

Ampliación de Señales y Sistemas

Curso 2013/14

Tema 6:

Introducción al diseño de filtros digitales

Esquema del tema

- 6.1 Fundamentos del filtrado digital
- 6.2 **Diseño de filtros FIR**
- 6.3 **Panorámica del diseño de filtros IIR**
- 6.4 **Diseño de filtros digitales con Matlab**

6.1 FUNDAMENTOS DEL FILTRADO DIGITAL

En procesamiento de señal, la función de un filtro es la de eliminar partes no deseadas de la señal (como ruido) o extraer partes útiles, como las componentes en una determinada banda de frecuencias.



Aplicaciones:

Comunicaciones

Procesado de voz

Radar/Teledetección

Minería de datos

Sismología

Tratamiento de imagen y vídeo

Bio-Ingeniería

Reconocimiento de patrones

Filtros digitales vs filtros analógicos

Existen dos categorías de filtros, analógicos y digitales, que son bastante diferentes tanto en su fabricación física como en sus principios de funcionamiento.

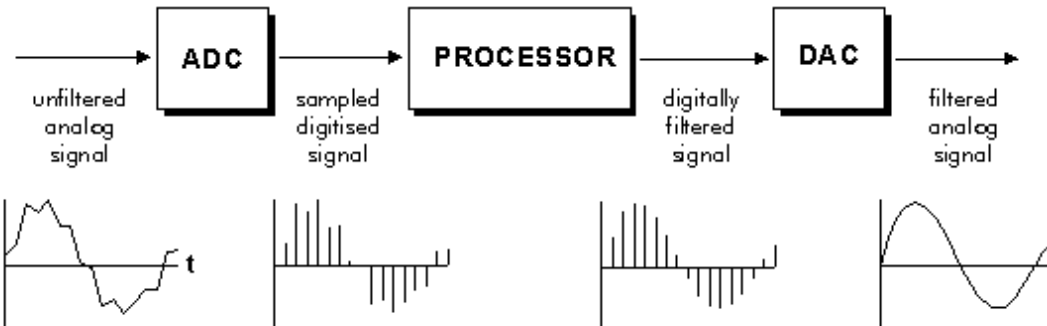
Filtros analógicos:

- Utilizan componentes electrónicos analógicos (R, C, L, Amp)

Filtros digitales:

- Utilizan un sistema digital que realiza cálculos numéricos (micro, DSP, FPGA, GPU...)

Los filtros digitales pueden utilizarse también para procesar señales analógicas (muestreando previamente)



Principales ventajas de los filtros digitales

1. Un filtro digital es **programable**, por lo que puede ser modificado con facilidad.
2. Los filtros digitales **se diseñan, prueban e implementan de manera sencilla**.
3. Los filtros digitales son muy **versátiles** (filtros adaptativos).
4. El comportamiento de los filtros digitales **no cambia en el tiempo** por derivas o deterioro de sus componentes.
5. Las necesidades **hardware** son relativamente **simples** y **homogéneas**.
6. Los filtros digitales **trabajan con señales de baja frecuencia** de forma **precisa**.
7. Con el aumento de la **velocidad de procesamiento**, hoy en día los filtros digitales pueden utilizarse también para **señales de alta frecuencia**.

Diseño de filtros digitales

Los filtros digitales son SLTI → podemos utilizar todas las herramientas estudiadas en esta asignatura para **analizar** el comportamiento de los filtros. Sin embargo, como ingenieros, también estamos interesados en **diseñar** filtros.

Procedimiento genérico para el diseño de filtros:

Paso 1. Escoger la respuesta deseada, basándose en los requisitos de la aplicación para la que se diseña el filtro.

Paso 2. Escoger un tipo de filtro.

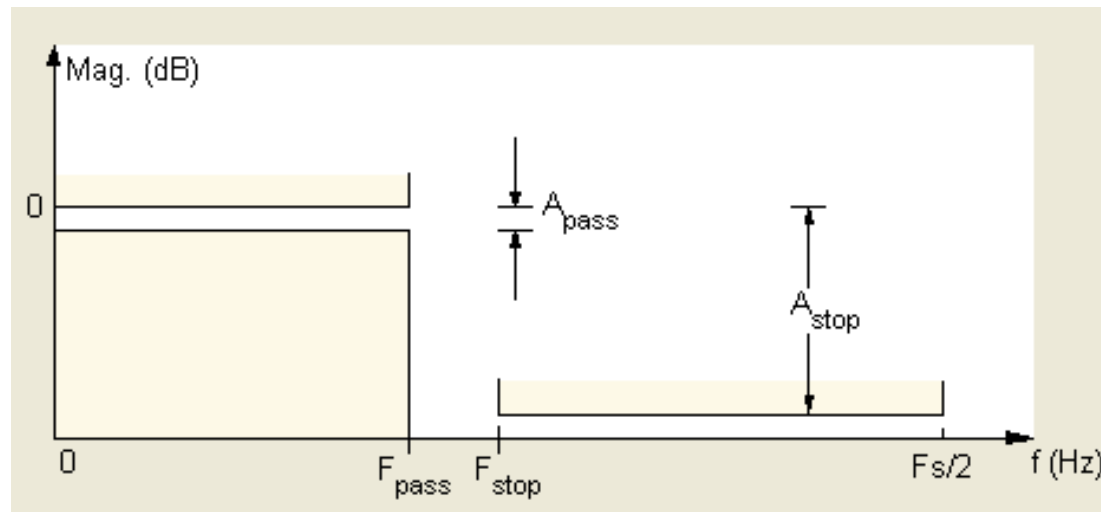
Paso 3. Escoger una figura de mérito (medida de calidad).

Paso 4. Determinar el filtro que optimice el criterio en 3.

Paso 1: Respuesta deseada

Las respuestas en frecuencia ideales no son posibles den la práctica (no causales, retardo infinito, elevado coste...)

En la práctica



Frecuencia: Pass-Band, Transition-Band, Stop-Band

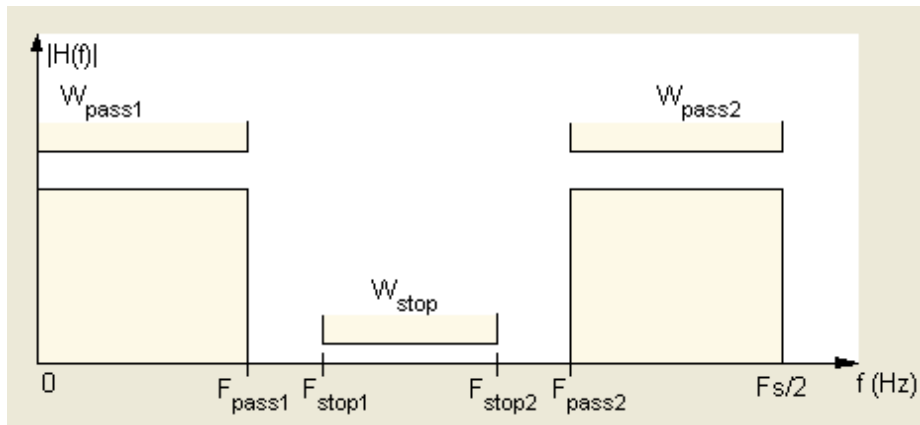
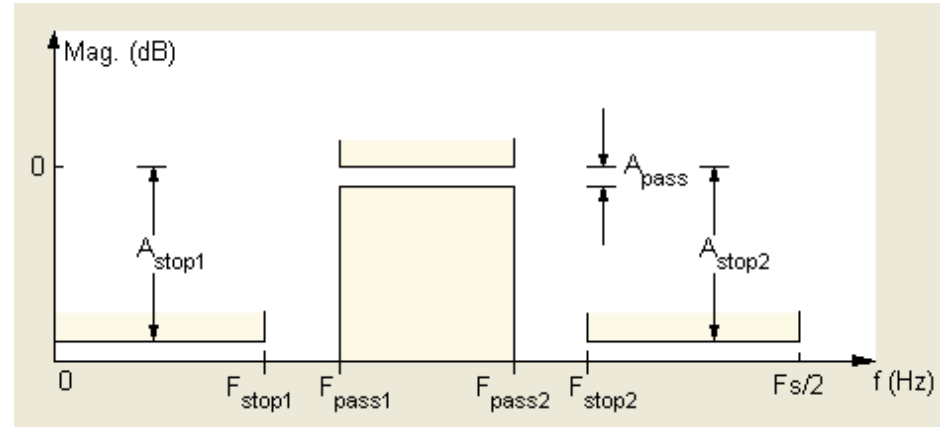
F_s (frecuencia de muestreo)

Magnitud: Ampl. Deviation/Tolerance (Pass), Ampl. Margin (Stop)

Desired response

Los principales tipos de filtros son: paso-bajo, paso-alto, paso-banda y banda-rechazada

Paso-banda



Banda-rechazada

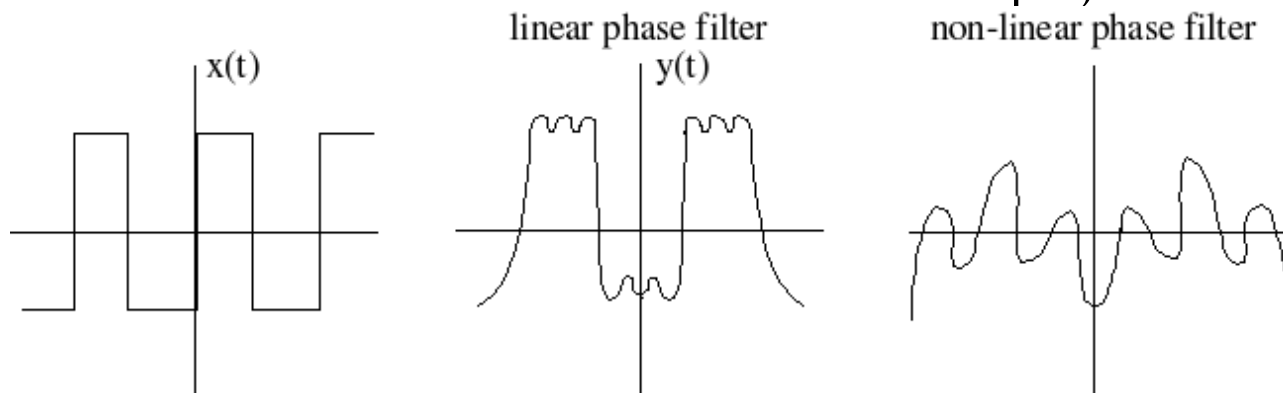
Fase lineal

Y ¿qué pasa con la fase? ¿Es una parte importante de la respuesta del filtro?

En general, para $-\pi \leq \omega \leq \pi$ $H(\omega) = |H(\omega)|e^{-i\theta(\omega)}$

De forma rigurosa, $H(\omega)$ tiene fase lineal si $H(\omega) = |H(\omega)|e^{-i\omega K}e^{-i\theta_0}$

¡Una respuesta en fase lineal genera el mismo retardo temporal para todas las frecuencias! (Recuérdese la propiedad de desplazamiento en la TF.) Esta es una característica deseable cuando la forma temporal de la señal es importante (p.e. para su visualización/sincronismo en un osciloscopio).



Fase lineal

¿Qué debe cumplir $h(n)$ para que tenga una fase lineal?

M = longitud de $h(n)$

Para $\phi_0 = 0$,

$$\begin{aligned} h(0) + h(M-1) &= \text{real number} \\ h(0) - h(M-1) &= \text{pure imaginary number} \\ h(1) + h(M-2) &= \text{pure real number} \\ h(1) - h(M-2) &= \text{pure imaginary number} \\ &\vdots \end{aligned}$$

$$h(k) = h^*(M-1-k)$$

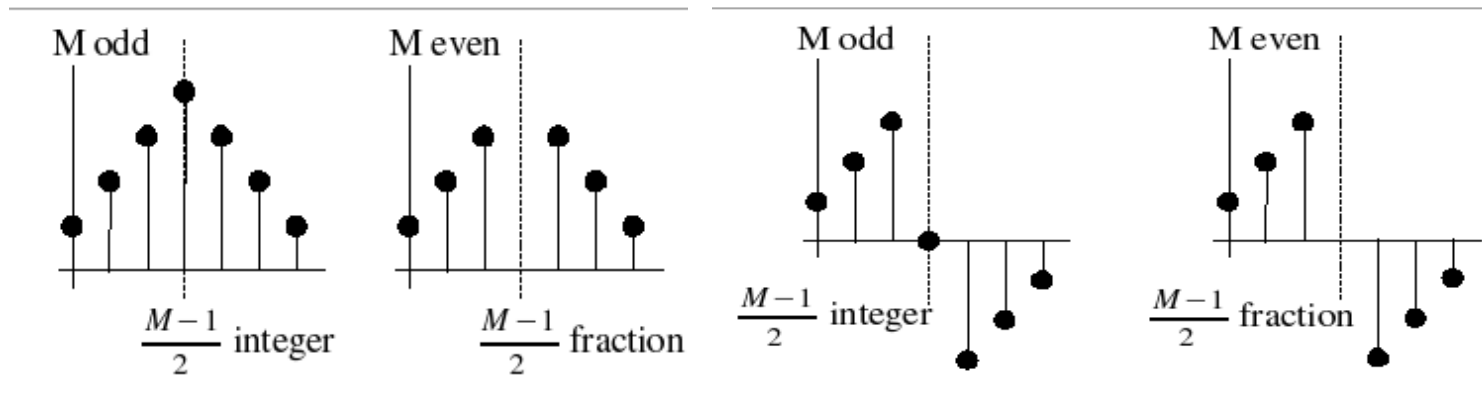
Para $\phi_0 = \frac{\pi}{2}$,

$$\begin{aligned} h(0) + h(M-1) &= \text{pure imaginary} \\ h(0) - h(M-1) &= \text{pure real number} \\ &\vdots \end{aligned}$$

$$h(k) = -(h^*(M-1-k))$$

¡Necesitamos simetría!

Ejemplos para coeficientes reales



$\phi_0 = 0$ (Symmetric Filters). $h(k) = h(M-1-k)$.

$\phi_0 = \frac{\pi}{2}$ (Anti-Symmetric Filters). $h(k) = -(h(M-1-k))$.

Paso 2: Elegir un tipo de filtro

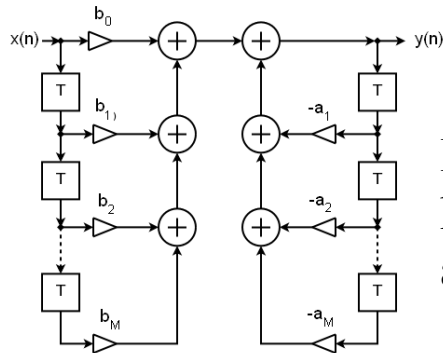
Hay dos tipos básicos de filtros digitales:

IIR: Infinite Impulse Response ($h[n]$ tiene longitud infinita)

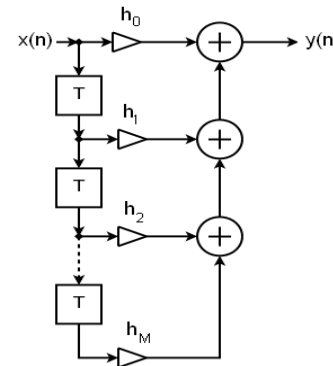
FIR: Infinite Impulse Response ($h[n]$ tiene longitud finita)

Estabilidad

Siempre estables



Es difícil obtener $h[n]$ de forma analítica



Obtener $h[n]$ es trivial \rightarrow la convolución es una opción para calcular $y[n]$

Otro parámetro muy importante en un filtro es el **orden del filtro**:

IIR: número máximo de retardos

FIR: número máximo de retardos (i.e., longitud de la respuesta al impulso)

FIR vs IIR

Las herramientas para diseñar filtros FIR y filtros IIR son muy diferentes, como también lo es su comportamiento.

Ventajas de los filtros FIR

1. Conceptualmente sencillos y fácilmente implementables
2. Pueden implementarse vía convolución
3. Siempre son estables
4. Poco sensibles a la cuantización de la amplitud
5. Pueden tener fase lineal (mismo retardo temporal para todas las frec.)

Ventajas de los filtros IIR

1. Más adecuados para la aproximación/adaptación de diseños analógicos
2. Para una respuesta en frecuencia dada, los filtros IIR necesitan típicamente muchos menos cálculos teóricos que su equivalente FIR, especialmente para bandas de transición estrechas (filtros abruptos).

problema estabilidad

Paso 3: Figuras de mérito

El diseño de filtros puede verse como un problema de aproximación (ajuste) de funciones

Dadas ciertas restricciones (especificaciones), para encontrar el diseño óptimo necesitamos saber cuál es la medida de calidad/coste que queremos maximizar/minimizar

Criterio de optimización L2: minimiza la diferencia de energía entre la respuesta deseada y la respuesta real

$$\min_{h[n]} \left\{ \int_{-\pi}^{\pi} (|H_d(\omega) - H(\omega)|)^2 d\omega \right\}$$

Criterio de optimización L ∞ : dada una función de ponderación del error $W(\omega)$, minimiza el error máximo $E(\omega)$

$$\operatorname{argmin}_h \operatorname{argmax}_{\omega \in F} |E(\omega)| = \operatorname{argmin}_h \|E(\omega)\|_{\infty}$$

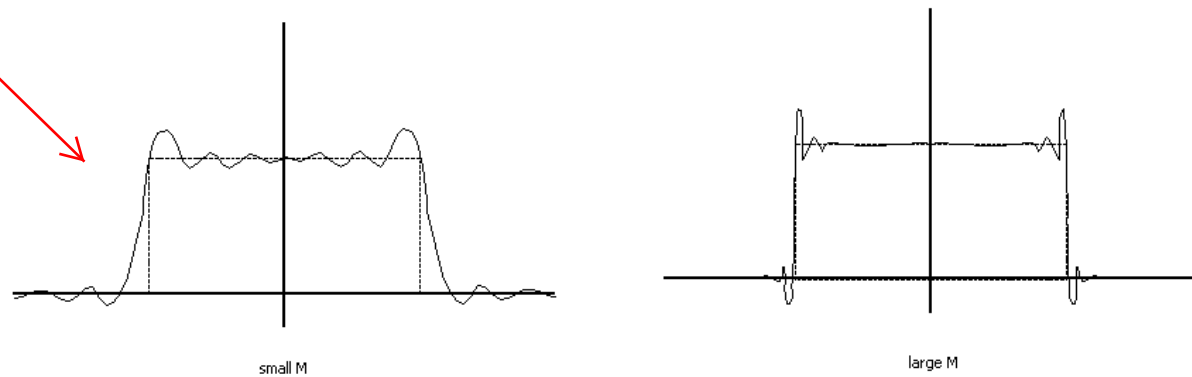
$$E(\omega) = W(\omega)(H_d(\omega) - H(\omega))$$

y F es un subconjunto compacto de $\omega \in [0, \pi)$, (i.e., toda ω en las frecuencias de paso y rechazo).

Figuras de mérito

¿Por qué L2 (error cuadrático medio) no es suficiente? → Depende de la aplicación

Generalmente si mimizamos el error cuadrático medio, se observa el fenómeno de Gibbs



Para muchas aplicaciones, estos picos no son tolerables (violaríamos la máscara de transmisión) → La norma L^∞ garantiza el cumplimiento de la máscara

Paso 4: Encontrar el filtro

El método específico para encontrar la solución de nuestro diseño está fuertemente relacionado con el filtro elegido:

Diseños analíticos → válido para filtros IIR, orden pequeño,

Diseño numérico sistemático → Filtros FIR (Parks-McClellan)

Diseño numérico asistido por ordenador → Toolboxes (Matlab) o códigos propios

Por otro lado, para un tipo específico de filtro hay más de un método que puede ser utilizado

6.2 DISEÑO DE FILTROS FIR

Existen distintos métodos para el diseño de filtros FIR, entre ellos:

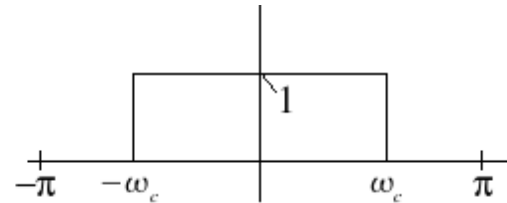
Enventanado → Minimiza la norma L2

Diseño basado en el muestreo de la respuesta en frecuencia → Muy baja complejidad

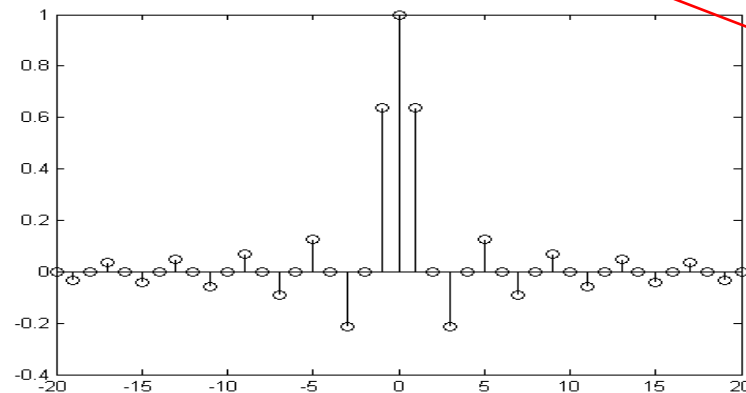
Pese a que en situaciones específicas, para alguno de los casos anteriores pueda encontrarse una expresión cerrada para $h[n]$, en general el diseño del filtro (dada una máscara de frecuencia encontrar la respuesta al impulso óptima) no puede resolverse de forma analítica.

Enventanado

Consideremos el caso en el que queremos diseñar un filtro paso-bajo ideal:



¿Qué ocurre si calculamos $h(n)$ como $TF^{-1}\{H(\omega)\}$? $\rightarrow h(n) = \sin(\omega_c n)/(\pi n) \rightarrow h(n)$ es infinito y no causal

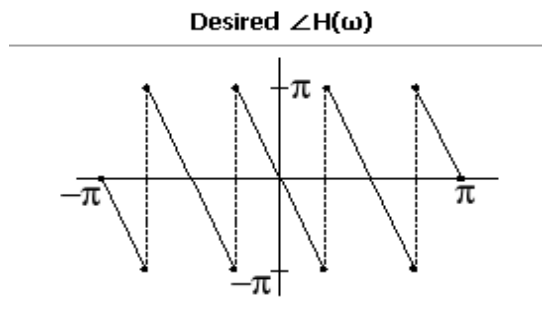
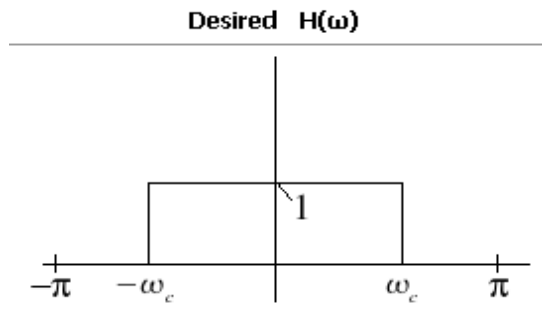


NOTA: AQUI SERIA UNA h(t) pero se considera que esta muestreada....y tiene h[n]

Pero podríamos desplazar $h(n)$ hacia la derecha (simplemente introduciría una fase lineal) y **truncar la señal para quedarnos con un número finito de valores** (introduciría una pequeña distorsión en el módulo y la fase)

Enventanado

Versión desplazada



The slope of each line is $-\left(\frac{M-1}{2}\right)$.

$$\begin{cases} e^{-\left(i\omega\frac{M-1}{2}\right)} & \text{if } -\omega_c \leq \omega \leq \omega_c \\ 0 & \text{if } -\pi \leq \omega < -\omega_c \vee \omega_c < \omega \leq \pi \end{cases}$$

Truncando la versión desplazada tenemos

$$h[n] = \begin{cases} h_d[n] & \text{if } 0 \leq n \leq M-1 \\ 0 & \text{if else} \end{cases}$$

Alternativamente, podemos definir

$$w[n] = \begin{cases} 1 & \text{if } 0 \leq n < M \\ 0 & \text{if else} \end{cases}$$

Ventana rectangular

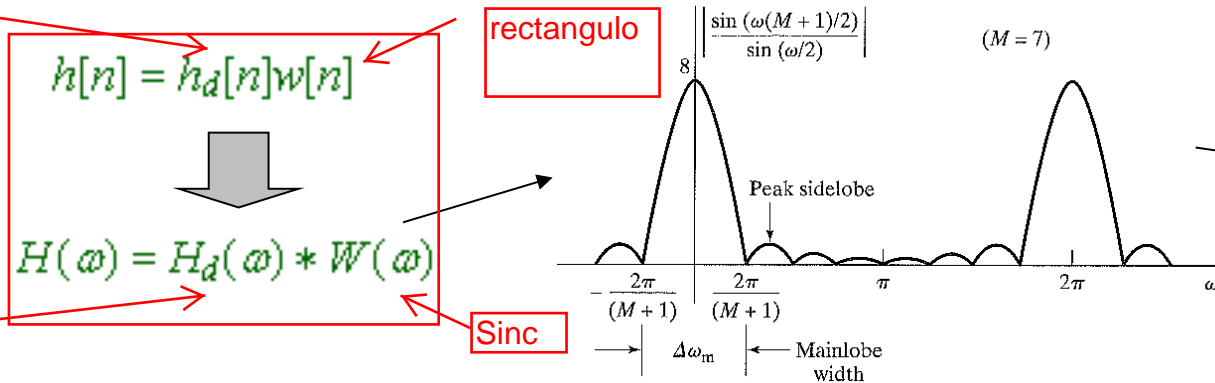
donde $w(n)$ recibe el nombre de ventana (más específicamente, ventana rectangular) y la $h(n)$ real puede obtenerse como la multiplicación de la $h(n)$ deseada y la ventana $w(n)$

$$h[n] = h_d[n]w[n]$$

Enventanado

¿Cómo de bueno es este diseño? → Analicémoslo → Si multiplicamos en tiempo estamos convolucionando en el dominio de la frecuencia

Sinc

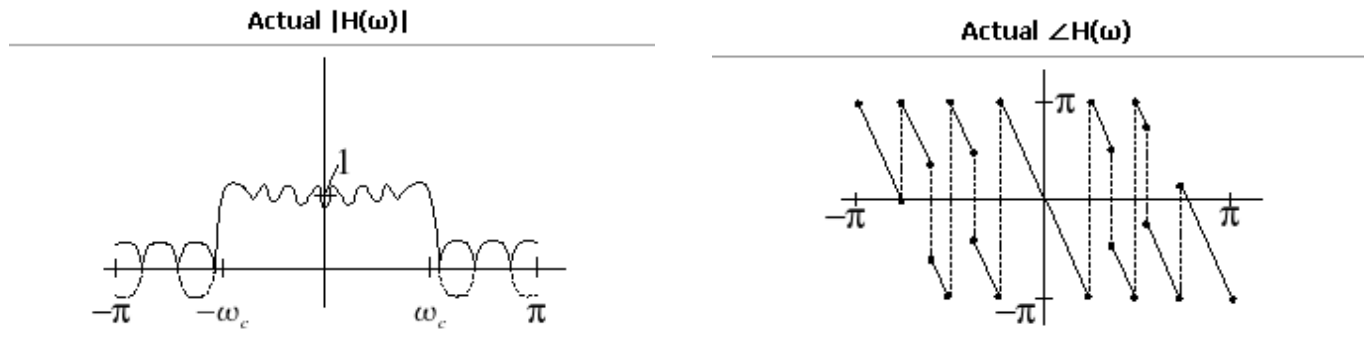


Mejor cuanto más parecido a una delta

rectángulo

Sinc

La forma de la sinc (y especialmente los lóbulos secundarios) provoca que aparezcan ligeras ondulaciones (rizado) en la amplitud del filtro, así como cambios bruscos de fase (debido al cambio de signo en la sinc).



Tipos de ventanas

antes habiamos
considerado un
rectangulo....

Barlett (triangular)

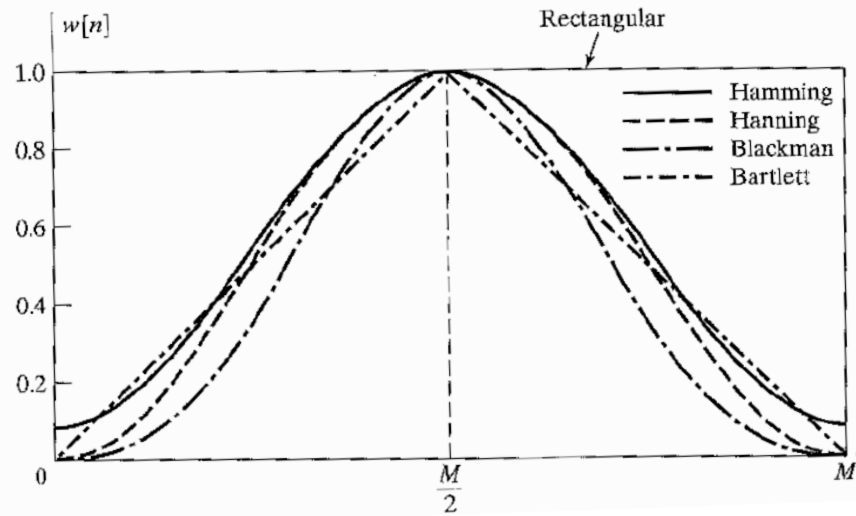
$$w[n] = \begin{cases} 2n/M, & 0 \leq n \leq M/2, \\ 2 - 2n/M, & M/2 < n \leq M, \\ 0, & \text{otherwise} \end{cases}$$

Hanning

$$w[n] = \begin{cases} 0.5 - 0.5 \cos(2\pi n/M), & 0 \leq n \leq M, \\ 0, & \text{otherwise} \end{cases}$$

Hamming

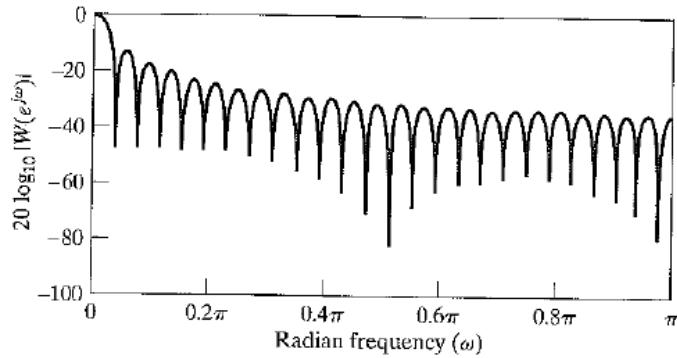
$$w[n] = \begin{cases} 0.54 - 0.46 \cos(2\pi n/M), & 0 \leq n \leq M, \\ 0, & \text{otherwise} \end{cases}$$



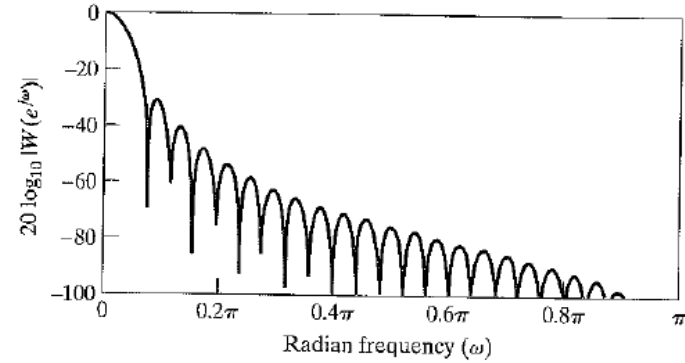
Tipos de ventanas

son modulos y en escalas logaritmica

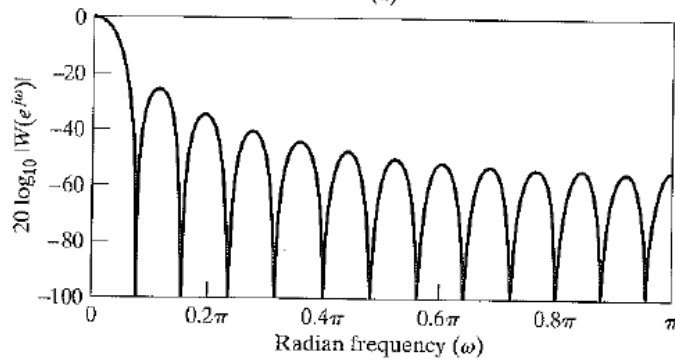
Respuesta en frecuencia...



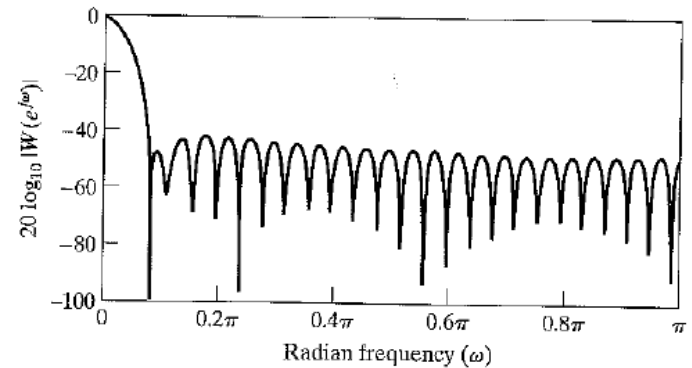
(a)



(c)



(b)



(d)

(a) Rectangular (b) Triangular (c) Hanning (d) Hamming

Enventanado

Comparación de las ventanas:

Type of Window	Peak Side-Lobe Amplitude (Relative)	Approximate Width of Main Lobe
Rectangular	-13	$4\pi/(M + 1)$
Bartlett	-25	$8\pi/M$
Hanning	-31	$8\pi/M$
Hamming	-41	$8\pi/M$

La ventana rectangular es la más estrecha (más cercana a una delta), pero también la que tiene unos lóbulos secundarios más altos

Nótese que todas las ventanas son simétricas → Fase lineal

Desde un punto de vista de optimización, el Teorema de Parseval puede utilizarse para demostrar que un filtro con ventana rectangular minimiza la norma L2

Muestreo de la respuesta en frecuencia

Dada una respuesta en frecuencia deseada, el método del muestreo de la respuesta en frecuencia, diseña un filtro cuya respuesta en frecuencia es exactamente igual a la del filtro deseado en un conjunto finito de frecuencias ω_k

creamos un sistema NxN !!!
Esto lo sabeis hacer !!!

$$\forall k, k = [0, 1, \dots, N-1] : H_d(\omega_k) = \sum_{n=0}^{M-1} (h(n)e^{-i\omega_k n})$$

Nótese que si se quiere que el filtro tenga fase lineal, esta característica deberá incluirse en los valores de la respuesta en frecuencia suministrados

La expresión anterior define un sistema lineal de ecuaciones → puede resolverse de manera sencilla

Además, si ω_k son tales que $\forall k, 0 \leq k \leq M-1 : \omega_k = \frac{n\pi k}{M}$, entonces

$$h[n] = \text{IDFT}[H_d(\omega_k)]$$

¡Los algoritmos FFT pueden utilizarse para diseñar nuestro filtro!

Comentarios al diseño basado en el muestreo de la respuesta en frecuencia

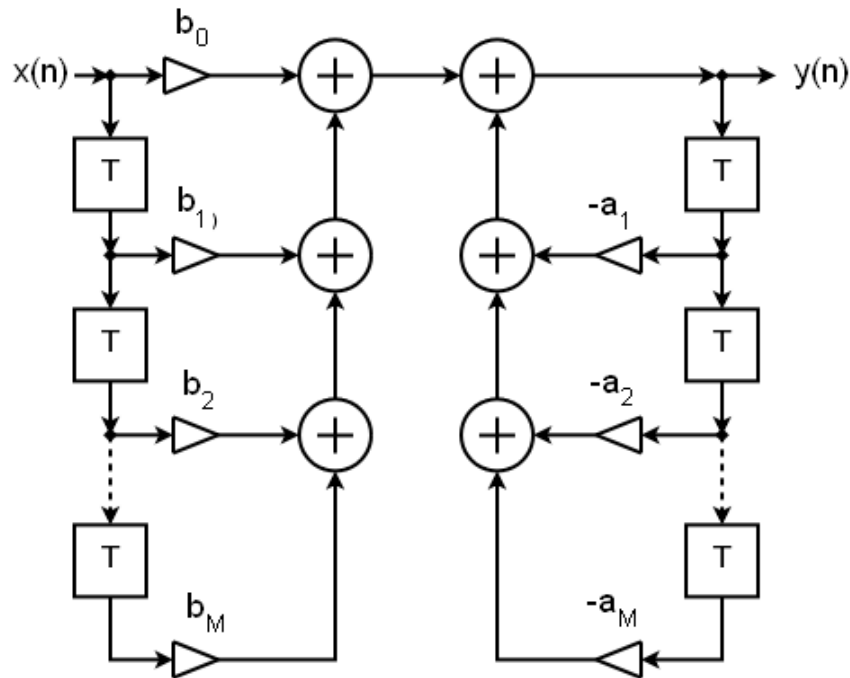
Este método es sencillo conceptualmente y, para el caso de valores equiespaciados, muy eficiente desde un punto de vista numérico, puesto que $h(n)$ puede calcularse utilizando la IDFT

La respuesta diseñada $H(\omega)$ coincide exactamente con los valores especificados en los puntos de muestreo $\omega \neq \omega_k$, pero puede ser muy diferente en el resto de frecuencias → Este es el mayor problema de este método de diseño

Posible solución para este problema: especificar más valores de muestreados que grados de libertad (orden del filtro) y validar el diseño del filtro mediante la minimización del error en las muestras que no han sido inicialmente consideradas. (“Validación heurística”)

O USAR
MIMIMOS
CUADRADOS!!!

6.3 PANORÁMICA DEL DISEÑO DE FILTROS IIR



$$y(n) = - \left(\sum_{k=1}^{M-1} (a_k y(n-k)) \right) + \sum_{k=0}^{M-1} (b_k x(n-k))$$

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_M z^{-M}}$$

El diseño de filtros IIR se reduce a: elegir los $\{a_i\}$ y $\{b_i\}$ que mejor aproximan la respuesta $|H_d(\omega)|$ deseada (en alguna ocasión incluso $H_d(\omega)$)

Crterios

Existen distintas técnicas en función de las figuras de mérito que quieran utilizarse:

Transformación Bilineal: transforma diseños analógicos que son óptimos en L^∞ (y otros) en diseños IIR digitales.

Método de Prony: método ad-hoc de ajuste de la respuesta al impulso en el dominio en el tiempo que es cuasi óptimo en L_2 .

Diseño L_p (norma p) óptimo: diseño óptimo de filtros (minimiza la norma p , con $1 < p < \infty$) utilizando técnicas de optimización no-lineal.

Comentarios:

- Las técnicas Prony de predicción lineal suelen utilizarse para: (a) tener un prototipo básico del filtro y (b) ajuste de datos e identificación de sistemas.
- Las técnicas de optimización L_p se utilizan para el diseño de filtros para los cuales no existen soluciones analógicas.
- En esta sección, únicamente discutiremos con al de detalle el método de transformación bilineal.

Transformación bilineal y filtros analógicos

El diseño de filtros analógicos se basa en la transformada de Laplace:

$$H(s) = \int_{-\infty}^{\infty} h_a(t) e^{-(st)} dt$$

Desde principios del s. XX, ha habido un gran esfuerzo investigador en el diseño de filtros analógicos de la forma

$$H(s) = \frac{b_0 + b_1s + b_2s^2 + \dots + b_Ms^M}{1 + a_1s + a_2s^2 + \dots + a_Ms^M}$$

Transformación bilineal: transforma un filtro analógico $H_a(s)$ en un filtro digital $H(z)$

Procedimiento para el diseño:

1. Diseño del filtro analógico
2. Transformación del filtro analógico en el filtro digital

Diseñando un filtro analógico

Puesto que **un filtro IIR causal no puede tener fase lineal** (no existe ningún punto posible de simetría) → La investigación en el diseño de filtros analógicos se ha concentrado en obtener filtros que acoten el rizado (L^∞) de la respuesta en frecuencia del filtro.

Tipos tradicionales de filtros IIR analógicos L^∞ :

Butterworth

Chebyshev

Elliptic

Otros tipos más específicos son: filtro de Bessel, filtro Gaussiano, filtro óptimo de Legendre, filtro Linkwitz-Riley...

Filtros analógicos tradicionales

Los filtros de **Butterworth** se diseñan de manera que la respuesta en frecuencia en la banda de paso sea **tan plana como matemáticamente sea posible**, y decaiga hasta cero en la banda de rechazo. Estos filtros también reciben el nombre de ‘filtros de respuesta máximamente plana’.

El filtro de tipo Butterworth fue propuesto por el ingeniero británico Stephen Butterworth.

Los filtros de **Chebyshev** tienen la propiedad de **minimizar el error** entre la respuesta ideal del filtro y la respuesta real sobre el rango del filtro, pero **acotando el rizado en la banda de paso o en la de rechazo**.

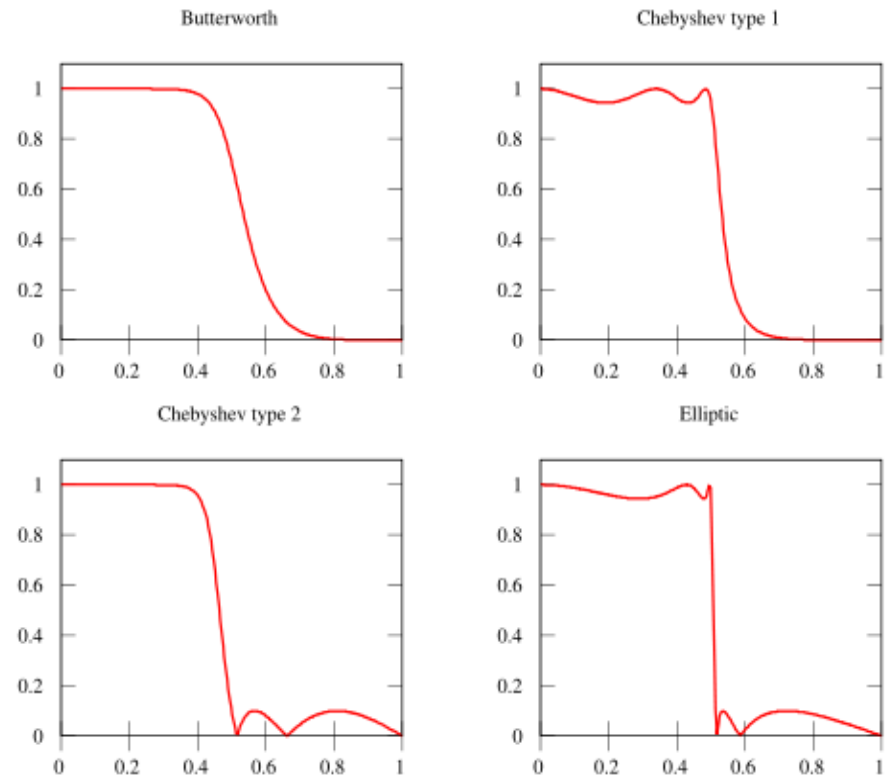
El nombre de este filtro se debe al matemático Pafnuty Chebyshev puesto que el diseño del mismo se basa en los polinomios de Chebyshev.

Hay dos variantes: Tipo I (rizado en la banda de paso) y Tipo II (rizado en la banda de rechazo)

Filtros analógicos tradicionales

Los filtros **elípticos** (también conocidos como filtros de Cauer) **acotan el rizado** tanto en la banda de paso como en la de rechazo. El **nivel de rizado en cada banda se puede ajustar de forma independiente** y, entre los filtros de igual orden, tiene la **transición de ganancia más rápida** entre la banda de paso y la banda de rechazo.

(Si se desea, pueden converger a los filtros de Chebyshev)



6.4 DISEÑO DE FILTROS DIGITALES CON MATLAB

Matlab ofrece múltiples herramientas para el análisis y diseño de filtros digitales:

Toolboxes (librerías): **Signal Processing** y **Filter Design**

Funciones matemáticas genéricas: `filter()`, `conv()`,

Funciones específicas: `hamming()`, `window()`, `butter()`, `cheb1ord()`, `cheb2ord()`, `ellipord()`, ...

Nosotros nos centraremos en la descripción de una Interfaz gráfica para el diseño y análisis de filtros digitales: `FDATool`.

fdatool

Syntax: `fdatool`

Descripción básica:

Esta herramienta se utiliza para:

- Diseñar filtros
- Cuantizar filtros
- Analizar filtros
- Modificar diseños de filtros ya existentes
- Crear filtros multitasa
- Realizar modelos de Simulink de filtros FIR
- Importar filtros a FDATATool
- Realizar transformaciones de frecuencia para filtros digitales

La descripción detallada puede encontrarse en la ayuda de Matlab (contiene incluso tutoriales y demostraciones gráficas de su funcionamiento)

Abriendo fdatool

Filter Design & Analysis Tool - [untitled.fda]

File Edit Analysis Targets View Window Help

Current Filter Information

Structure: Direct-Form FIR
Order: 50
Stable: Yes
Source: Designed

Store Filter ...
Filter Manager ...

Filter Specifications

Mag. (dB)

0

F_{pass} F_{stop} $F_s/2$ f (Hz)

A_{pass} A_{stop}

Response Type

Lowpass
 Highpass
 Bandpass
 Bandstop
 Differentiator

Design Method

IIR Butterworth
 FIR Equiripple

Filter Order

Specify order: 10
 Minimum order

Options

Density Factor: 20

Frequency Specifications

Units: Hz
Fs: 48000
Fpass: 9600
Fstop: 12000

Magnitude Specifications

Units: dB
Apass: 1
Astop: 80

Design Filter

Ready

fdatool: opciones

Eligiendo el tipo de respuesta

Eligiendo el método de diseño del filtro

Fijando las especificaciones del filtro

Calculando los coeficientes del filtro

Analizando el filtro

Editando el filtro utilizando el editor de polos y ceros

Modificando el filtro a través de la modificación de polos y ceros

Importando un filtro

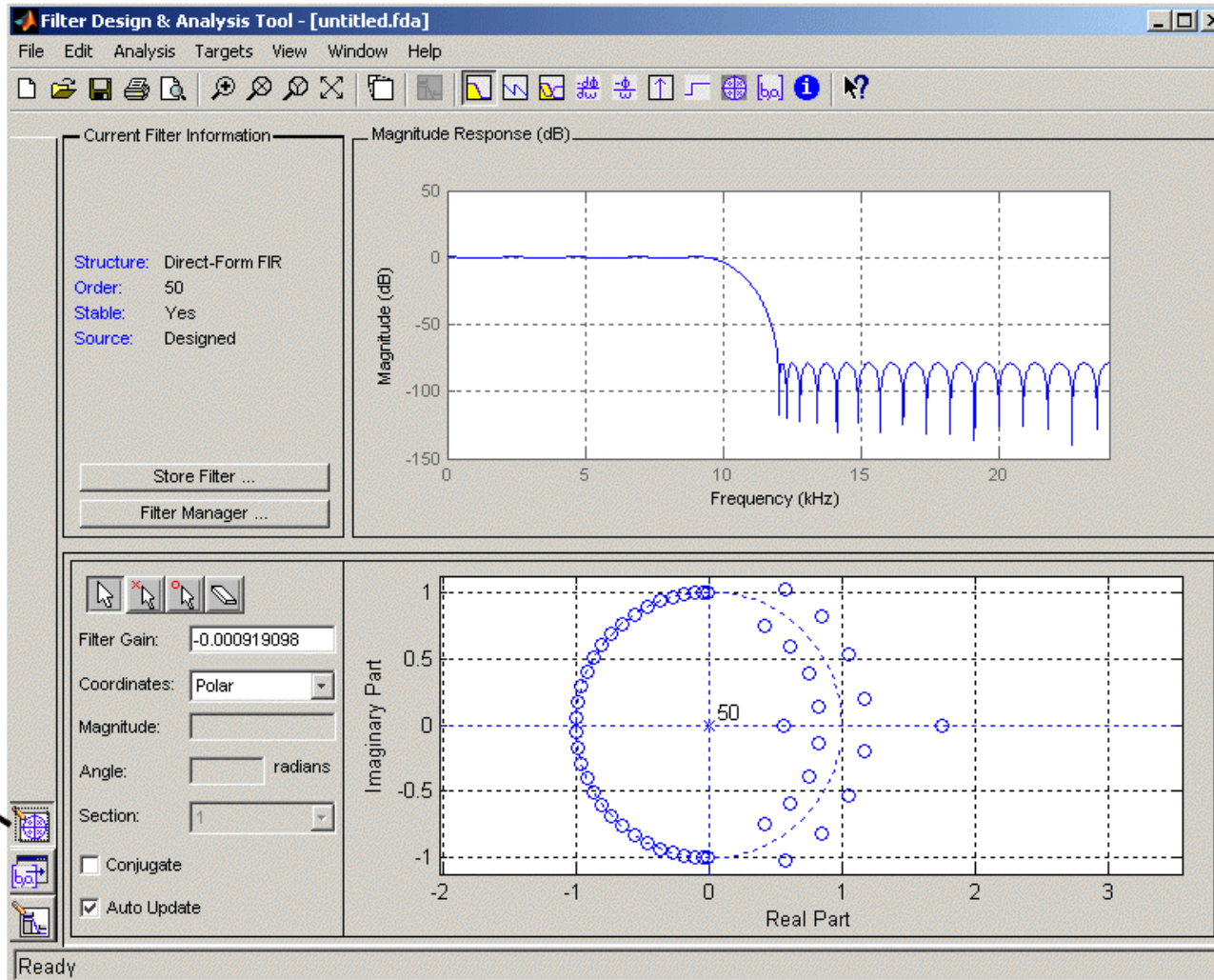
Exportando un filtro (.m, C, ...)

Analizando el filtro



- ◆ Magnitude response
- ◆ Phase response
- ◆ Magnitude and Phase responses
- ◆ Group delay response
- ◆ Phase delay response
- ◆ Impulse response
- ◆ Step response
- ◆ Pole-zero plot

Editando el filtro



Pole/Zero Editor

REFERENCIAS Y BIBLIOGRAFÍA

A. V. Oppenheim, R. W. Schaffer, “Discrete-Time Signal Processing”, 2a Ed., Upper Saddle River, NJ: Prentice-Hall, 1999. (Chapter 7).

D. Jones, “Digital filter Design Institution”, On-line Course, Rice University.

A. Antoniou, “Digital filtros: Analysis, Design, and Applications”, New York, NY: McGraw-Hill, 1993.

Ayuda del programa MATLAB 7.1 (Mathworks)

Referencias históricas:

-S. Butterworth, “On the Theory of filtro Amplifiers”, Wireless Engineer (also called Experimental Wireless and the Radio Engineer), vol. 7, pp. 536-541, 1930.

-T.W. Parks and J.H. McClellan, “Chebyshev Approximation for Nonrecursive Digital filtros with Linear Phase”, IEEE Trans. Circuit Theory, vol. 19, pp. 189-194, Mar. 1972.

-A.G. Deczky, “Synthesis of Recursive Digital filtros Using the Minimum p-Error Criterion”, IEEE Trans. Audio Electroacoust., vol. 20, pp. 257-263, Oct. 1972.