

Diseño y análisis de filtros en procesamiento de audio

Marta Ruiz Costa-jussà
Helenca Duxans Barrobés

PID_00188065



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació para la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

Índice

Introducción	5
Objetivos	6
1. Concepto y tipos de filtros. ¿Por qué se debe aprender cómo funciona un filtro?	7
2. Conceptos básicos para el diseño de filtros digitales	11
2.1. Caracterización de un filtro: respuesta impulsional y función de transferencia	11
2.2. Especificación de la entrada-salida de un filtro: ecuaciones en diferencias finitas	15
3. Filtros reales: plantilla de especificación de un filtro	19
4. Diseño de filtros digitales	22
4.1. Diseño de filtros FIR	22
4.1.1. Método de ventanas	22
4.1.2. Método de Parks-McClellan	26
4.2. Diseño de filtros IIR	29
4.2.1. Transformación bilineal	30
4.2.2. Diseño de filtros analógicos: breve resumen	31
4.2.3. Transformación bilineal y teoría de filtros analógicos ...	33
5. De la teoría a la práctica	35
5.1. La elección de un filtro FIR o IIR. ¿Cuándo debemos utilizar uno u otro?	35
5.2. Caso práctico: diseño de un filtro desde cero con Audacity y con Matlab	36
5.2.1. Caso 1. Filtrado de una señal de audio utilizando Audacity	36
5.2.2. Caso 2. Filtrado de una señal de audio utilizando Matlab	36
Anexo	39

Introducción

El objetivo principal de este módulo es explicar la **teoría básica de diseño de filtros digitales y analógicos**. Concretamente, se pretende dar al estudiante la capacidad de aprender a utilizar cualquier **software** de diseño de filtros y que así pueda diseñar sus propios filtros dadas unas especificaciones concretas. Asimismo, se combina la teoría y la práctica del **diseño de filtros** con el estudio de diferentes utilidades de los filtros digitales y analógicos en el área de procesamiento de la señal de audio.

El proceso de diseño de un filtro digital engloba básicamente tres etapas: las **especificaciones** de las propiedades que queremos que verifique el filtro, la **determinación de los coeficientes** de un sistema causal que se aproxime a las especificaciones queridas y, finalmente, la **realización del sistema**.

Objetivos

Concretamente, al acabar de trabajar este módulo, el estudiante debe:

1. Conocer, saber diseñar e interpretar la respuesta impulsional que caracteriza un filtro.
2. Conocer los parámetros de diseño de un filtro: banda de paso y de atenuación, entre otros.
3. Diferenciar entre filtrado analógico y digital.
4. Conocer diferentes métodos para diseñar los filtros digitales y analógicos.
5. Tener la capacidad, dada una aplicación que requiera filtrado digital, para elegir el tipo de filtros digitales (FIR/IIR) más adecuado según las especificaciones en módulo y fase del sistema.

1. Concepto y tipos de filtros. ¿Por qué se debe aprender cómo funciona un filtro?

Un filtro, genéricamente hablando, es un sistema que discrimina parte del objeto que procesa, según uno o más atributos. Por ejemplo, un filtro de aire deja pasar el aire pero impide el paso a las partículas de polvo presentes en el aire. Un filtro en procesamiento de señal es un concepto muy amplio, dado que puede ser cualquier sistema que trate la señal. Nos centraremos en los sistemas o filtros que, dada cualquier señal, permiten el paso de un conjunto determinado de frecuencias y atenúan o eliminan el resto.

Partiendo de la misma definición de filtro, podemos definir los tipos clásicos de filtro siguientes:

- Un filtro **paso bajo** permite el paso de las frecuencias bajas y atenúa las frecuencias altas.

Ejemplo

Por ejemplo, una pared hace de filtro paso bajo cuando oímos la música que viene de otra habitación. Dado que las frecuencias altas (es decir, los sonidos agudos) tienden a reflejarse sobre cualquier superficie rígida (por ejemplo, las paredes), en nuestra habitación estas frecuencias llegan atenuadas y solo oímos las frecuencias bajas o graves de la música original. Por lo tanto, la señal original (la música) ha sido filtrada paso bajo.

- Un filtro **paso alto** permite el paso de frecuencias altas y atenúa las frecuencias bajas. Se pueden utilizar para hacer que las frecuencias altas de una señal de audio vayan a unos altavoces para sonidos agudos.
- Un filtro **paso banda** permite el paso de frecuencias de una determinada banda de frecuencias, eliminando las frecuencias bajas y altas. Encontramos filtros paso banda en la transmisión y recepción de la señal, para eliminar el contenido frecuencial que no se encuentra en la banda de la señal que se ha de transmitir.
- Un filtro de **banda eliminada** es complementario al filtro paso banda porque elimina las frecuencias de una determinada banda de frecuencias y deja pasar las frecuencias que hay fuera de esta banda. Se utilizan frecuentemente en ecualizadores.

Nota

Un ecualizador procesa señales de audio y es capaz de modificar la amplitud de las frecuencias, entre otras características. En la práctica, se utiliza para adaptar el sonido a las necesidades del oyente.

La figura 1 muestra un ejemplo ideal de cada uno de estos filtros:

Representación frecuencial de algunos filtros ideales

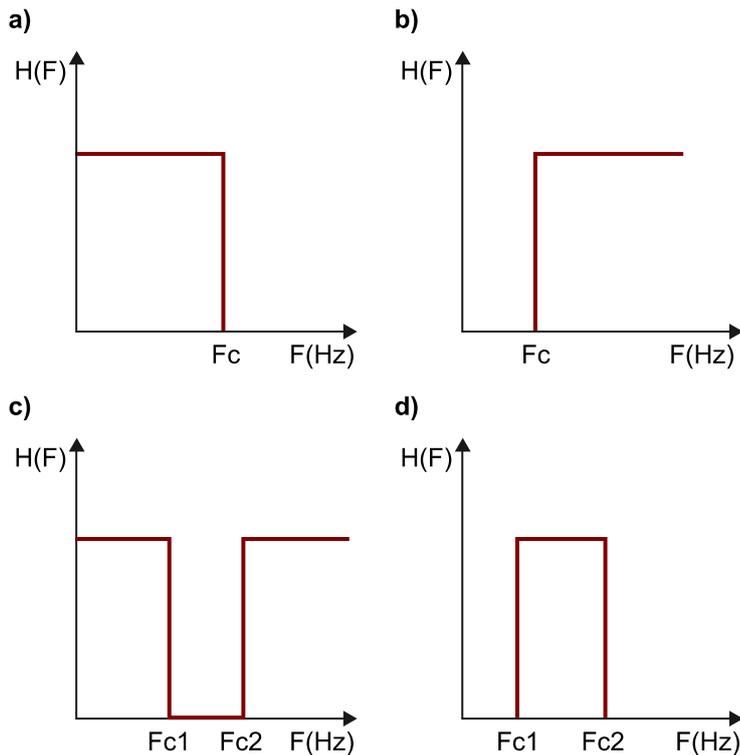


Figura 1. Representación frecuencial de los filtros ideales siguientes: a) paso bajo; b) paso alto; c) de banda eliminada; d) paso banda. Los filtros son simétricos respecto al eje de ordenadas.

La figura 1 muestra el módulo de los filtros ideales. Respecto a la fase, un filtro ideal tiene la fase lineal. Esto significa que la fase de la respuesta del filtro es una función lineal de la frecuencia. Por lo tanto, la señal filtrada no tiene distorsión de fase.

El concepto de fase lineal y no lineal aparece explicado con detalle en la página 428 y siguientes del libro *Señales y sistemas*, de A. V. Oppenheim y A. S. Willsky.

Sin ser conscientes, en el día a día, constantemente topamos con filtros acústicos, tanto si son **analógicos** (tratan señales de variable continua, como por ejemplo t) como **digitales** (tratan señales de variable discreta, como por ejemplo n o, dicho de otro modo, listas de números ordenados). Por ejemplo, cuando hablamos por teléfono, las frecuencias de la señal que se transmite se filtran a 4 kHz (si se trata de una línea analógica, el filtrado es analógico), y esto es una limitación muy fuerte comparada con lo que el oído humano es capaz de oír (hasta 20 kHz). Por este motivo, escuchar un CD de música por la línea telefónica no es tan agradable como escucharlo directamente. La grabación del CD de música se filtra a 22 kHz para asegurar que se mantiene todo lo que el oído humano puede oír.

Los músicos electrónicos casi siempre utilizan filtros digitales en cada pieza de música que crean; por ejemplo, para hacer reverberación digital.

La reverberación es solo un pequeño ejemplo de las capacidades de los filtros digitales. Debemos ser conscientes de que un filtro digital es capaz de manipular el espectro de un sonido, como veremos en el módulo 6, de efectos de audio. Un objetivo de esta asignatura es ayudar a los estudiantes a tener una idea sobre las herramientas necesarias para diseñar filtros personalizados.

Existe bastante software disponible para el diseño de filtros digitales (en la asignatura utilizaremos Matlab y Octave). Ahora bien, para utilizar el software es esencial el conocimiento teórico de cómo funcionan los filtros digitales. Incluso en aplicaciones estándar, el uso eficaz de un programa de diseño de filtros requiere una comprensión de los parámetros de diseño, que al mismo tiempo requiere una cierta comprensión de la teoría de filtros.

Reverberación

El concepto de reverberación se estudia en el módulo de introducción a la acústica. Brevemente, la reverberación es el fenómeno acústico por el cual un receptor (oyente) recibe al mismo tiempo el sonido directo que emite un emisor (fuente acústica) y las reflexiones de este sonido producidas por la sala donde están el emisor y el receptor, lo que mejora la calidad del sonido y crea una sensación de calidez en el oyente.

⁽¹⁾Para pasar de una figura a otra, utilizamos la transformada discreta de Fourier.

Ejemplo

Veamos el efecto de filtrar un sonido paso bajo. Tenemos el sonido de un canario. La figura siguiente representa la forma de onda en el tiempo y en frecuencia.

Sonido de canario

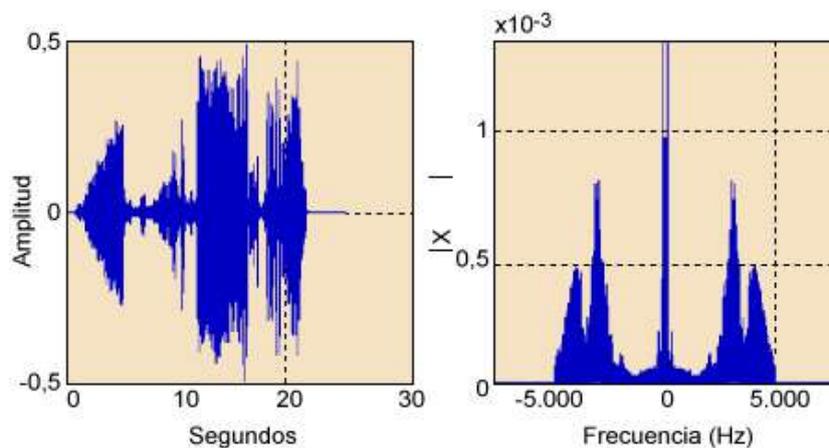


Figura 2. Sonido de canario

Si filtramos¹ con un paso bajo a 2 kHz, tenemos las señales siguientes:

Sonido de canario filtrado

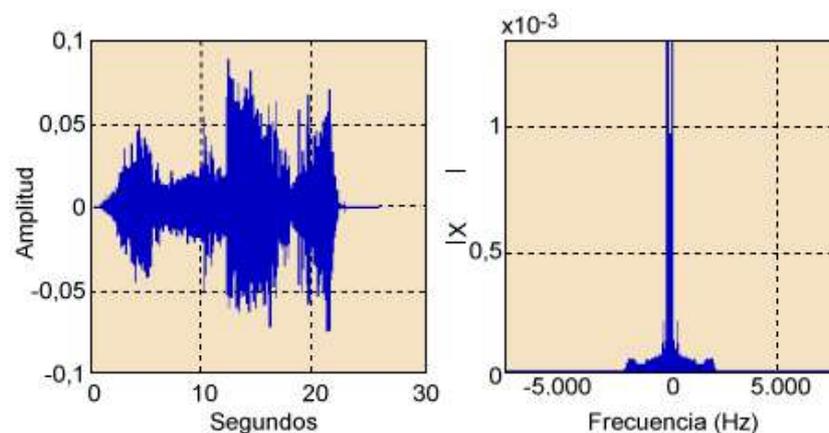


Figura 3. Sonido de canario filtrado

Observad que ahora el sonido del canario se ha suavizado.

Como hemos visto en el apartado “Conversión A/D y D/A: entorno analógico y entorno digital” del módulo 1, en la naturaleza encontramos señales analógicas, como es el caso de la voz humana; por ello, si las queremos manipular con ordenadores hay que convertirlas en digitales –este proceso se denomina *conversión analógico-digital (A/D)*. A continuación, una vez se ha manipulado la señal digital, por ejemplo filtrándola, esta señal se debe volver a convertir en señal analógica –este paso se denomina *conversión digital-analógico (D/A)*. Por lo tanto, podemos decir que hay muchos filtros digitales que trabajan en un entorno analógico convirtiendo las señales analógicas en digitales y, después de la manipulación con el ordenador, volviéndolas a convertir en analógicas.

2. Conceptos básicos para el diseño de filtros digitales

Un **filtro digital clásico** se puede definir como un sistema que modifica una señal digital de manera que deja pasar un determinado intervalo de frecuencias y atenúa otras con el objetivo de obtener una salida con unas características determinadas. Los filtros “clásicos” trabajan en el espacio frecuencial y las plantillas se diseñan en este espacio.

Nota

Una señal discreta es una señal muestreada pero no cuantificada y una señal digital es una señal muestreada y cuantificada. Por ejemplo, una señal discreta es $x = [0,1 \ 0,4 \ 0,9]$ y la misma señal digital es $x_d = [0 \ 0 \ 1]$ (con un paso de cuantificación de 1).

Concretamente, estudiaremos los filtros digitales **realizables**. Por esta razón, estos sistemas tienen que verificar ciertas propiedades: la linealidad, la invariancia en el tiempo, la causalidad y la estabilidad.

2.1. Caracterización de un filtro: respuesta impulsional y función de transferencia

Como hemos dicho (y como hemos visto en las asignaturas de *Señales y sistemas I y II*), en el dominio temporal la respuesta impulsional relaciona la entrada y la salida de un sistema lineal e invariante en el tiempo (SLIT).

Si hablamos de sistemas digitales, que es el dominio que trataremos más (es decir, filtros digitales), la salida es la suma de convolución entre la entrada $x[n]$ y la respuesta impulsional $h[n]$.

Esta relación la vemos representada en la ecuación y figura siguientes:

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$$

Ved también

Hemos visto la definición de sistema en el apartado “Definición de sistema” del módulo “Conceptos de señales y sistemas”.

Lectura recomendada

Encontraréis más información sobre la suma de convolución en las páginas 77 a 90 del libro *Señales y sistemas*, de A. V. Oppenheim y A. S. Willsky.

Relación genérica de entrada y salida de un sistema digital con entrada y salida

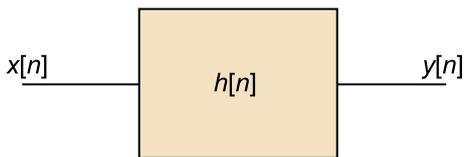


Figura 4. Relación genérica de entrada y salida de un sistema digital $h[n]$ con entrada y salida

La principal ventaja de un filtro lineal e invariante, de ahora adelante **SLTI** (sistema lineal e invariante en el tiempo o *linear time invariant system*), es que se puede caracterizar por la **respuesta impulsional** en el dominio del tiempo y por la función de transferencia en el dominio Z. Solo los filtros SLTI verifican la propiedad de **superposición (propiedad de linealidad)**: así, podemos asegurar que, con un filtro SLTI, el filtrado de una suma de señales de audio se puede obtener como la suma de cada señal filtrada por separado. Por otro lado, las propiedades de causalidad y estabilidad permiten que el filtro sea realizable. Por lo tanto, los filtros representados en la figura 1 son filtros **ideales** o **norealizables**, puesto que la respuesta impulsional es no causal e inestable.

Así, en el dominio Z, la función de transferencia $H(z)$ relaciona la entrada y la salida de un sistema LTI. Por lo tanto, la función de transferencia es la transformada Z de la respuesta impulsional:

$$y[n] = x[n] * h[n] \xrightarrow{TZ} Y(z) = X(z)H(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{n=0}^{\infty} h[n]z^{-n}$$

Finalmente, en el dominio frecuencial, que es una particularización del dominio Z, la respuesta frecuencial es la transformada de Fourier de secuencias discretas (TFSD) de la respuesta impulsional:

$$H(f) = \sum_{n=0}^{\infty} h[n]e^{-j2\pi fn}$$

En el caso de los filtros, la respuesta impulsional y la función de transferencia que tienen determinan los conceptos siguientes:

- La ganancia ($G(f)$) se define como la amplificación de la señal de salida respecto a la entrada. Si esta amplificación es negativa, hablamos de atenuación:

$$G(f) = 10 \log \frac{|H(f)|^2}{H_{ref}^2}$$

Referencias

En cuanto a la transformada Z y a la transformada discreta de Fourier, podéis ver el apartado “Transformación del dominio temporal al dominio frecuencial” del módulo 1.

Encontraréis más información sobre módulo y fase en las páginas 427 a 430 del libro *Señales y sistemas*, de A. V. Oppenheim y A. S. Willsky.

Recordad

Estamos utilizando la variable para indicar frecuencia discreta, y equivale, por lo tanto, a la frecuencia analógica normalizada por la frecuencia de muestreo.

- La respuesta en amplitud de un filtro se define como el módulo de la respuesta frecuencial del filtro:

$$A(f) = |H(f)|$$

Así, la respuesta en amplitud de un filtro ideal es 1 en la banda de paso y 0 en la banda atenuada.

- La respuesta en fase de un filtro se define como la fase de la respuesta impulsional:

$$\theta(f) = \angle H(f)$$

- El orden de un filtro es el número de coeficientes que tiene un filtro y coincide con el máximo retardo (en términos de muestras) que se utiliza o de la señal de entrada o de salida anteriores para calcular $y[n]$. Por otra parte, el orden de un filtro se puede definir como el orden de la respuesta impulsional. Y, a la vez, el orden de un polinomio se define como el valor más alto del exponente del mismo polinomio. En el caso de una función racional, el orden se define como el máximo entre el orden del polinomio del numerador y el del denominador.

Así, por ejemplo, un filtro definido por la ecuación siguiente tiene orden 3:

$$y[n] = x[n] + x[n-1] - y[n-3]$$

- La banda de paso de un filtro es el rango de frecuencias que el filtro deja pasar de la entrada a la salida con una atenuación, por ejemplo, más pequeña de 3 dB, o ganancia $G(f) = -3$ dB (este valor se toma por convención y puede variar dependiendo de las especificaciones del sistema). Y la banda atenuada es la complementaria. Por extensión, las frecuencias de paso y de atenuación definen el rango de banda de paso y atenuada. En el caso de un filtro ideal, la frecuencia de paso y atenuación coinciden en la frecuencia de corte. Veamos la figura siguiente:

Banda de paso y atenuada en un filtro paso bajo ideal

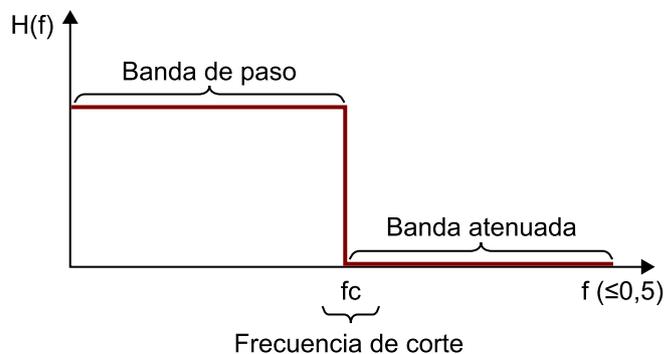


Figura 5. Banda de paso y atenuada en un filtro paso bajo ideal

Nota

La f no va en hercios. Recordad que es la frecuencia analógica normalizada y se puede decir que $f \leq 0,5$.

Definiciones

Un **decibelio** (dB) se puede definir como una medida logarítmica de potencia en relación con una potencia patrón. Básicamente, dicho de otro modo, es una unidad no dimen-

sional que representa la relación entre una señal de interés y una señal de referencia. Un dB se basa en un logaritmo de base 10 del cociente entre las dos magnitudes.

La **frecuencia de corte** de un filtro es la frecuencia a partir de la cual la señal de salida se atenúa significativamente. Normalmente, se considera que la señal de entrada se debe atenuar un valor de 3 dB (o 3 dB si hablábamos de ganancia). La relación entre ganancia y atenuación viene dada por $A(f) = -G(f)$.

La respuesta frecuencial se caracteriza por su módulo $|H(f)|$ y fase $\theta_h(f)$: $H(f) = |H(f)|e^{j\theta_h(f)}$. En un caso genérico, si tenemos la señal en frecuencia de entrada $X(f)$ y el filtro anterior, entonces el módulo y la fase de la señal de salida ($Y(f)$) son los siguientes:

$$|Y(f)| = |H(f)||X(f)|$$

$$\theta_y(f) = \theta_h(f) + \theta_x(f)$$

El módulo $|H(f)|$ determina la ganancia del filtro (recordad la definición de ganancia) y la fase $\theta_h(f)$ determina el retardo del filtro. Una propiedad muy importante relacionada con la fase de la respuesta del filtro es el retardo de grupo:

$$\tau_g = -\frac{1}{2\pi} \frac{d\theta}{df}$$

El retardo de grupo evalúa la señal de salida respecto al de entrada en muestras para cada frecuencia. Por lo tanto, se evalúa el retardo global que experimenta una señal. Si el retardo de grupo es constante, quiere decir que la fase es lineal (retarda todas las frecuencias del mismo modo). En caso de que el retardo de grupo sea diferente según la frecuencia, se produce una distorsión en el espectro de la señal.

Si el **módulo es constante** (H_0) y la **fase es lineal en frecuencia** ($f \cdot m$), tenemos la relación entrada-salida siguiente:

$$Y(f) = H(f)X(f) = H_0 e^{-j2\pi f m} X(f)$$

Al mismo tiempo vemos que la señal de salida en el tiempo es la señal de entrada retardada m y multiplicada por H_0 (aplicando propiedades de la TFSD).

$$y[n] = H_0 x[n - m]$$

En estas condiciones específicas, se verifica que la **señal de entrada no cambia de forma**.

Ausencia de cambio de forma

El hecho de que la señal de entrada no cambie de forma significa que todas las frecuencias que conforman la señal se retardan del mismo modo. Recordemos que esto significa que el retardo de grupo es constante y la fase es lineal.

Ejemplo

Veamos un caso particular. Si tomamos la señal discreta de entrada $x[n] = \sin(2\pi fn + \varphi)$ y la pasamos por el filtro $H(f)$ a la salida tenemos:

$$y[n] = |H(f)| \sin\left(2\pi fn + \varphi + \theta_h(f)\right) = |H(f)| \sin\left(2\pi f\left(n + \frac{\theta_h(f)}{2\pi f}\right) + \varphi\right)$$

donde $\frac{\theta_h(f)}{2\pi f}$ es el retardo de fase. Comprobemos que para que se verifique $|H(f)|$ debe ser constante y $\theta_h(f)$ debe ser lineal. La figura siguiente muestra un ejemplo de la influencia de un filtro en amplitud y retardo sobre una señal sinusoidal:

Seno a la entrada y salida del filtro

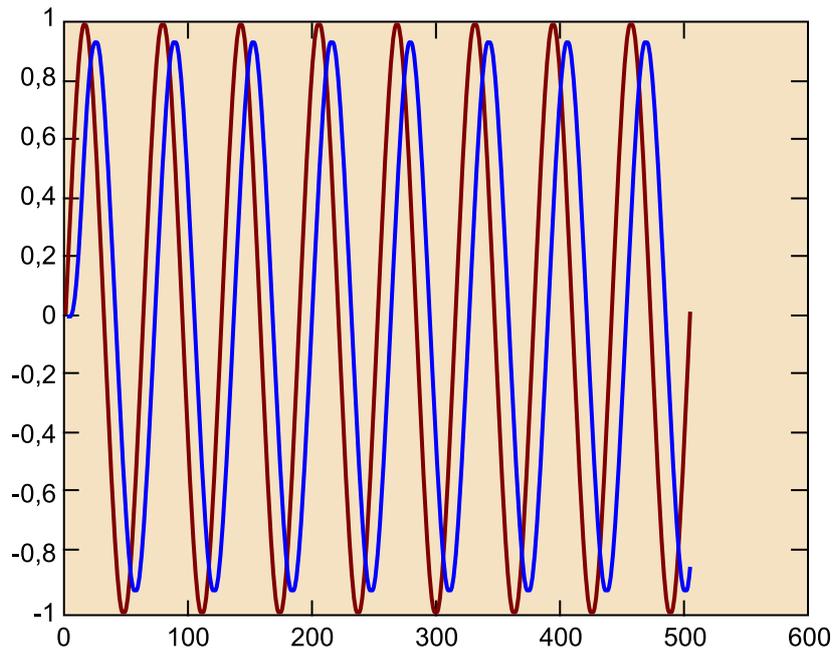


Figura 6. Seno a la entrada del filtro, es decir, antes de aplicar un filtro (rojo). Seno a la salida del filtro, es decir, después de aplicar el filtro (azul). La salida queda ligeramente atenuada y retardada. El eje X son muestras y el eje Y, amplitud.

Esta propiedad de mantener la señal de entrada sin distorsión (sin que la señal cambie de forma) nos puede interesar mucho en la práctica. Ahora bien, hemos de tener presente que para que un sistema causal con respuesta impulsional real y función de transferencia racional tenga **fase lineal**, la **respuesta impulsional debe ser simétrica o antisimétrica (respecto a su muestra central) y finita**.

2.2. Especificación de la entrada-salida de un filtro: ecuaciones en diferencias finitas

Una manera directa para describir la relación entrada-salida² de un filtro causal es mediante una **ecuación en diferencias finitas**.

Referencia bibliográfica

Podéis ver la demostración de esto en:

J. G. Proakis; D. G. Manolakis (2007). *Tratamiento digital de señales*. Madrid: Pearson Prentice Hall.

⁽²⁾Para ver gráficamente la relación entrada-salida de un filtro causal, podéis ver la figura 7 en este mismo apartado.

Esta ecuación es una fórmula para calcular la muestra n de salida, que depende de muestras de entrada presentes y pasadas (en el dominio del tiempo). Una ecuación genérica en diferencias finitas se puede escribir como sigue:

$$y[n] = b_0x[n] + b_1x[n-1] + \dots + b_Mx[n-M] - a_1y[n-1] - \dots - a_Ny[n-N]$$

que, compactada en sumatorios, se escribe:

$$y[n] = \sum_{i=0}^M b_i x[n-i] - \sum_{i=1}^N a_i y[n-i]$$

donde $x[n]$ es la señal discreta de entrada y $y[n]$ es la señal de salida, y las constantes b_i ($i=0,1,2,\dots, M$) y a_i ($i=1,2,\dots, N$) son el número de coeficientes del filtro.

Si analizamos estas ecuaciones en el dominio Z , obtendremos la función de transferencia:

$$y[n] = x[n] * h[n] \xrightarrow{TZ} Y(z) = X(z)H(z)$$

$$H(z) = \frac{Y(z)}{X(z)}$$

Sabemos que:

$$y[n] = \sum_{i=0}^M b_i x[n-i] - \sum_{i=1}^N a_i y[n-i] \xrightarrow{TZ} Y(z) = \sum_{n=0}^M b_n X(z)z^{-n} - \sum_{n=1}^N a_n Y(z)z^{-n}$$

Entonces:

$$Y(z)\left(1 + \sum_{n=1}^N a_n z^{-n}\right) = \sum_{n=0}^M b_n X(z)z^{-n}$$

Por lo tanto:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{n=0}^M b_n z^{-n}}{1 + \sum_{n=1}^N a_n z^{-n}}$$

La función de transferencia se puede reformular en términos de las raíces de los polinomios numerador y denominador de la manera siguiente:

Información complementaria

Encontraréis más información en lo referente a las ecuaciones en diferencias finitas en las páginas 120 a 124 del libro *Señales y sistemas*, de A. V. Oppenheim y A. S. Willsky.

Nota

Podéis mirar el vídeo "Raíces de un polinomio".

$$H(z) = \frac{\sum_{n=0}^M b_n z^{-n}}{1 + \sum_{n=1}^P a_n z^{-n}} = \frac{b_0 \prod_{n=0}^M (1 - z_k z^{-1})}{a_0 \prod_{k=1}^P (1 - p_k z^{-1})}$$

donde z_k son los ceros (porque cuando $z = z_k$, la función transferencia vale cero) y p_k son los polos (porque cuando $z = p_k$, la función transferencia es infinita). Recordemos que los polos se representan en el plano Z con una cruz y los ceros, con un círculo, tal como se ve representado en la figura siguiente:

Recordeu

El diagrama de polos y ceros lo obtenemos a partir del $H(z)$.

Ejemplo de diagrama de polos y ceros

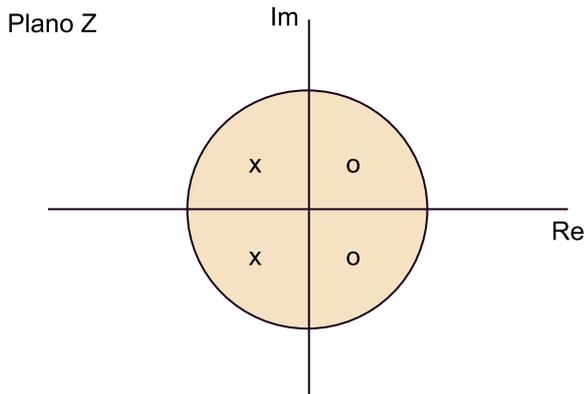


Figura 7. Ejemplo de diagrama de polos y ceros

Ved también

En cuanto a la transformada Z, podéis ver el apartado “Transformación del dominio temporal al dominio frecuencial” del módulo 1.

Condiciones que debe verificar una función de transferencia para que sea causal y estable

Analicemos qué condiciones debe verificar una función de transferencia para que sea causal y estable.

La estabilidad se garantiza si, y solo si, $|z| = 1$ pertenece a la región de convergencia (para sistemas LTI causales).

La causalidad se garantiza si, y solo si, $|z| > \max\{|p_k|\}$ pertenece a la región de convergencia y $H(z)$, expresada como una razón de polinomios en z , verifica que el orden del numerador no es mayor que el orden del denominador.

La única manera de verificar estos dos criterios es que el módulo del polo sea más pequeño que 1.

La definición de un filtro en ecuación en diferencias finitas nos permite hacer claramente una distinción entre dos tipos de filtros:

- **Filtros no recurrentes:** son aquellos en los que los coeficientes son cero y, por lo tanto, solo dependen de la señal de entrada. La respuesta impulsional del filtro tiene un número finito de muestras diferentes de cero y, por ello, reciben el nombre de **filtros FIR** (respuesta impulsional finita o *finite impulse response*).
- **Filtros recurrentes:** son aquellos en los que la respuesta impulsional tiene un número infinito de muestras diferentes de cero, y reciben el nombre de

filtros IIR (respuesta impulsional infinita o *infinite impulse response*). Esto se sabe porque los coeficientes a_k y b_k son diferentes de cero.

Finalmente, y como hemos visto en las definiciones del primer apartado de este módulo, el **orden** de un filtro se define como el máximo entre M y P (que son el orden del polinomio numerador y denominador, respectivamente).

Orden de un filtro

El orden de un filtro determina la complejidad de implementación del propio filtro. Cuanto mayor es el orden, más recursos computacionales se necesitan.

Ejemplo

Para ejemplificar algunos de los conceptos que hemos definido, utilizaremos un filtro FIR muy sencillo definido por la ecuación de entrada-salida siguiente:

$$y[n] = x[n] + x[n-1]$$

donde $x[n]$ es la señal discreta de entrada e $y[n]$ es la señal discreta de salida.

La señal de salida es el resultado de hacer la convolución entre la señal de entrada y la respuesta impulsional y se puede expresar como sigue:

$$y[n] = x[n] * h[n] = \sum_{i=0}^1 h[i]x[n-i] = h[0]x[n] + h[1]x[n-1]$$

Si aplicamos la transformada Z de la expresión anterior, obtendremos la función de transferencia del filtro:

$$Y(z) = H[0]X(z) + H[1]X(z)z^{-1}$$

$$\frac{Y(z)}{X(z)} = H[0] + H[1]z^{-1}$$

En este caso, los coeficientes de la respuesta impulsional son $h[0]=1$, $h[1]=1$. Dado que solo se utiliza un retardo de una muestra, el filtro es de orden 1. La respuesta impulsional tiene un número finito de muestras diferentes de cero; por lo tanto, podemos afirmar que se trata de un filtro FIR.

Este filtro se puede programar fácilmente en Matlab u Octave de la manera siguiente:

1. Definimos la señal de entrada: $x = [1,2,3,4,5,6,7,8,9,10]$.
2. Definimos los coeficientes de la respuesta impulsional: $h = [1,1]$.
3. Inicializamos la suma de convolución: $v = 0$.
4. Definimos el orden del filtro: $M = 1$.
5. Definimos la longitud de la señal de entrada: $N = 10$.
6. Escribimos el código de la suma de convolución:

```
for n = 1:N
y(n) = h*[x(n);v]; % producto escalar de dos vectores
      : vector fila*vector columna
v=[x(n)];
```

Obtenemos la señal de salida: $y = [1,3,5,7,9,11,13,15,17,19]$.

3. Filtros reales: plantilla de especificación de un filtro

Como hemos dicho, los filtros representados en la figura 1 son filtros **ideales** o **no realizables** puesto que la respuesta impulsional es no causal. A la hora de especificar las características, a un filtro **real** o **realizable** se permite una relajación de las especificaciones teóricas requeridas (esto lo denominamos *margen de tolerancia* o *relajación*). Por ello, en la práctica, antes de diseñar un filtro, lo primero que necesitamos es la **plantilla de especificaciones**³, que básicamente determina cómo queremos que sea el filtro y qué margen de tolerancia o relajación⁴ aceptamos.

⁽³⁾Dentro de las especificaciones encontramos frecuencia de corte y frecuencia de paso, entre otras.

⁽⁴⁾Por relajación de las especificaciones teóricas requeridas entendemos el hecho de permitir un margen de variación en las especificaciones.

El filtro **real** se diseña de modo que la función de transferencia corresponda a la siguiente ecuación:

$$H(Z) = \frac{\sum_{n=0}^M b_n z^{-n}}{1 + \sum_{n=1}^P a_n z^{-n}} = \frac{b_0 \prod_{n=0}^M (1 - z_k z^{-1})}{a_0 \prod_{k=1}^P (1 - p_k z^{-1})}$$

La figura 8 muestra una plantilla de especificación correspondiente a la función real $H(f)$ y la correspondiente atenuación de un filtro paso bajo:

Plantilla de especificación correspondiente al módulo (izquierda) y la atenuación (derecha) del filtro paso bajo

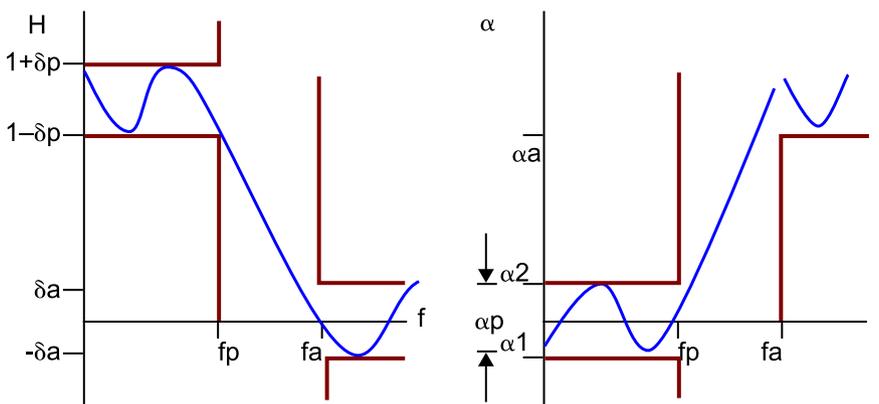


Figura 8. Plantilla de especificación correspondiente al módulo (izquierda) y la atenuación (derecha) del filtro paso bajo

Nomenclatura

- f_p : frecuencia de paso (hasta esta frecuencia el filtro deja pasar la señal).
- f_a : frecuencia de atenuación (a partir de esta frecuencia el filtro atenúa la señal).
- δ_p : tolerancia de paso en módulo, que es la relajación en módulo que permitimos en la banda de paso.
- δ_a : tolerancia de atenuación en módulo, que es la relajación en módulo que permitimos en la banda de atenuación.

- $\alpha_p = \alpha_2 - \alpha_1$: tolerancia de paso en atenuación, que es la relajación en atenuación que permitimos en la banda de paso.
- α_a : tolerancia de atenuación en atenuación, que es la relajación en atenuación que permitimos en la banda de atenuación.

Podemos definir la atenuación a partir de la función de transferencia:

$$\alpha(f) = -G(f) = -10 \log \frac{|H(f)|^2}{H_{ref}^2}$$

Observad que en la banda izquierda de la figura 8 se permite:

- Una tolerancia en torno al valor del módulo ($1 \pm \delta_p$).
- Una tolerancia sobre el valor de la banda atenuada (δ_a).
- Una banda de paso entre la frecuencia de paso y la frecuencia atenuada ($f_a - f_p$).

Observad que en la banda derecha de la figura 8 se permite:

- Una tolerancia en torno al valor de la banda de paso (α_p).
- Una tolerancia sobre el valor de la banda atenuada (α_a).
- Una banda de paso entre la banda de paso y la banda atenuada ($f_a - f_p$).

Cuanto más restrictivos seamos a la hora de definir las tolerancias, más costoso resultará implementar el filtro en cuestión.

Generalmente, el valor máximo de la función $H(f)$ es 1. A sabiendas de la relación entre la atenuación y la función de transferencia, encontraremos la relación entre las especificaciones de la atenuación y las especificaciones de la función de transferencia. Así, si tomamos $H_{max} = 1$, obtenemos que:

$$\alpha_p = \alpha_2 - \alpha_1 = 20 \log \frac{1 + \delta_p}{1 - \delta_p}; \quad \alpha_a = -20 \log \delta_a$$

$$\delta_p = \frac{10^{\frac{\alpha_p}{20}} - 1}{10^{\frac{\alpha_p}{20}} + 1}; \quad \delta_a = 10^{-\frac{\alpha_a}{20}}$$

Un filtro es un **sistema lineal e invariante**; por ello, se puede caracterizar con lo que conocemos como **respuesta impulsional**.

La respuesta impulsional relaciona la entrada y la salida de un sistema lineal e invariante en el tiempo.

Antes de diseñar un filtro, lo primero que necesitamos es la **plantilla de especificaciones**, que básicamente determina cómo queremos que sea el filtro: paso bajo, paso alto y la frecuencia de corte y de atenuación, entre otros parámetros.

4. Diseño de filtros digitales

En este apartado nos centraremos en el diseño de filtros digitales, que básicamente se subdividen en dos tipos: filtros FIR e IIR. Los primeros tienen una respuesta impulsional finita y los segundos se caracterizan por el hecho de tener una respuesta impulsional infinita. Dentro del subapartado de filtros IIR haremos un repaso muy breve del diseño de filtros analógicos que se necesitan para el diseño de filtros IIR.

4.1. Diseño de filtros FIR

Ya hemos visto que un filtro FIR realizable, además de ser lineal, invariante, causal y estable, se caracteriza por el hecho de tener una respuesta impulsional de longitud finita L , y la salida que tiene solo depende de valores de la entrada, nunca de la salida. En caso de querer tener fase lineal⁵, la respuesta impulsional debe ser simétrica o asimétrica.

La estructura directa genérica de un filtro FIR (conocida también como *estructura transversal*) se corresponde con la figura 9. Vemos que un filtro FIR solo usa muestras retardadas de la señal de entrada:

Estructura directa de un filtro FIR de orden M

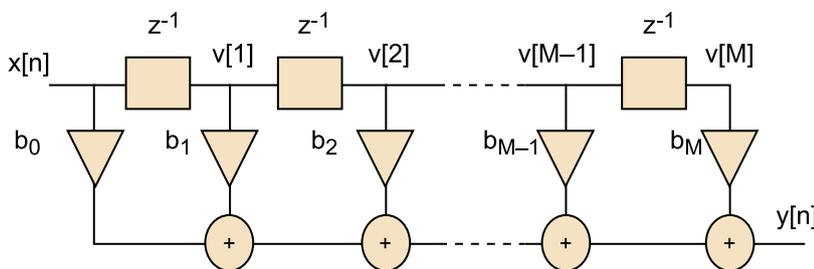


Figura 9. Estructura directa de un filtro FIR de orden M. Los triángulos se corresponden con multiplicadores y los elementos z^{-1} , con retardadores de una muestra. Observad que $v[i] = x[n - i]$.

Existen diferentes métodos para implementar filtros FIR. En esta asignatura veremos con detalle el método de ventanas y el método de Parks-McClellan. El primero se caracteriza por la simplicidad y el segundo, porque se utiliza frecuentemente en la práctica (de hecho, es el tipo de filtro que logra unas determinadas especificaciones con un orden más pequeño). Los dos están implementados en Matlab y Octave y veremos su utilización.

4.1.1. Método de ventanas

Una manera directa de hacer que la respuesta ideal sea finita es limitando una respuesta infinita. Así, si por ejemplo tenemos una respuesta impulsional ideal que, en el tiempo, es una función sinc (contiene infinitas muestras) y la multiplicamos por una función que tiene un número limitado de muestras

⁽⁵⁾Recordad que hemos dicho que un filtro de fase lineal significa que no distorsiona la forma de la onda de la señal original.

Lectura sugerida

Encontraréis explicado el concepto de simetría en la página 13 del libro *Señales y sistemas*, de A. V. Oppenheim y A. S. Willsky.

diferentes de cero, obtendremos una respuesta impulsional finita centrada en su origen para garantizar la simetría (observad que, para hacer el filtro causal, hemos de desplazar la ventana):

$$h[n] = h_i[n]w[n]$$

En esta ecuación, $h_i[n]$ es la respuesta impulsional del filtro ideal y $w[n]$ es la función que actúa de ventana (en inglés, *window*). Fijémonos en que en el dominio transformado esta multiplicación temporal es una convolución.

Otros tipos de ventanas que se utilizan en este método para limitar la función sinc que pertenece al filtro ideal son la triangular, Barlett, Blackman, Hanning, Hamming y Kaiser, y cada ventana tiene unas propiedades diferentes.

Matlab permite el uso de todas estas ventanas. Podéis ver cómo son con las órdenes siguientes:

```
L=64 % define la longitud de la ventana
wvtool(blackman(L)) % dibuja la ventana de Blackman en tiempo y en frecuencia
wvtool(hamming(L)) % dibuja la ventana de Hamming en tiempo y en frecuencia
wvtool(kaiser(L)) % dibuja la ventana de Kaiser en tiempo y en frecuencia
wvtool(kaiser(L),blackman(L),hamming(L)) % permite comparar diferentes tipos de ventanas
```

Ventanas simétricas

Observad que las ventanas son simétricas; esto significa que tenemos fase lineal y que el retardo de grupo es constante.

Respuesta en frecuencia de diferentes tipos de ventanas: Kaiser, Blackman y Hamming

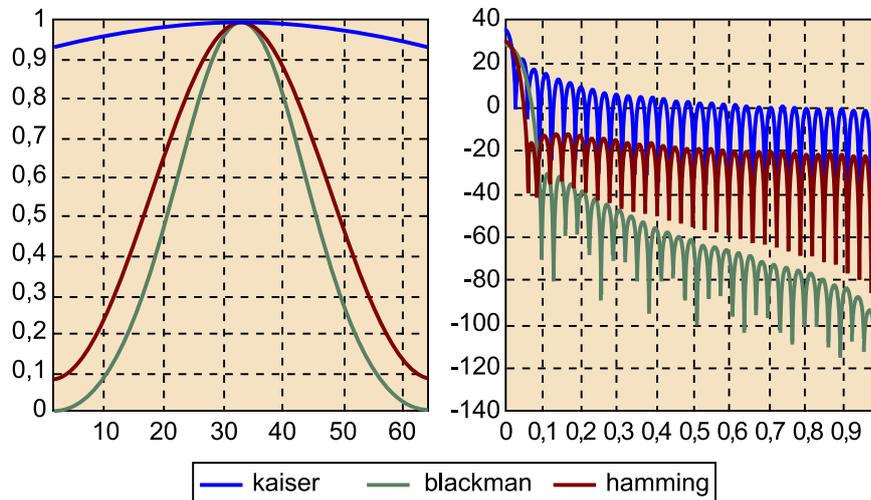


Figura 10. Respuesta en frecuencia de diferentes tipos de ventanas: Kaiser, Blackman y Hamming

Cada ventana presenta unas prestaciones diferentes. Por ejemplo, una ventana rectangular presenta una banda de paso más amplia, pero la atenuación en la banda eliminada es más pequeña. Tal como se ve en la figura 10, la amplitud del lóbulo principal es grande y la diferencia entre lóbulo principal y secundario, pequeña. Este comportamiento es totalmente opuesto en una ventana triangular. Así, pues, según las especificaciones de diseño, nos decidiremos por una ventana en concreto.

Comprobad los diferentes comportamientos de las ventanas siguiendo el ejemplo siguiente.

Así, a continuación ponemos un ejemplo concreto para ver cómo funciona este método. Utilizaremos la ventana rectangular, que es la más simple de las ventanas, y una ventana más elaborada, la de Hanning.

Ejemplo

En este ejemplo mostramos el filtrado (o, en este caso, el denominado *enventanado*) de una señal sinusoidal con un filtro de ventana rectangular y ventana de Hanning en el dominio temporal y en el dominio de la frecuencia calculado en Matlab:

```
n=512 % número de muestras de la fft
m=128 % número de muestras de la ventana
h = hanning(m);
sr = zeros(size(1:n));
sh = zeros(size(1:n));
f = pi / 3; % frecuencia de la señal sinusoidal

for i = 1:m ; sr(i) = sin(f * i) ; end
for i = 1:m ; sh(i) = h(i) * sin(f * i) ; end

plot(sr);
plot(sh);
```

A continuación mostramos el enventanado de la señal sinusoidal con un filtro de ventana rectangular y ventana de Hanning en el dominio temporal, respectivamente, en términos de amplitud y muestras temporales:

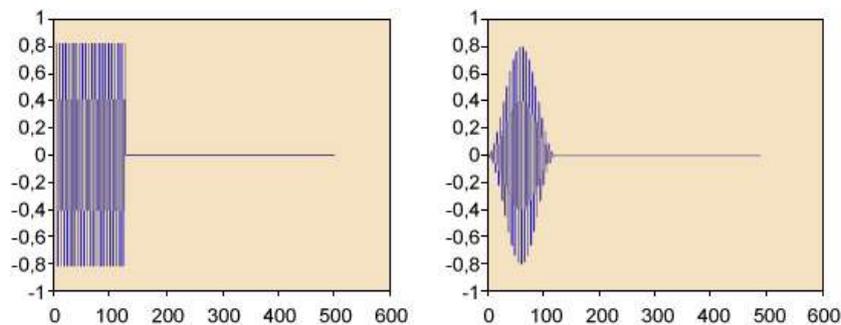


Figura 11. Enventanado en el dominio temporal de una señal sinusoidal con una ventana rectangular y una de Hanning, respectivamente. La figura muestra amplitud en términos de muestras temporales.

```
ar=abs(fft(sr));
ah=abs(fft(sh));

plot(ar(1:n/2))
plot(20 * log10(ar(1:n/2)))
plot(20 * log10(ah(1:n/2)))
```

A continuación mostramos el enventanado de la señal sinusoidal con un filtro de ventana rectangular y ventana de Hanning en el dominio frecuencial, respectivamente, en dB y muestras frecuenciales:

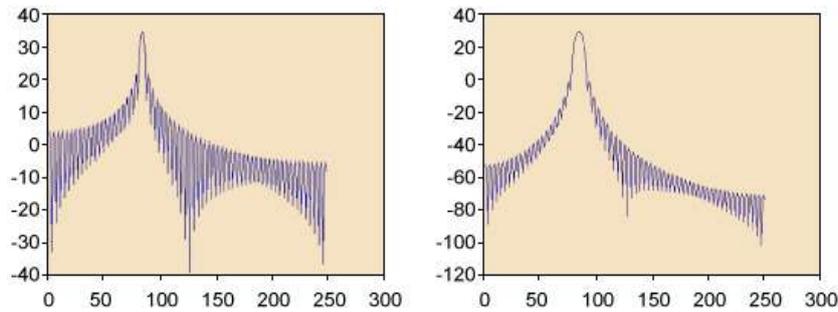


Figura 12. Enventanado en el dominio frecuencial de una señal sinusoidal con una ventana rectangular y una de Hanning, respectivamente. La figura muestra dB en términos de muestras frecuenciales.

A partir de estas imágenes, observad que la frecuencia central de los filtros es la misma. Por otro lado, vemos que la frecuencia de corte a 3 dB aproximadamente no coincide en los dos filtros. La frecuencia de corte a 3 dB es aquella frecuencia en la que la atenuación es mayor de 3 dB respecto a la atenuación mínima (la de la frecuencia central).

Finalmente, la amplitud del lóbulo principal y el máximo pico de los lóbulos laterales permite determinar la resolución y sensibilidad de un filtro, respectivamente. Denominamos *lóbulos* a los “picos” del filtro. El lóbulo principal es el pico del filtro donde la señal tiene menos atenuación. Los lóbulos laterales son todos los demás picos del filtro donde la señal tiene más atenuación.

El hecho de que la anchura del lóbulo principal sea más pequeña en el caso de la ventana rectangular que en la de Hanning quiere decir que hay una mayor resolución. Es decir, una discriminación más grande entre la frecuencia central y las frecuencias más cercanas a la central. Por otro lado, el hecho de que la relación lóbulo principal-lóbulo lateral sea más grande en el caso de la ventana de Hanning que en la rectangular provoca que esta ventana tenga más sensibilidad. Esto permite una mayor atenuación de las frecuencias no deseadas (que están a una distancia mayor de la que ocupa el lóbulo principal).

Con el método de ventanas tenemos una manera sencilla y entendible de elaborar filtros. Ahora bien, el principal inconveniente del método es que se observa un comportamiento oscilatorio entre la banda de paso y la banda atenuada y no permite controlar la separación entre las dos bandas.

Ejemplo

La siguiente figura muestra gráficamente la ecuación de construcción de un filtro por el método de ventanas (utilizando una ventana rectangular):

Referencia bibliográfica

Julián Quiroga Sepúlveda
(2009) *Diseño de filtros FIR: Método de ventanas. Procesamiento digital de señales. Pontificia Universidad Javieriana.*

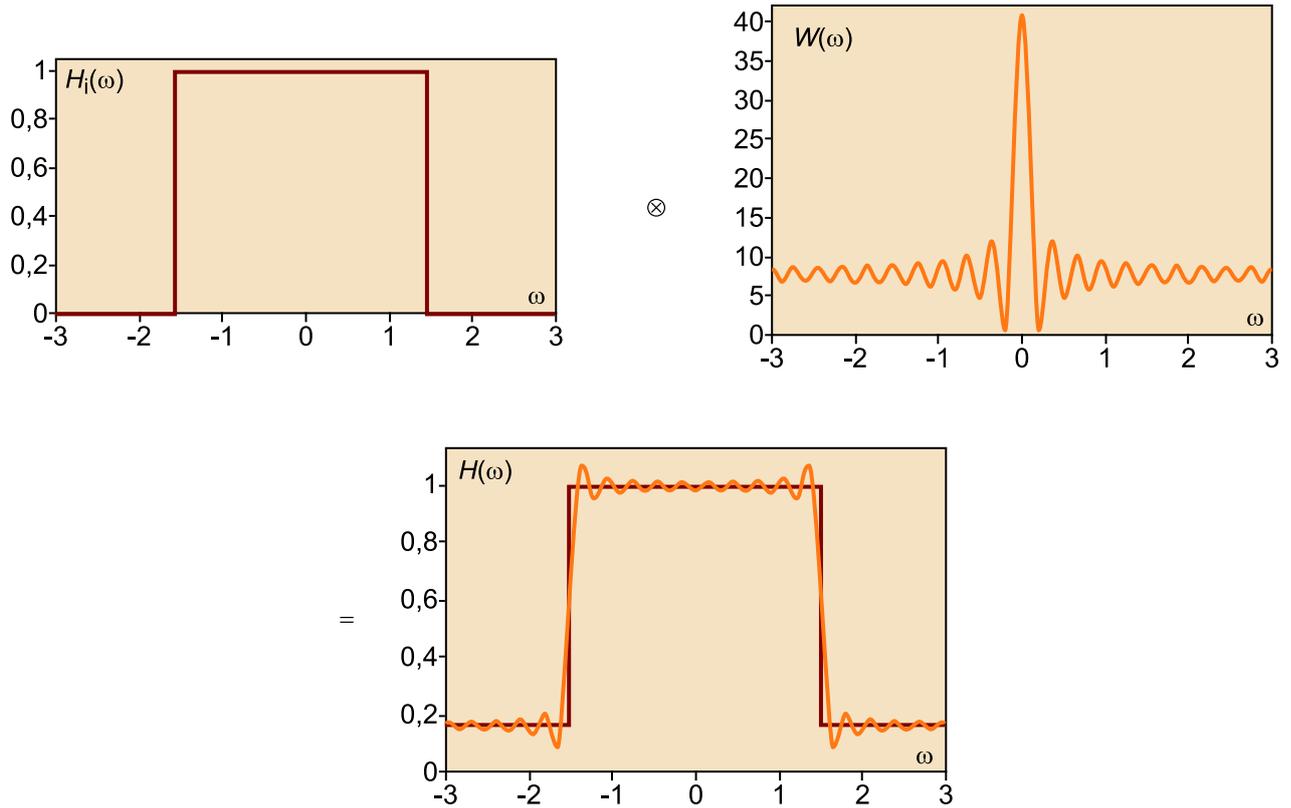


Figura 13. Construcción de un filtro FIR utilizando el método de ventanas

4.1.2. Método de Parks-McClellan

El algoritmo de Parks-McClellan se basa en aproximaciones polinómicas. Este método presenta un comportamiento de rizado constante⁶ en las bandas de paso y atenuada.

No entraremos en detalle en cómo se realizan los cálculos para obtener el filtro, puesto que para ello tenemos herramientas como la FDAtool de Matlab.

⁽⁶⁾Se denomina rizado cuando la forma de onda tiene picos y valles. Rizado constante quiere decir que los picos y los valles siempre tienen el mismo valor.

Referencia bibliográfica

Podrís profundizar sobre la FDAtool de Matlab en la página web de Mathworks.

FDAtool de Matlab

Esta herramienta proporciona una interfaz para diseñar y analizar filtros rápidamente. La FDAtool permite diseñar filtros FIR e IIR con determinadas especificaciones. Asimismo, la herramienta permite analizar las características de un filtro en concreto.

Para utilizar la FDAtool tenéis que entrar en Matlab y escribir *fdatool* en la línea de órdenes –es decir, a continuación del signo (>>)–. Después os aparecerá la ventana siguiente:

Lectura sugerida

Leed el artículo “Comparación de filtros FIR de fase lineal, por Método Óptimo y de Ventanas”.

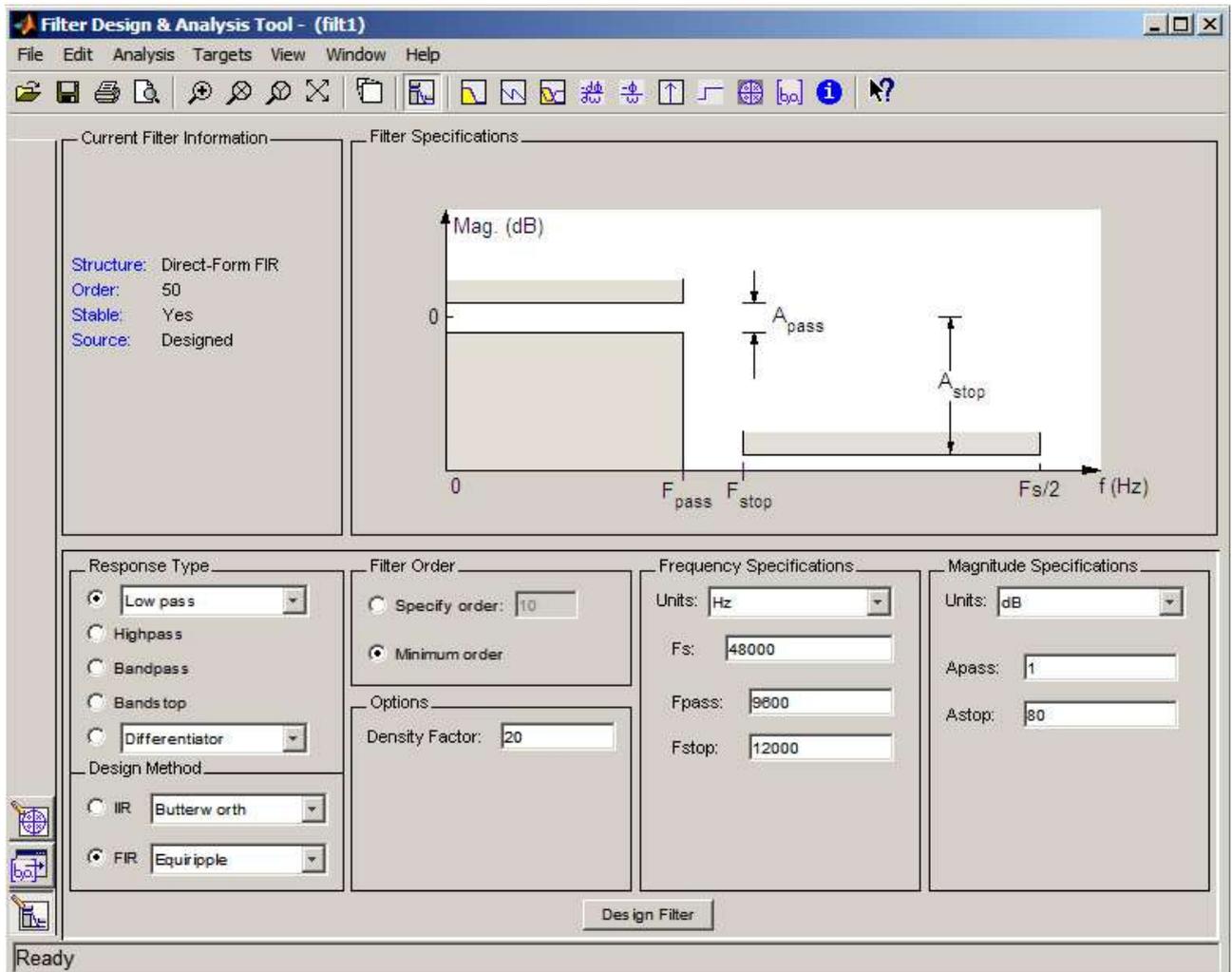


Figura 14. Herramienta FDAtool de Matlab

Diseñamos un filtro paso bajo FIR con las especificaciones siguientes:

- Método: Parks-McClellan (observad que Matlab lo denomina *Least-squares*, que es lo mismo que *equiripple*):
 - Design Method → FIR → Least-squares
 - Hay que tener seleccionado Response Type → Low pass
- Frecuencia de paso 8 kHz, frecuencia de atenuación a 6 dB (Fstop a fdatool) 8.8 kHz:
 - Frequency Specifications → Units → Hz
 - Entonces, Fpass 8000 y Fstop 8000

Si especificamos orden 5 (Order → Specify order --< 5), obtenemos el siguiente filtro:

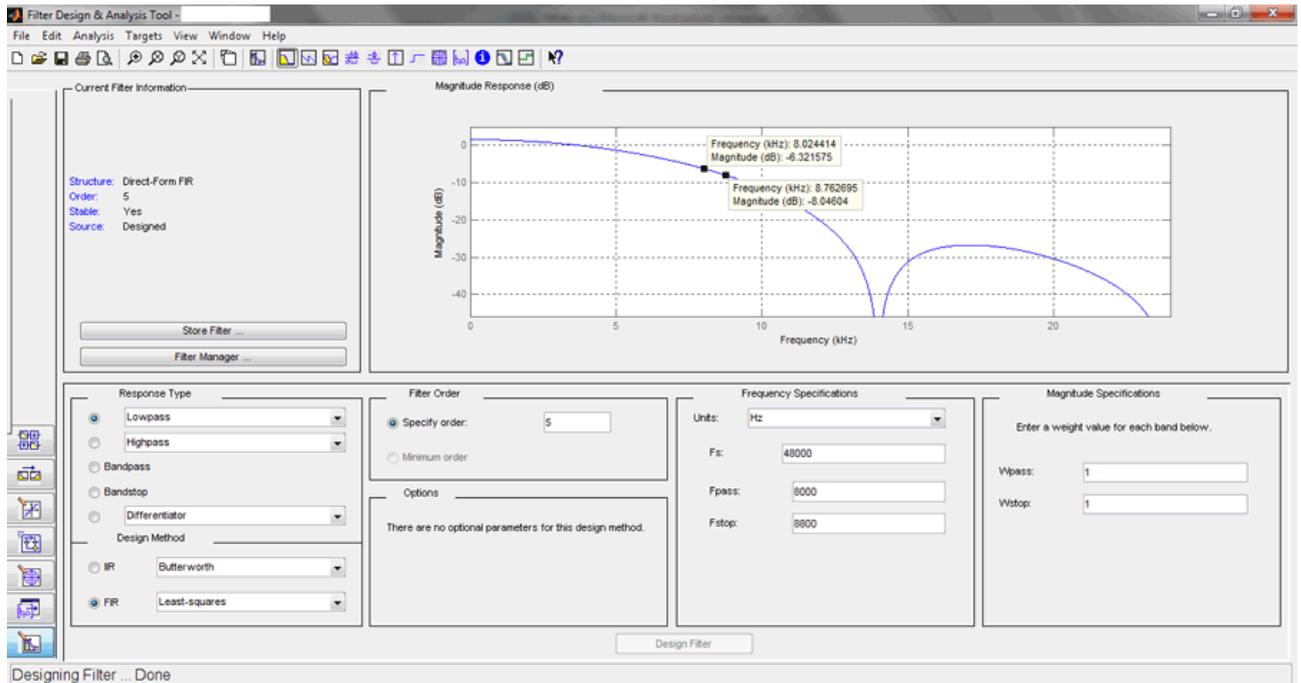


Figura 15. Pantalla de especificaciones

¿Consideráis que es el filtro que queremos?

No lo es, ya que las frecuencias de 8 kHz están atenuadas 6 dB; por lo tanto, tenemos que subir el orden del filtro.

Nuestras especificaciones son muy exigentes; por lo tanto, necesitamos órdenes más altos:

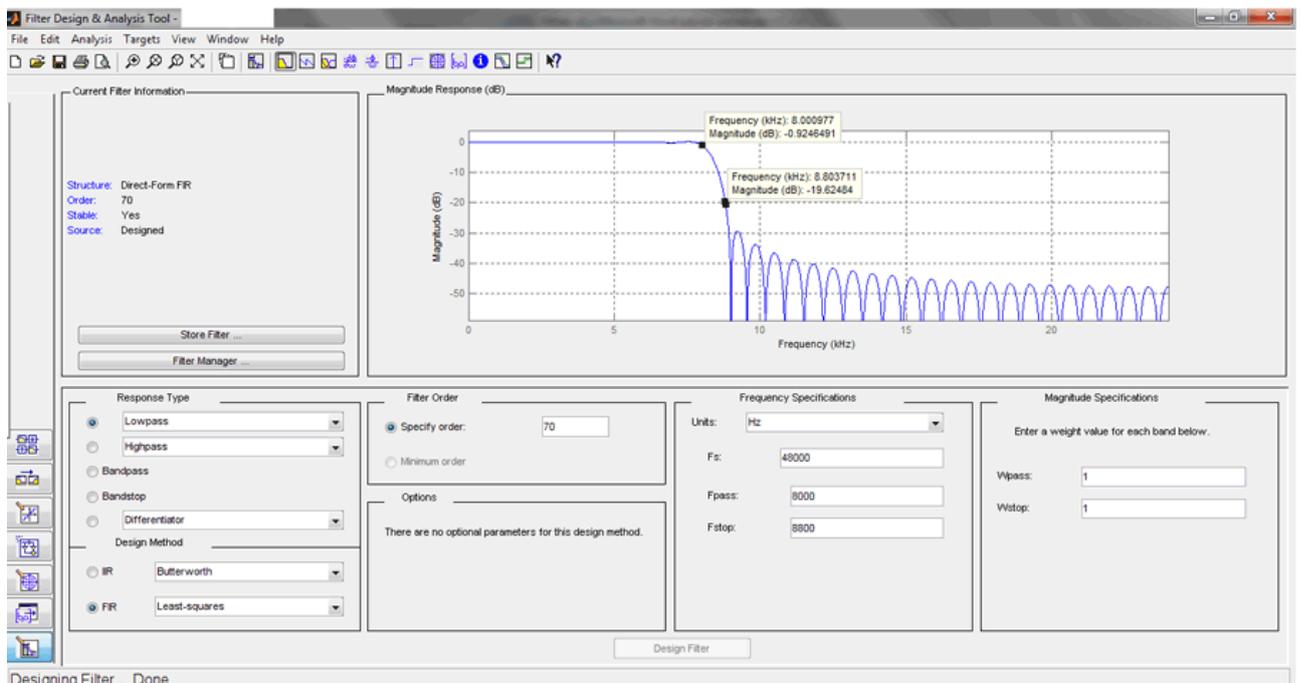


Figura 16. Filtro creado

Actividad

Con las mismas especificaciones que en el ejemplo anterior, comprobad qué orden necesitáis si en lugar de utilizar el método de Parks-McClellan (especificación 1 del ejemplo) utilizáis el método de ventanas con una ventana de Causer.

Un **filtro FIR** tiene una respuesta impulsional de longitud finita. Asimismo, la respuesta en frecuencia tiene fase lineal siempre que se cumplan las condiciones de simetría. El hecho de que la **fase sea lineal** implica que el filtro **no distorsione la salida a la banda de paso**. Esta característica de no-distorsión suele ser el principal motivo para utilizar filtros FIR.

4.2. Diseño de filtros IIR

Los **filtros IIR** se diferencian de los filtros FIR en que tienen una respuesta impulsional infinita y la señal de salida filtrada depende de valores anteriores de la entrada y de la salida. Esto quiere decir que la respuesta frecuencial no es de fase lineal porque no se cumplen las condiciones de simetría.

La principal ventaja de estos filtros es que se pueden lograr unas determinadas especificaciones de atenuación y frecuencias de paso o atenuadas con un orden inferior al orden que requiere un filtro FIR. La estructura directa de un filtro IIR se representa en la figura siguiente:

Estructura directa de un filtro IIR de orden $M = \max(P, Q)$

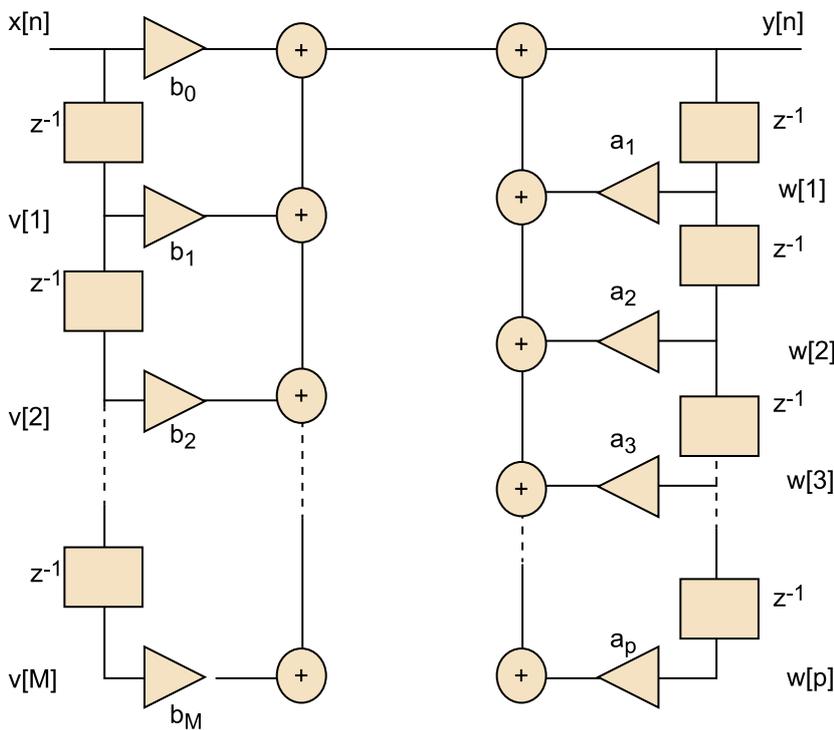


Figura 17. Estructura directa de un filtro IIR de orden $M = \max(P, Q)$. Observad que $v[i] = x[n-i]$ i $w[i] = y[n-i]$.

En el diseño de filtros IIR se determinan los coeficientes a_k y b_k de manera que verifiquen unas determinadas especificaciones. Ahora bien, no hay una teoría específica para diseñar filtros digitales IIR, sino que se utiliza la teoría de filtros analógicos. El motivo principal por el que se utiliza la teoría de filtros analógicos en digital es porque la teoría de filtros analógicos está muy desarrollada, estudiada y experimentada. Por lo tanto, vale la pena aprovecharla para filtros digitales.

La teoría de filtros analógicos para el diseño de filtros digitales se basa en realizar una transformación de variable. Hay diferentes procedimientos para hacer esta transformación. En estos apuntes revisaremos el método por transformación bilineal. Una vez efectuada la transformación bilineal, ya se pueden utilizar las aproximaciones analógicas clásicas, como Butterworth, Txebyxev, Inverso de Txebyxev o Cauer.

En resumen, para diseñar un filtro IIR digital lo que hacemos es diseñarlo en analógico y transformar este filtro de analógico a digital.

4.2.1. Transformación bilineal

La transformación se aplica a la función de transferencia de un filtro analógico $H_a(s)$ y el resultado que se obtiene es la función de transferencia de un filtro discreto $H(z)$.

Lectura recomendada

Encontraréis información sobre la transformada de Laplace en la página 654 y siguientes del libro *Señales y sistemas*, de A. V. Oppenheim y A. S. Willsky.

Recordad que hemos visto la transformada Z en el apartado “Transformación del dominio temporal al dominio frecuencial” del módulo 1.

Esta transformación ha de verificar las siguientes condiciones:

1. Garantiza una relación punto a punto entre el plano z y el plano s .
2. Para asociar la frecuencia entre los dos planos, la circunferencia de radio unidad del plano z se transforma en el eje imaginario del plano s .
3. Para asegurar la estabilidad, todo punto situado en el interior de la circunferencia de radio unidad en z se relaciona con un punto del semiplano izquierdo de s .

La transformación bilineal cumple estos requisitos:

$$s = \frac{1-z^{-1}}{1+z^{-1}}$$

Ved también

Veremos brevemente la teoría de los filtros analógicos en el apartado “Diseño de filtros analógicos: breve resumen” de este mismo módulo.

Ved también

Recordad el apartado “Conversión A/D y D/A: entorno analógico y entorno digital” del módulo 1.

Nota

Los filtros analógicos son estables si, y solo si, los polos de la función de transferencia tienen parte real negativa, como veremos a la próxima sección.

Ejemplo

Verificamos que un punto z situado sobre la circunferencia de radio unidad tiene como imagen en s un punto con parte real nula (por lo tanto, se sitúa sobre el eje imaginario):

$$z = e^{j\omega} \rightarrow s = \frac{1 - e^{-j\omega}}{1 + e^{-j\omega}} = j \tan \frac{\omega}{2} = j\Omega$$

De aquí podemos extraer que la pulsación del dominio analógico (Ω) queda relacionada con la pulsación (ω) del dominio discreto por la expresión siguiente:

$$\tan \frac{\omega}{2} = \Omega$$

Esta relación se utiliza para adaptar las especificaciones digitales, tal como veremos a continuación.

Actividad

Comprobad que un punto cualquiera $z_a = re^{j\omega a}$ con módulo $r < 1$ verifica la condición 3.

Una vez definida la transformación bilineal, ya podemos diseñar el filtro IIR.

4.2.2. Diseño de filtros analógicos: breve resumen

Un filtro analógico se puede definir como un sistema que manipula una señal continua (es decir, analógica) de manera que deja pasar un determinado intervalo de frecuencias y atenúa otras.

En muchos aspectos presentan muchas analogías con los filtros digitales. En particular, los filtros LTI analógicos se caracterizan por la respuesta impulsional que tienen (en este caso es continua en el tiempo). En lugar de utilizar ecuaciones en diferencias finitas, los filtros analógicos relacionan su entrada-salida con una ecuación diferencial. En lugar de utilizar la transformada Z para calcular la función de transferencia, se emplea la transformada de Laplace.

Los filtros analógicos mantienen las propiedades que hemos visto en la sección 1.3. Ya sabemos que para que una función de transferencia caracterice un sistema, este sistema tiene que ser LTI. La función de transferencia para un filtro analógico es el cociente entre la transformada de Laplace de la señal de salida y la transformada de Laplace de la señal de entrada:

$$H(s) = \frac{Y(s)}{X(s)}$$

Como ya hemos dicho, la relación de entrada-salida utiliza ecuaciones diferenciales:

$$a_n \frac{d^n y(t)}{dt^n} + a_{n-1} \frac{d^{n-1} y(t)}{dt^{n-1}} + \dots + a_0 y(t) = b_m \frac{d^m x(t)}{dt^m} + \dots + b_0 x(t)$$

$$a_n(s)^n Y(s) + \dots + a_0 Y(s) = b_m(s)^m X(s) + \dots + b_0 X(s)$$

Lectura recomendada

Encontraréis información sobre la transformada de Laplace en la página 654 y siguientes del libro *Señales y sistemas*, de A. V. Oppenheim y A. S. Willsky.

$$H(s) = \frac{Y(s)}{X(s)} = \frac{\sum_{l=0}^m b_l s^l}{\sum_{i=0}^n a_i s^i}$$

Si hacemos la transformada inversa de $H(s)$, necesitamos utilizar la transformada de Laplace siguiente:

$$\frac{k_i}{s - p_i} \xrightarrow{\mathcal{L}^{-1}} k_i e^{p_i t} u(t), \text{ donde } k_i, p_i \in \mathbb{C}$$

$$h(t) = \sum k_i e^{(\sigma_i + j\omega_i)t} u(t)$$

Veamos cuáles son los requisitos para que el filtro analógico sea causal y estable:

- La causalidad se verifica porque los valores $t < 0$ son 0 debido a $u(t)$ (la función escalón es cero para valores inferiores a 0, y 1 para valores superiores a 0).
- La estabilidad se verifica si:

$$\sum_{t=0}^{\infty} |h(t)| < \infty$$

Por lo tanto, hemos de garantizar que $\sigma_i < 0$. Además, observad que para que el filtro sea real, las exponenciales complejas deben aparecer en polos conjugados (de este modo las partes imaginarias se anulan entre sí).

A continuación veremos algunos parámetros que son útiles a la hora de diseñar un filtro analógico:

- Constante de selectividad (es decir, el cociente, la frecuencia de atenuación y la frecuencia de paso).
- Relación atenuación y función frecuencial (ahora utilizaremos F como frecuencia analógica):

$$\alpha(F) = -10 \log \frac{|H(F)|^2}{H_{max}^2}$$

Las plantillas de atenuación de los filtros se definen exactamente como en el apartado 1.3. La figura siguiente muestra unos ejemplos de los filtros analógicos principales (para una frecuencia de paso de 0,5):

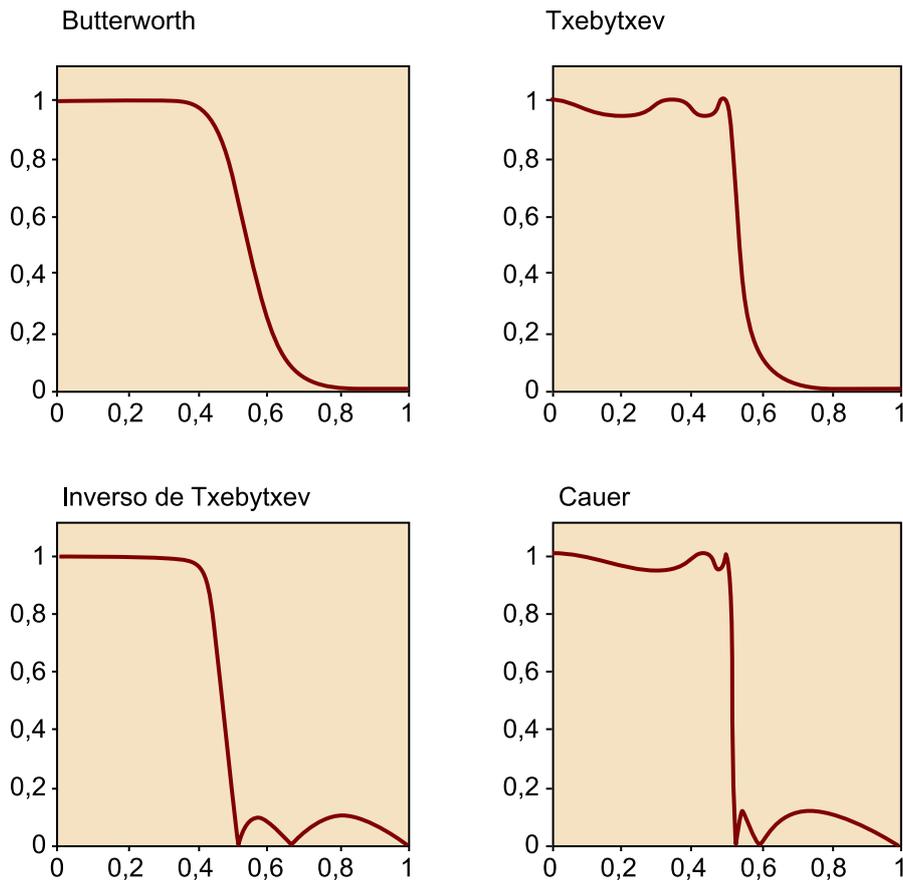


Figura 18. Función de transferencia de diferentes tipos de filtros analógicos paso bajo: Butterworth, Tchebychev, inverso de Tchebychev y Cauer

Las respuestas más comunes de filtros analógicos paso bajo son las que se muestran en la figura. Butterworth asegura una respuesta máximamente plana en la banda de paso y una tasa adecuada de atenuación en la banda atenuada. Por ello, el filtro de Butterworth es adecuado para aplicaciones dentro del procesamiento de audio.

El filtro de Tchebychev da una atenuación mucho más pronunciada, pero la amplitud no constante de la banda de paso lo hace inadecuado para los sistemas de audio.

El filtro de tipo elíptico o de Cauer es un filtro complejo pero que resulta muy adecuado en la práctica porque, dadas unas especificaciones, las cumple con el mínimo orden posible. Asimismo, si queremos una respuesta plana en la banda de paso, se puede obtener aumentando ligeramente el orden mínimo.

Ahora bien, si pensamos en aplicar filtros analógicos dentro de entornos digitales, se debe tener en cuenta que la transformación bilineal provoca una fuerte distorsión de fase.

4.2.3. Transformación bilineal y teoría de filtros analógicos

Para hacer el diseño de un filtro IIR seguimos el procedimiento siguiente:

- Debemos obtener las especificaciones del filtro analógico a partir de las especificaciones del filtro digital IIR.
 - Los límites de atenuación α_a y α_p se mantienen.
 - Transformamos los límites frecuenciales f_p y f_a en Ω_p y Ω_a mediante la relación que hemos obtenido en (ec. 3).
- A partir de las especificaciones analógicas, diseñamos el filtro analógico utilizando alguna técnica analógica (es decir, Butterworth, Txebyxev, inverso de Txebyxev o Cauer).
- Teniendo $H_a(s)$ y aplicando la transformación bilineal, obtenemos la función de transferencia del filtro digital IIR $H(z)$:

$$H(z) = H_a(s) \Big|_{s=\frac{1-z^{-1}}{1+z^{-1}}}$$

Así, si tenemos un filtro pao bajo, como por ejemplo $H_a(s) = \frac{1}{1+s}$, el filtro digital después de aplicar la transformación bilineal sería el siguiente:

$$H(z) = H_a(s) \Big|_{s=\frac{1-z^{-1}}{1+z^{-1}}} = \frac{1}{1+s} \Big|_{s=\frac{1-z^{-1}}{1+z^{-1}}} = \frac{1}{1+\frac{1-z^{-1}}{1+z^{-1}}} = 1 + \frac{1}{z}$$

Ejemplo

Extraídos de Internet, encontramos ejemplos de filtros analógicos hechos en Matlab como los que encontraréis en el anexo:

butter_ana_ex.m

txeby_ana_ex.m

Nota

La FDATool de Matlab realiza la transformación bilineal internamente, de modo que el diseñador no la tiene que hacer explícitamente.

Un **filtro IIR** presenta una respuesta impulsional infinita, lo cual implica que no tienen una fase lineal y pueden ser estables o inestables.

5. De la teoría a la práctica

En este apartado desarrollaremos un par de conceptos prácticos de filtros. Por un lado, veremos criterios para elegir un filtro FIR o un IIR. Por otro lado, veremos cómo hemos de diseñar un filtro desde cero.

5.1. La elección de un filtro FIR o IIR. ¿Cuándo debemos utilizar uno u otro?

Después de ver los principales aspectos teóricos de los filtros, ahora nos centraremos en aspectos más prácticos.

Respecto a las **ventajas** de utilizar este tipo de filtros, vemos que los filtros FIR pueden tener fase lineal, lo que permite mantener la forma de la señal. Ahora bien, en procesamiento de audio, en general, no es necesario tener una fase lineal o mantener la señal de onda.

Asimismo, los filtros FIR son siempre estables.

Por otro lado, los filtros IIR suelen ser de orden pequeño (por lo tanto, menos coste computacional).

En cuanto a los **inconvenientes** principales, vemos que los filtros FIR son de orden grande y los filtros IIR son de fase no lineal, y se debe controlar la inestabilidad, porque si son inestables no son realizables.

Hay que tener en cuenta que en un filtrado audio, cuando está bien hecho, no tenemos por qué percibir si hemos filtrado con un filtro FIR o IIR.

Finalmente, **a la hora de elegir un filtro FIR o IIR en la práctica, preferentemente utilizaremos el filtro de Parks-McClellan si queremos un filtrado FIR, y el filtro de Cauer si queremos un filtrado IIR.** Las otras aproximaciones, ventanas en el caso FIR y Butterworth o Txebyxev en el caso IIR, requieren un orden mayor para cumplir las mismas especificaciones, tal como hemos visto con la herramienta FDAtool de Matlab.

Nota

El oído humano, genéricamente hablando, no es extremadamente sensible a la distorsión de fase. En cambio, la fase es más importante en imagen para la sensibilidad del ojo.

5.2. Caso práctico: diseño de un filtro desde cero con Audacity y con Matlab

5.2.1. Caso 1. Filtrado de una señal de audio utilizando Audacity

Audacity es un programa libre y de código abierto que permite grabar y editar sonidos. En esta asignatura lo utilizaremos en diferentes módulos.

Así, como ejemplo de aplicación abriremos Audacity y elegiremos una canción que tengamos en formato .MP3, por ejemplo. Para ello, haced clic en *Abrir* y buscad la canción que os apetezca editar. Una vez abierta, seleccionad con el ratón el fragmento que queréis editar. Después, haced clic en *Efecto* y a *FiltroFFT*. Especificad el filtro que queréis probar y haced clic en *Aplica*.

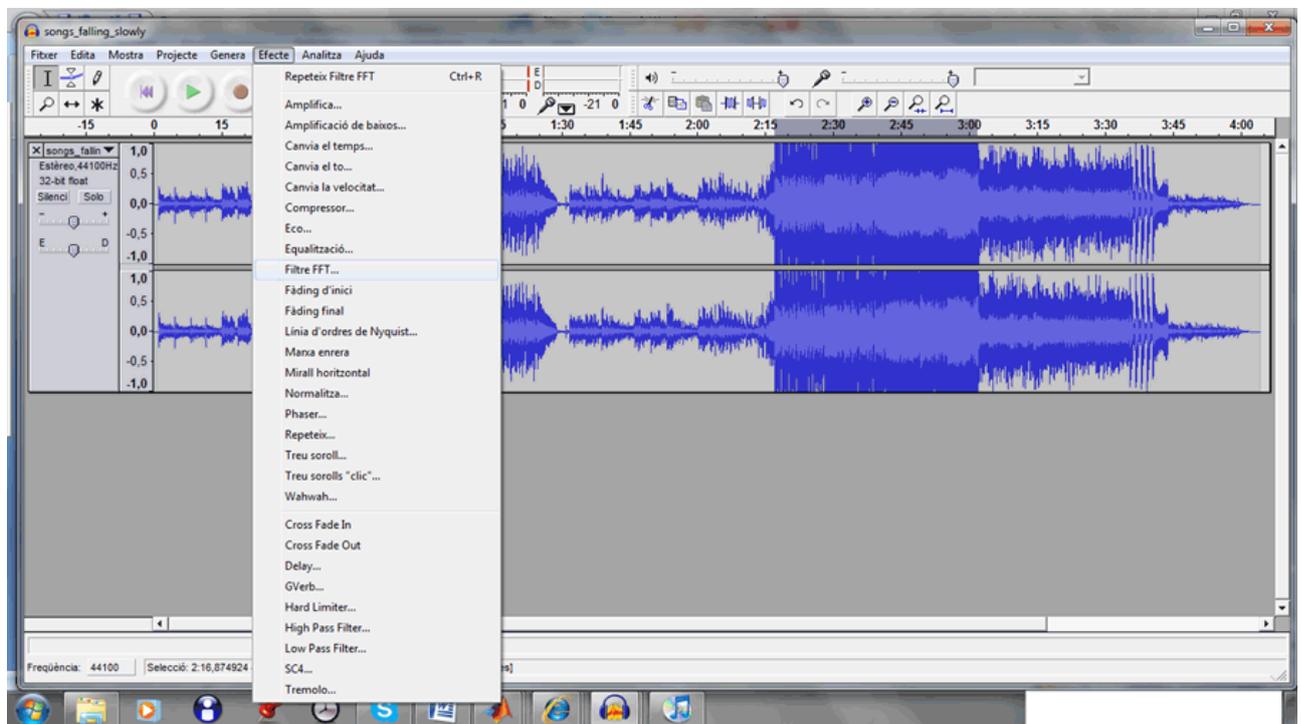


Figura 19. Visualización de Audacity haciendo filtrado

5.2.2. Caso 2. Filtrado de una señal de audio utilizando Matlab

Matlab permite jugar con más parámetros que Audacity.

Para empezar, leeremos el fichero de sonido en formato .wav, que es el formato de sonido que lee Matlab por defecto.

```
[Y, Fs]=wavread(&#8216;file.wav&#8217;);
```

Lectura de la línea de código

Y es el sonido codificado de manera digital, con frecuencia de muestreo F_s .

El fichero *file.wav* puede ser cualquiera que tengáis en vuestro ordenador.

Si queremos comprobar que el sonido se ha codificado correctamente, lo tenemos que escuchar de la manera siguiente:

```
sound(Y, Fs)
```

Ahora diseñamos un filtro como queramos, utilizando la herramienta FDA-tool. En este punto se trata que penséis qué especificaciones queréis. Por poner un ejemplo, nosotros definiremos el filtro siguiente:

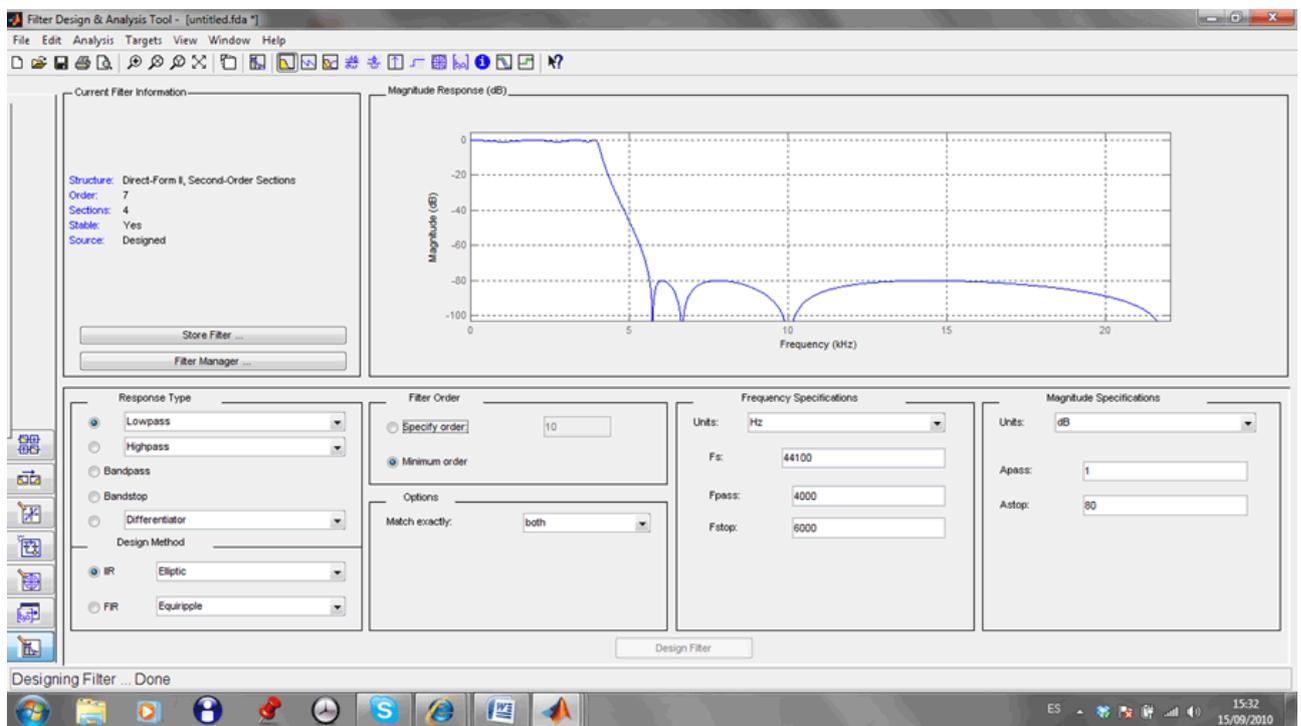


Figura 20. Filtrado con Matlab utilizando la FDAtool

A continuación, hacemos *FileGenerate M-File* y lo ponemos en la línea de órdenes.

```
Fs = 44100;      % Sampling Frequency
Fpass = 4000;   % Passband Frequency
Fstop = 6000;  % Stopband Frequency
Apass = 1;     % Passband Ripple (dB)
Astop = 80;    % Stopband Attenuation (dB)
match = 'both'; % Band to match exactly
% Construct an FDESIGN object and call its ELLIP method
h = fdesign.lowpass(Fpass, Fstop, Apass, Astop, Fs);
Hd = design(h, 'ellip', 'MatchExactly', match);
```

Para utilizar este filtro lo que hacemos es:

```
X=filter(Hd,Y)
```

Escuchamos la nueva señal filtrada:

```
sound(X, 44100)
```

Anexo

Filtro de Butterworth: butter_ana_ex.m

Dirección web

Podéis consultar la página de la Escuela de Ingenieros de la Universidad de Navarra (TECNUM)

```
%% Ejemplo de filtro paso bajo de Butterworth analógico
clear all;

fp=20e3;wp= 2*pi*fp;
Ap=0.5;
fs=90e3;ws= 2*pi*fs;
As=50;

% Cálculo de e2
e2 = 10^(0.1*Ap)-1;

% Orden del filtro
n = log(sqrt((10^(0.1*As)-1)/e2))/log(fs/fp);
n = ceil(n);

% Frecuencia de corte
wc = wp/(e2^(1/(2*n)));

% Parámetros para calcular los polos del filtro
k=1:n;
alpha = (2*k-1)*pi/(2*n);

% Determinación de los polos en el prototipo
pk = -sin(alpha) +j*cos(alpha);

[B,A]=zp2tf([],pk,1);

[B1,A1] = lp2lp(B,A,wc);

N=500;
wf=2*pi*100e3;

W=0:wf/N:wf;
H=freqs(B1,A1,W);

subplot(1,2,1); plot(W/(2*pi),20*log10(abs(H)));grid;
ylabel(['|H(w)|']);
xlabel(['Frecuencia (Hz)']);
subplot(1,2,2); plot(W/(2*pi),20*log10(abs(H)));grid;
```

```

ylabel(['|H(w)|']);
xlabel(['Frecuencia (Hz)']);
axis([0 1.1*fp -Ap-0.1 0.1]);
print -depsc2 butter_ej
pause;close;

% Retardo de grupo

dW=W(2)-W(1);
phase=unwrap(angle(H));
dphase = -diff(phase)/dW;

plot(W(1:length(W)-1)/(2*pi),dphase);grid;
ylabel(['Retardo de grupo (s)']);
xlabel(['Frecuencia (Hz)']);
print -depsc2 butter_groupdelay_ej
pause;close;

```

Txebychef: txeby_ana_ex.m

```

%% Ejemplo de filtro paso bajo de Txebyxev-I analógico

clear all;

fp=20e3;wp=2*pi*fp;
Ap=0.5;
fs=90e3;ws=2*pi*fs;
As=50;

% Cálculo de e2
e2 = 10^(0.1*Ap)-1;

% Orden del filtro
n = acosh(sqrt((10^(0.1*As)-1)/e2))/acosh(fs/wp);
n = ceil(n);

% Parámetros para calcular los polos y ceros del filtro

theta = (1/n)*asinh(1/sqrt(e2));
k=1:n;
alpha = (2*k-1)*pi/(2*n);

% Determinación de los polos en el prototipo
pk = -sinh(theta)*sin(alpha) +j*cosh(theta)*cos(alpha);

[B,A]=zp2tf([],pk,1);

```

```

% Ajustar ganancia para que |H(0)|=1
if rem(n,2)==0
    K=A(length(A))*(1/sqrt(1+e2));
else
    K=A(length(A));
end

[B1,A1] = lp2lp(K*B,A,wp);

N=500;
wf=2*pi*100e3;

W=0:wf/N:wf;
H=freqs(B1,A1,W);

subplot(1,2,1); plot(W/(2*pi),20*log10(abs(H)));grid;
ylabel(['|H(w)|']);
xlabel(['Frecuencia (Hz)']);
subplot(1,2,2); plot(W/(2*pi),20*log10(abs(H)));grid;
ylabel(['|H(w)|']);
xlabel(['Frecuencia (Hz)']);
axis([0 1*fp -Ap-0.1 0.1]);
print -depsc2 chebl_ej
pause;close;

%% Calcular el retardo de grupo
dW=W(2)-W(1);
phase=unwrap(angle(H));
dphase = -diff(phase)/dW;

plot(W(1:length(W)-1)/(2*pi),dphase);grid;
ylabel(['Retardo de grupo (s)']);
xlabel(['Frecuencia (Hz)']);
print -depsc2 chebl_groupdelay_ej
pause;close;
plot(pk,'x');axis([-1.1 0 -1.1 1.1]);axis equal;

```

Código de la función de Matlab easyspec.m

```

function [s,f]=easyspec(x,fs)
%EASYSPEC Easy plot of spectrum estimate
%
% S=EASYSPEC(X) will return a spectrum vector of
% X. [S,F]=EASYSPEC(X,FS) will also return a frequency
% axis in F, while the sample frequency is given in FS.
%
% EASYSPEC(X) and EASYSPEC(X,FS), will plot the spectrum

```

```
% and return the S vector in Matlab's "ans".
%
% Notes:
%
% 1. Spectrum values are in dB.
%
% 2. The frequency domain goes from DC to the Nyquist
%    frequency. The "mirror part" of the spectrum is omitted.
%
% 3. The sample segments, from which the spectrum is
%    calculated are picked by random. The function might
%    return significantly different results each time it's
%    called to if X isn't stationary.
%
% 4. EASYSPEC uses a hanning window and zero-pads by a
%    factor of 4. The spectrum vector will look smooth.

% Eli Billauer, 17.1.01 (Explicitly not copyrighted).
% This function is released to the public domain; Any use is allowed.

if nargin==0
    error('No input vector given');
end

if (nargin==1)
    fs=2;
end

NFFT=16384; NWIN=NFFT/4;
LOOP=100;
win=hanning(NWIN)';

x=x(:) *(17.127/NFFT/sqrt(LOOP));
n=length(x);
maxshift=n-NWIN;

if (n<2*NWIN)
    error(['Input vector should be at least of length '...
        num2str(2*NWIN)]);
end

s=zeros(1,NFFT);

for i=1:LOOP
    zuz=floor(rand*maxshift);
    s=s+abs(fft([win.*x(1+zuz:NWIN+zuz) zeros(1,NFFT-NWIN)]).^2);
end

s=10*log10(s(1:NFFT/2));
f=linspace(0,fs/2,NFFT/2);
```

```
if nargin==0
    hold off;
    plot(f,s);
    ylabel('Espectro');
    xlabel('Frecuencia');
    grid on; zoom on;
end
```

