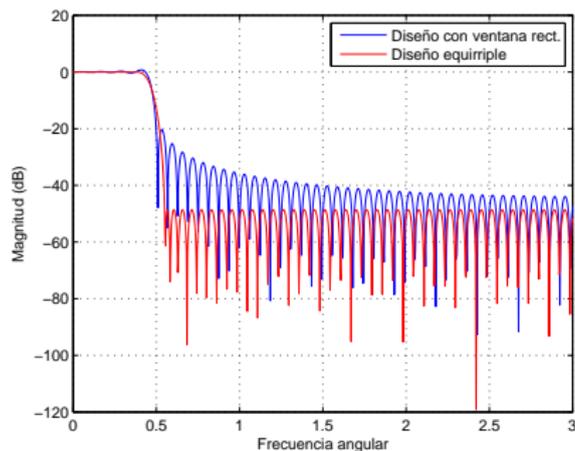
Filtros equiripple II

Se tienen las siguientes especificaciones:

$$0,95 \leq |H(e^{j\omega})| \leq 1,05 \quad 0 \leq \omega \leq 0,125\pi$$

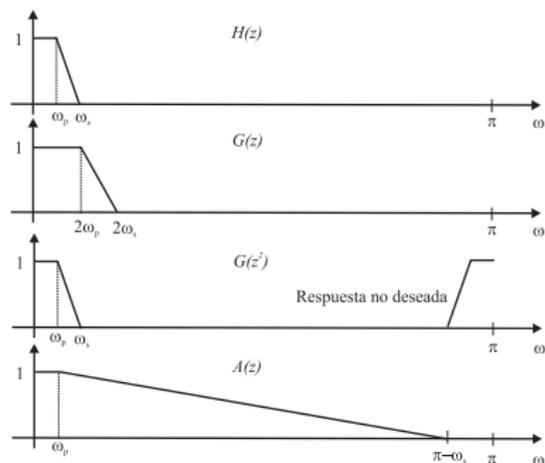
$$|H(e^{j\omega})| \leq 0,003 \quad 0,175\pi \leq \omega \leq \pi$$

En ambos casos el número de coeficientes es de 100.



Diseños multistage I

Este método está orientado para el diseño de pasabajos muy estrictos.



La idea es diseñar un filtro $G(z)$ con especificaciones tan severas. Al hacer $G(z^2)$ generamos la respuesta que necesitamos. Pero también aparece una copia no deseada. Dicha copia se puede eliminar con un filtro no muy estricto.

De esta forma podemos generar un filtro $G(z^2)A(z)$ cuya longitud puede ser menor que la del filtro original $H(z)$!!

Diseños multistage II

Se sabe que para el filtro $H(z)$ se tiene

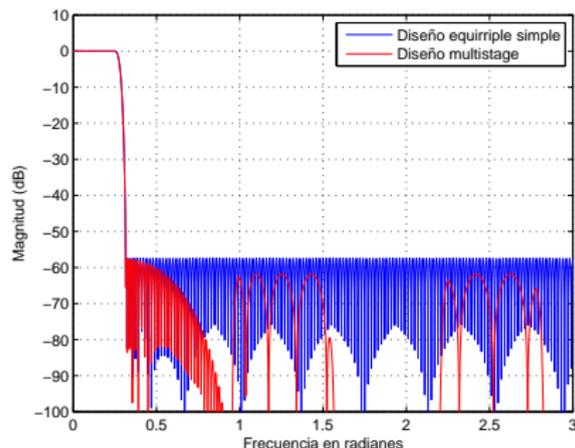
$$N_H \approx \frac{D(\delta_1, \delta_2)}{\Delta\omega}, \quad \Delta\omega = \omega_p - \omega_s$$

Claramente $G(z)$ tiene $N_G = N_H/2$ coeficientes y $A(z)$ tiene N_A coeficientes que satisfacen $N_A \ll N_H$.

Esto quiere decir que la cascada $G(z^2)A(z)$ tiene aproximadamente $N_H/2$ coeficientes!!. Podemos reducir la cantidad de multiplicaciones necesarias! (El número de elementos de memoria necesarios permanece igual).

Esto se puede generalizar al caso en que la expansión se hace con un factor M genérico!! Sin embargo si M es muy grande el filtro $A(z)$ puede tener muchos coeficientes y puede no haber ganancia!!

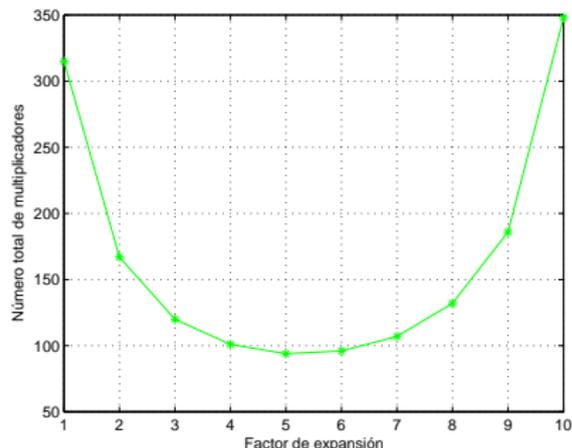
Diseños multistage III



Ejemplo con

$$0,98 \leq |H(e^{j\omega})| \leq 1,02 \quad 0 \leq \omega \leq 0,08\pi$$

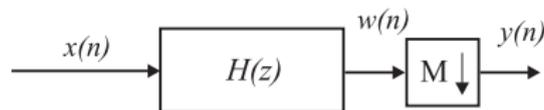
$$|H(e^{j\omega})| \leq 0,001 \quad 0,1\pi \leq \omega \leq \pi$$



Con $M = 4$ obtenemos el mínimo número de coeficientes (94) con una reducción de más del 60% de los coeficientes con respecto al caso simple (315 coeficientes.)!

Tener en cuenta que el sistema debe implementarse en forma práctica como la cascada de los sistemas para aprovechar la ganancia en implementación!

Muchas veces es necesario realizar la siguiente operación:



$$y(n) = w(nM),$$

$$Y(e^{j\omega}) = \frac{1}{M} \sum_{k=0}^{M-1} W\left(e^{j\frac{\omega-2\pi k}{M}}\right)$$

Se ve entonces que la función $H(z)$ es preparar la señal antes de ser decimada para minimizar el efecto del aliasing!

Esta operación es básica en todo sistema que necesite disminuir la tasa de información que transmite una determinada señal!

Muchas veces $H(z)$ tiene una banda de transición extremadamente estrecha lo que implica muchos coeficientes (multiplicaciones). Sin embargo:

El decimador se queda con una de cada M muestras! Se puede realizar el filtrado por $H(z)$ en una de cada M unidades de tiempo.

En una unidad de tiempo por cada M nuestro procesador va a tener un pico de actividad y en el resto no tendrá ninguna! Esto puede resultar en un pobre aprovechamiento de los recursos!

Es posible compensar esto y redistribuir las multiplicaciones que $H(z)$ debe realizar por cada muestra útil durante las M unidades de tiempo.

Para realizar esto vamos a recurrir a la descomposición polifásica de orden M para cualquier $H(z)$:

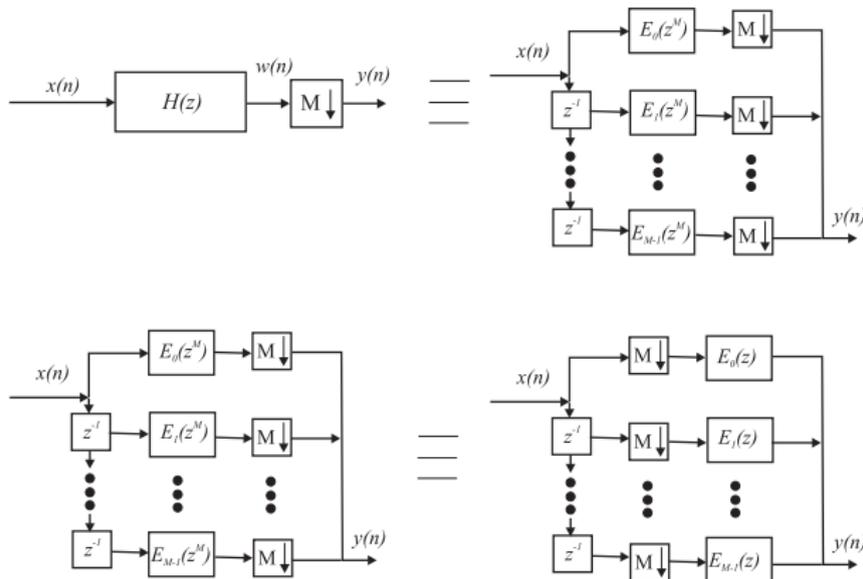
$$\begin{aligned}
 H(z) &= \sum_{n=-\infty}^{\infty} h(n)z^{-n} \\
 &= \sum_{n=-\infty}^{\infty} h(nM)z^{-nM} + z^{-1} \sum_{n=-\infty}^{\infty} h(nM+1)z^{-nM} + \dots \\
 &\quad + z^{-M+1} \sum_{n=-\infty}^{\infty} h(nM+M-1)z^{-nM} \\
 &= E_0(z^M) + z^{-1}E_1(z^M) + \dots + z^{-M+1}E_{M-1}(z^M)
 \end{aligned}$$

$$E_i(z) = \sum_{n=-\infty}^{\infty} h(nM+i)z^{-i},$$

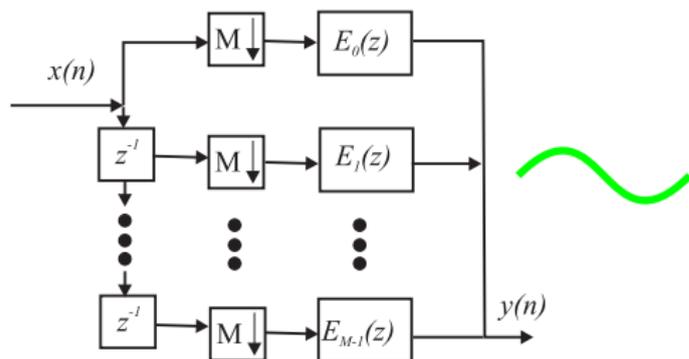

$$i = 0, \dots, M-1$$



Decimadores IV



La estructura queda como

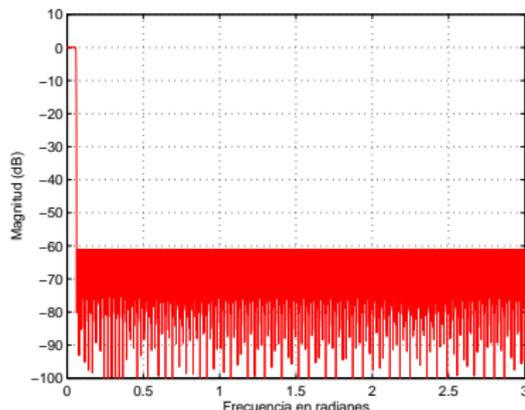


Cada uno de los $E_i(z)$ tiene aproximadamente N/M coeficientes donde N es el número de coeficientes de $H(z)$. Como cada $E_i(z)$ opera a una tasa M veces menor entonces el número de multiplicaciones por unidad de tiempo se reduce a $N/M!!$

Ahora el procesador realiza N/M multiplicaciones por unidad de tiempo en forma continua y la carga computacional se encuentra mejor balanceada!!

Sea un el diseño de un decimador con factor $M = 50$. Las características del filtro $H(z)$ cuya frecuencia de corte debe ser $\pi/50$ son:

$$0,98 \leq |H(e^{j\omega})| \leq 1,02 \quad 0 \leq \omega \leq 0,08\pi, \quad |H(e^{j\omega})| \leq 0,001 \quad 0,1\pi \leq \omega \leq \pi$$



El filtro $H(z)$ tiene 3335 coeficientes. Dado que $M = 50$ la longitud de los filtros $E_i(z)$ es como máximo de 67 coeficientes.

Supongamos que la muestras de $x(n)$ surgen de muestrear una señal a 8KHz

- La implementación del decimador directa requeriría garantizar ≈ 27 millones de multiplicaciones por segundo.
- La implementación polifásica sólo requiere garantizar $\approx 0,5$ millones de multiplicaciones por segundo.

- Documentación del *Filter Design Toolbox* de Matlab®.
- A. Oppenheim, R. Schafer and J. Buck, *Discrete-time signal processing*. Upper Saddle River, N.J.: Prentice Hall 1999.
- P. Vaidyanathan, *Multirate Systems And Filter Banks*. Upper Saddle River, N.J.: Prentice Hall 1992.
- B. Porat, *A Course in Digital Signal Processing*. New York: John Wiley 1997.
- E. Feachor and B. Jervis, *Digital Signal Processing: A practical approach*. Upper Saddle River, N.J.: Prentice Hall 2002.
- U. Meyer-Baese, *Digital Signal Processing with FPGA*. Berlin, Springer-Verlag, 2001.
- K. Rangarao and R. Mallik, *Digital Signal Processing: A Practitioner Approach*. New York: John Wiley 2005.
- Papers varios de *IEEE Transactions on Signal Processing*.