



Ampliación de Señales y Sistemas

Tema 6:

Introducción al diseño de filtros digitales

Esquema

Tema 1: Señales y sistemas discretos en el dominio del tiempo.

Tema 2: Señales y sistemas discretos en el dominio de la frecuencia.

Tema 3: Muestreo.

Tema 4: Fundamentos de la Transformada Discreta de Fourier.

Tema 5: Transformada Z.

Tema 6: Introducción al diseño de filtros discretos.

- 6.1. Fundamentos del filtrado digital.
- 6.2. Diseño de filtros digitales:
 - 6.2.1. Diseño de filtros FIR.
 - 6.2.2. Diseño de filtros IIR.
- 6.3. Diseño de filtros digitales con Matlab.

Comentarios:

- Biblio: [BB3: McC&Sch&Yod] Cap. 5, Cap. 10; [BB2: Opp&Sch] Cap. 7;
[BB4: Lathi] Cap. 12;

Fundamentos del filtrado digital

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



Aplicaciones:

Comunicaciones

Sismología

Procesado de voz

Tratamiento de imagen y vídeo

Radar/Teledetección

Bio-Ingeniería

Minería de datos

Reconocimiento de patrones

Fundamentos del filtrado digital

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.

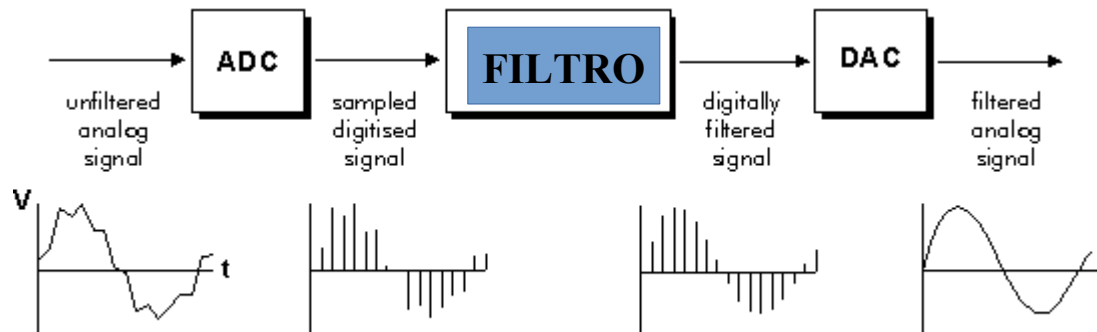
Filtros analógicos:

- 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 análogicas (muestreando previamente)



Principales ventajas de los filtros digitales

- Un filtro digital es **programable**, por lo que puede ser modificado con facilidad.
- Los filtros digitales **se diseñan, prueban e implementan de manera sencilla**.
- Los filtros digitales son muy **versátiles** (filtros adaptativos).
- El comportamiento de los filtros digitales **no cambia en el tiempo** por derivas o deterioro de sus componentes.
- Las necesidades **hardware** son relativamente **simples y homogéneas**.
- Los filtros digitales **trabajan con señales de baja frecuencia** de forma **precisa**.
- 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 digitales:

Paso 1. Determinación de las especificaciones del filtro a partir de los requisitos de la aplicación para la que se diseña el filtro:

- Bandas de frecuencia deseadas.
- Tolerancias.

Paso 2. Cálculo de los coeficientes del filtro que mejor aproximan las especificaciones de la respuesta en frecuencia deseada:

- ¿Sistema causal?
- Decidir qué tipo de filtro se ha de utilizar: FIR o IIR.

Paso 3. Implementación del filtro:

- Pueden utilizarse diferentes estructuras.

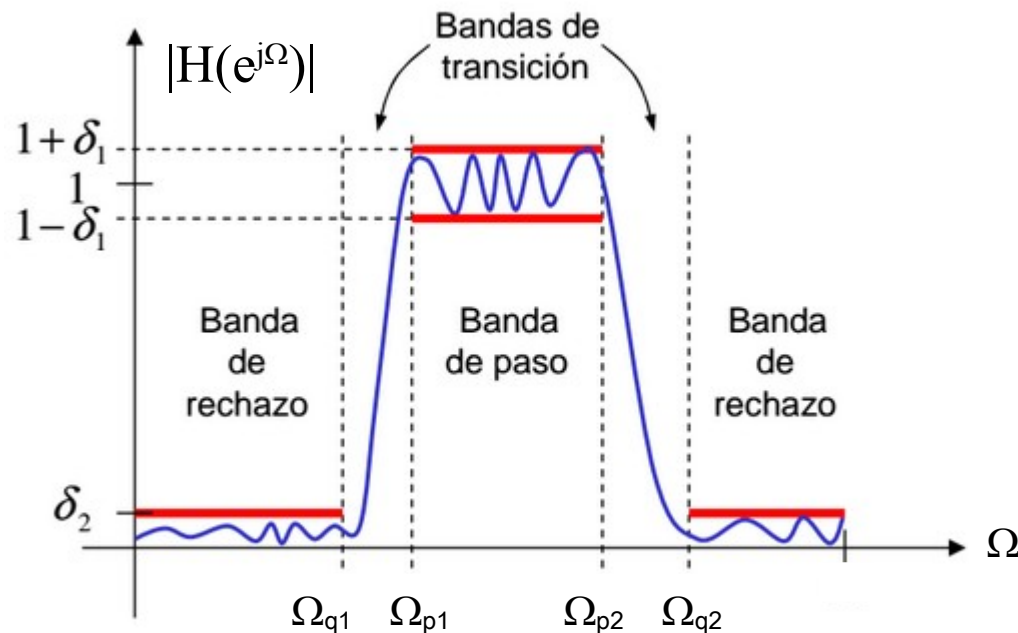
Diseño de filtros digitales

Paso 1: Especificaciones

Se suele realizar en el dominio de la frecuencia, especificando la respuesta en frecuencia deseada por medio de unos parámetros.

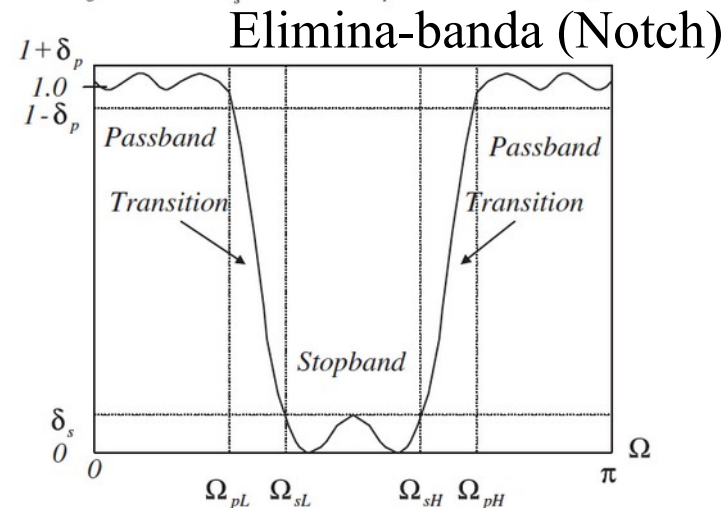
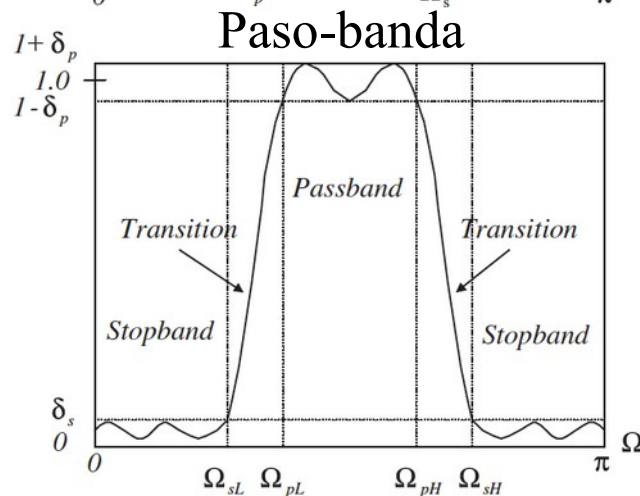
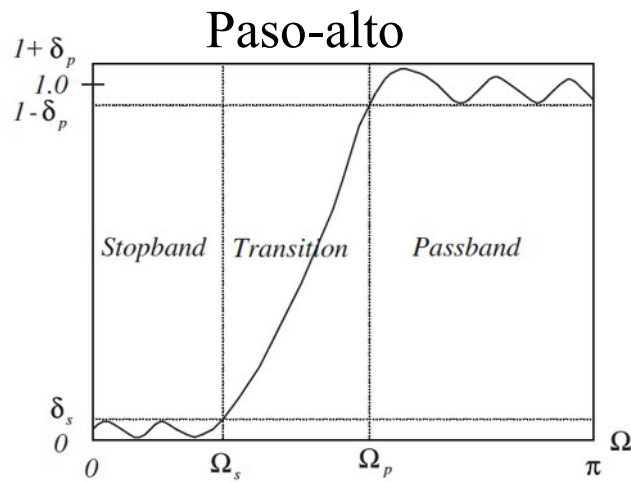
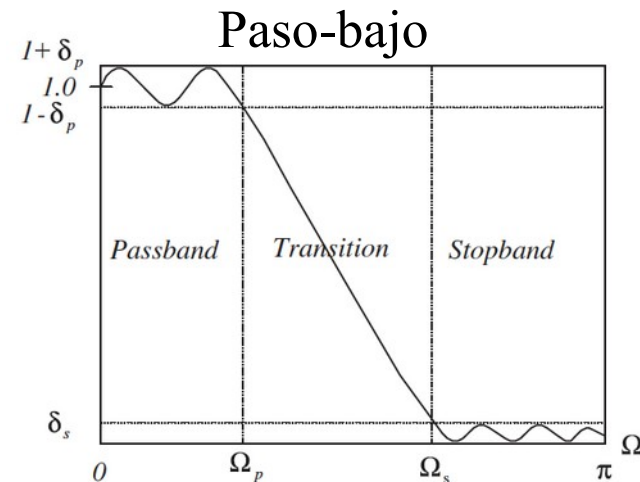
P.e., los parámetros utilizados en la especificación de un filtro paso-banda son:

$$\delta_1, \delta_2, \Omega_{q1}, \Omega_{p1}, \Omega_{p2}, \Omega_{q2}$$



Diseño de filtros digitales

Paso 1. Respuesta en frecuencia deseada



Los principales tipos de filtros son: **paso-bajo** (deja pasar las bajas frecuencias y atenúa las altas), **paso-alto** (deja pasar las altas frecuencias y atenúa las bajas), **paso-banda** (deja pasar una banda de frecuencias) y **elimina-banda** (atenúa una banda de frecuencias).

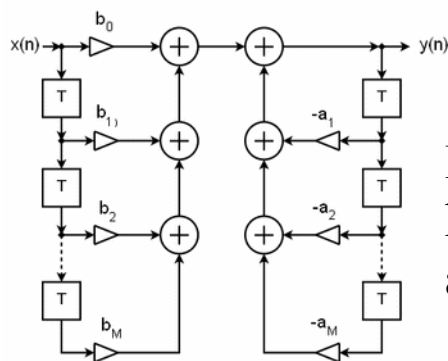
Diseño de filtros digitales

Paso 2: Elegir un tipo de filtro

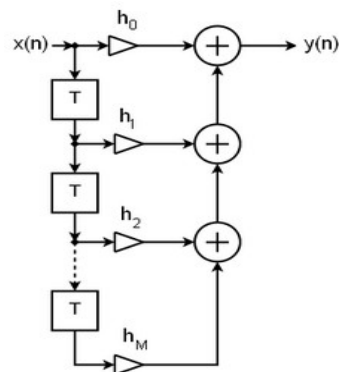
Hay dos tipos básicos de filtros digitales:

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

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



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).

Paso 3: Estructuras

- Existen muchas formas de realizar las operaciones en un sistema LTI discreto.
- Cada estructura busca minimizar uno de los siguientes objetivos:
 - Tiempo de cálculo
 - Memoria
 - Número de bits de los coeficientes
 - Número de bits de las muestras
 - ...

FIR vs IIR

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

Ventajas de los filtros FIR:

- Conceptualmente sencillos y fácilmente implementables.
- Pueden implementarse vía convolución.
- Siempre son estables.
- Poco sensibles a la cuantización de la amplitud.
- Pueden tener fase lineal (mismo retardo temporal para todas las frec.).
- Se usan cuando se requiere fase lineal.

Ventajas de los filtros IIR:

- Más adecuados para la aproximación/adaptación de diseños analógicos.
- 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).
- Se usan cuando no se requiere fase lineal pero sí ahorrar costes.
- Tienen lóbulos secundarios menores en la banda de rechazo que un FIR con el mismo número de parámetros.
- Si se tolera distorsión de fase, mejor un IIR porque requiere menos parámetros que un FIR → menos memoria y menor complejidad.

DISEÑO DE FILTROS FIR

Métodos para el diseño de filtros FIR:

- Enventanado: se basa en calcular la IDTFT de la respuesta en frecuencia y truncarla utilizando alguna ventana para suavizar su efecto. En MATLAB: `fir1`, `fir2`.
- Basado en el muestreo de la respuesta en frecuencia: se especifica la respuesta en frecuencia deseada del filtro en M puntos equiespaciados y se calcula $h[n]$ como la DFT inversa de esas muestras. Como existen cuatro posibles respuestas en frecuencia para los filtros FIR de fase lineal aparecerán cuatro expresiones distintas para el cálculo de $h[n]$.
- Por optimización: se basan en minimizar el error entre la respuesta en frecuencia deseada y la obtenida con el filtro según algún criterio:
 - Mínimos cuadrados (minimiza error cuadrático). MATLAB: `firls`
 - Parks-McClellan o “equiripple” (rizado uniforme). MATLAB: `remez`
 - Otros.

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 respuesta en frecuencia encontrar la respuesta al impulso óptima) no puede resolverse de forma analítica.

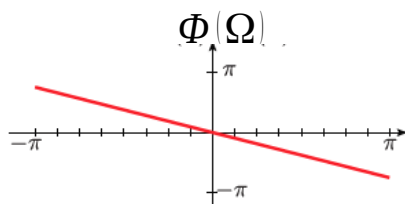
DISEÑO DE FILTROS FIR (cont.)

Son de la forma:

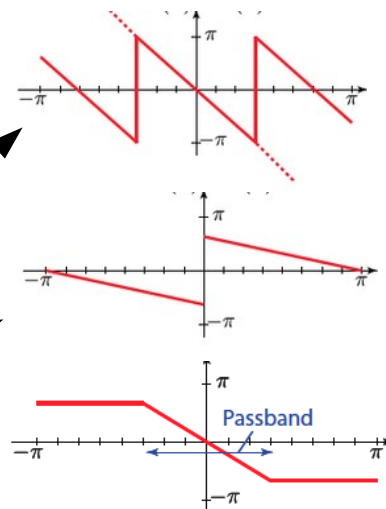
$$y[n] = \sum_{k=0}^{M-1} h[k] x[n-k]$$

Ventaja: Se pueden diseñar para que su respuesta en fase sea lineal, es decir:

$$H(e^{j\Omega}) = |H(e^{j\Omega})| e^{j\Phi(\Omega)}, \text{ con } \Phi(\Omega) = \pm A\Omega + B$$



una recta en la banda de
paso (*Passband*),
también sirven



En ese caso, su retardo de grupo (la pendiente de la recta en negativo) es cte \rightarrow ¡Una respuesta en fase lineal genera el mismo retardo temporal para todas las frecuencias! (Recuérdese la propiedad de desplazamiento en la TF.) Ésta es una característica deseable cuando la forma temporal de la señal es importante (p.e. para su visualización/sincronismo en un osciloscopio).

¿Qué debe cumplir $h[n]$ para que tenga una fase lineal?:

Simetría en los coeficientes del filtro:

$$h[n] = \pm h[M-n], n = 0, \dots, M-1$$

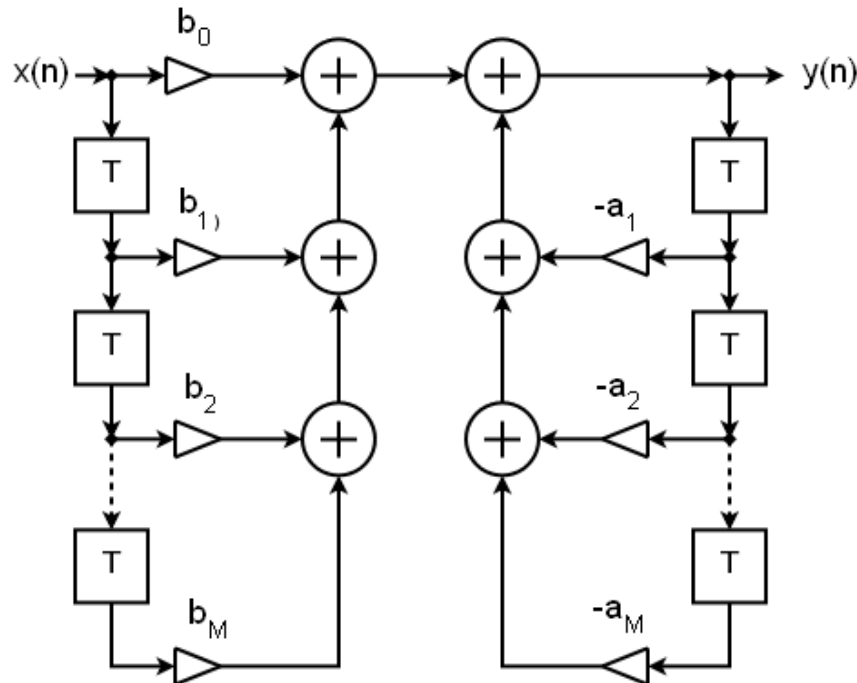
¿Por qué la fase lineal es importante?

Para mantener la “estructura” original de la señal en las frecuencias de la banda de paso.
Se entiende muy bien viendo el módulo y la fase de la DFT de una imagen.

DISEÑO DE FILTROS IIR

Son filtros de la forma:

$$\sum_{k=0}^{N-1} a_k y[n-k] = \sum_{k=0}^{M-1} b_k x[n-k]$$



- Son la alternativa a los FIR para conseguir mejores prestaciones. Para el mismo número de parámetros, obtienen mejor compromiso entre ringing y banda de transición.
- Consiguen cumplir unas determinadas especificaciones con un orden mucho menor que el necesario con filtros FIR.
- Si el sistema tienen polos es un sistema IIR \rightarrow pueden ser inestables. Si no tienen polos (exceptuando los polos en $z=0$ y $z=\infty$), el sistema es FIR.
- NO pueden tener fase lineal \rightarrow no existe ningún punto posible de simetría

El diseño de filtros IIR se reduce a: elegir los $\{a_i\}$ y $\{b_i\}$ que mejor aproximan la respuesta $|H_d(e^{j\Omega})|$ deseada.

DISEÑO DE FILTROS IIR (cont.)

Métodos para el diseño de filtros IIR.

- **Indirectos o derivados de métodos para filtros analógicos:** se basan en diseñar filtros analógicos que verifiquen las especificaciones y luego transformarlos en digitales preservando estas características. Tipos de filtros analógicos tradicionales:

- Butterworth. MATLAB: `butter`
- Chebyshev 1 y 2. MATLAB: `cheby1`, `cheby2`
- Elíptico. MATLAB: `ellip`
- Bessel. MATLAB: `besself`

- **Directos:** se obtienen a partir de la salida deseada para una entrada dada. Se basan en determinar los coeficientes del filtro minimizando el error entre la respuesta en frecuencia deseada y la obtenida por algún criterio, p.e., el de mínimos cuadrados. Se utilizan cuando conocemos la respuesta al impulso de un sistema y queremos determinar la función de transferencia que mejor se ajusta. Ejemplo en MATLAB: `yulewalk`.

Diseñar filtros discretos a partir de sus especificaciones

Pasos a seguir:

1. Decidir tipo de respuesta al impulso (FIR o IIR). Se supone que debe ser causal y estable.
 - Si el filtro es FIR, decidir si debe ser de fase lineal.
2. Decidir las bandas de frecuencia a dejar pasar (banda de paso), rechazar (banda eliminada), así como las bandas de transición para pasar de una banda a otra.
3. Decidir el rizado de la amplitud en la banda de paso y el rizado de la amplitud en la banda eliminada o atenuada.

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()`, ...

Además, Matlab ofrece una interfaz gráfica para el diseño y análisis de filtros digitales: `filterDesigner`

Ejecute en la línea de comandos de matlab: `filterDesigner`

Diseño de filtros digitales con Matlab

Descripción básica de `filterDesigner`

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
- Realizar transformaciones de frecuencia para filtros digitales

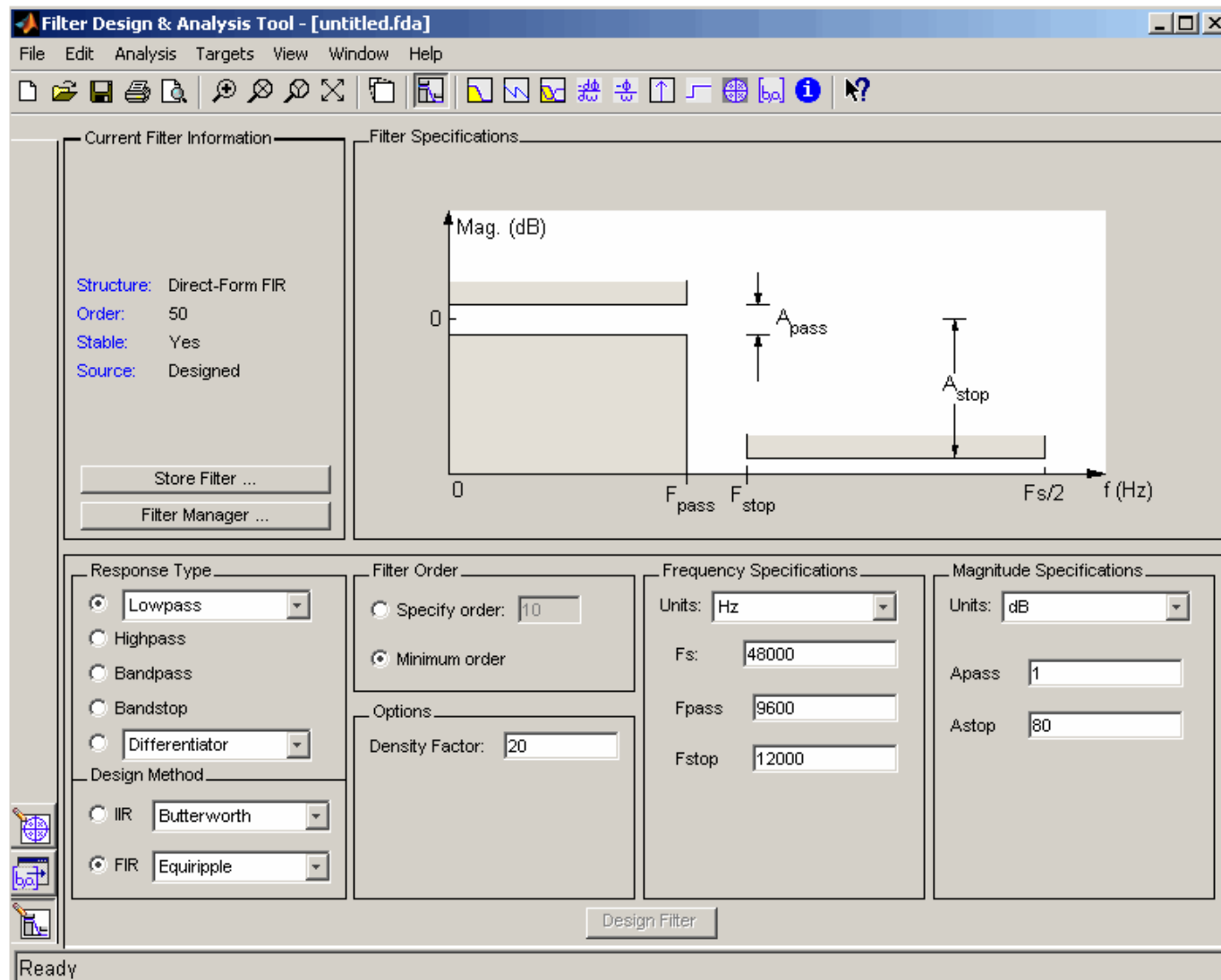
La descripción detallada puede encontrarse en la ayuda de Matlab:

<https://es.mathworks.com/help/signal/ug/getting-started-with-filter-designer.html>

Contiene tutoriales y demostraciones gráficas de su funcionamiento.

Diseño de filtros digitales con Matlab

Abriendo filterDesigner



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, ...)

Diseño de filtros digitales con Matlab

Editando el filtro

