

**UNIVERSIDAD AUSTRAL DE CHILE
FACULTAD DE CIENCIAS DE LA INGENIERÍA
ESCUELA DE INGENIERÍA ACÚSTICA**

*Profesor Patrocinante
Pedro Reumay Romero
Instituto de Matemática*

*Profesor Copatrocinante
Jorge Arenas Bermúdez
Instituto de Acústica*

**CONSTRUCCIÓN DE FILTROS DIGITALES
MEDIANTE WAVELETS**

*Tesis presentada como parte de los requisitos
para optar al grado de Licenciado en Acústica y
al Título profesional de Ingeniero Acústico*

**MARIO ALEJANDRO GONZÁLEZ MONTENEGRO
VALDIVIA-CHILE
2004**

*Ser consciente de su propia ignorancia
es un gran paso hacia el saber*

AGRADECIMIENTOS

Gracias padre celestial, por haberme dado la vida, innumerables son las bendiciones que me has entregado cada día y has sido fuente de inspiración para realizar este sueño.

Mario y Raquel: faltan las palabras para agradecer lo que han entregado todos estos años. En las buenas y en las malas, jamás me ha faltado su cariño y apoyo. Gracias papas.

A la abuelita Selime y la tía Alicia, por su alegría, cariño y buen humor, que han servido para embalsamar las infaltables asperezas de la vida.

A Víctor y Belen, Uds. son muy importantes; gracias por todo hermanos.

También deseo agradecer en forma especial a los profesores que conforman la comisión de tesis: A don Pedro Reumay, a quién considero además mi amigo; he recibido grandes muestras de su estima y aprecio, quien ha manifestado gran entusiasmo en cada avance y progreso. Al profesor Jorge Arenas, que siempre atendió a mis dudas y consultas con gran paciencia y deferencia y por sus consejos muy oportunos para mejorar este trabajo. Al profesor Alfio Yori, por sus prácticos consejos para poder continuar el avance de la tesis durante el verano.

Agradezco en forma general, a todos y cada uno de los docentes y administrativos del instituto de acústica y matemáticas que de alguna manera me han brindado un trato cordial y que en más de alguna ocasión me han prestado su ayuda. En forma especial, a la Sra. Patricia Pelaez, gracias por toda su preocupación, amabilidad y su amistad.

A mi compañero de carrera y amigo Pablo Kogan, con quién hemos compartido mucho durante la carrera. Te deseo mucho éxito en todo viejo. Tus consejos y críticas sobre la tesis han sido vitales. Lo mismo le agradezco al profesor Fernando Miranda, quien entre broma y broma, me mostraba “como la verdad asoma”. A la profesora Mónica Canales y Michael, por haberme ayudado con la bibliografía de la tesis en un tema de matemáticas, y sobre la FFT. A mi socio Alejandro Santana, Lilí y la Fran, Uds. me han brindado incondicionalmente su amistad y hospitalidad sin pedir nada a cambio. Al profesor Hugo Ramírez, con quien he compartido amenas charlas sobre matemáticas y otros temas y que ha mostrado un especial interés por mi trabajo. A Karla y Cinthia, gracias por su amistad y paciencia. Quiero agradecer en forma especial a Lilian y a Marisol. Uds. en su momento fueron un gran apoyo, sin contar los bellos momentos que vivimos juntos. Gracias.

Finalmente, agradezco una vez más a aquel que es todopoderoso, e infinito en amor y misericordia. A Jesús, el buen pastor. Gracias Señor.

RESUMEN

En este trabajo se explica la relación que hay entre la teoría de wavelets y la teoría de filtros digitales. Los primeros capítulos abordan los fundamentos de la teoría de wavelets, del procesamiento digital de señales y el diseño de filtros digitales. Los siguientes capítulos tratan sobre el diseño de sistemas wavelet cuando estos son vistos como filtros digitales FIR. Tal punto de vista se presenta en la teoría del banco de filtros. Con las técnicas básicas del diseño de sistemas wavelet se presenta un algoritmo que permite diseñar filtros digitales FIR que, además, provienen de una familia wavelet. Finalmente, se presenta una implementación del algoritmo en MATLAB 6.0 y su desempeño en el diseño de filtros digitales.

ABSTRACT

In this work the relationship between the wavelets and the digital filters theory is explained. The first few chapters deal with the fundamentals of the wavelets theory, digital signal processing and digital filters design. The following chapters are referred to system wavelet design when these are viewed as FIR digital filters. This point of view is introduced in the theory of bank filters. Besides a knowledge of the basic techniques of wavelet system design it is introduced an algorithm that allows to design FIR digital filters that are part of a wavelet family. Finally, an implementation of the algorithm for MATLAB 6.0 is presented and its performance in designing digital filters is studied.

OBJETIVOS GENERALES

1. Entregar un resumen de la teoría de filtros digitales, Wavelets y Banco de Filtros.
2. Mostrar las bases del diseño de filtros creados con Wavelets con el fin de encontrar algoritmos que puedan ser implementados en un computador.
3. Presentar una aplicación de filtrado con un enfoque a la acústica.

OBJETIVOS ESPECÍFICOS

1. Describir el funcionamiento de los filtros digitales FIR e IIR, tanto en el dominio del tiempo, como el de la frecuencia.
2. Desarrollar la teoría Wavelet en términos básicos, como un modelo matemático de filtros digitales del tipo FIR.
3. Sintetizar los resultados de tal modelo matemático con el fin de implementar un algoritmo que permita diseñar filtros Wavelet de tipo FIR.
4. Utilizar el software MATLAB 6.0 para implementar los modelos encontrados, así como otros ya establecidos antes, para poder hacer una comparación de su rendimiento, respuesta en frecuencia y de fase.

INDICE

Agradecimientos	3
Resumen	4
Objetivos	5
Índice	6
Introducción	9
CAPÍTULO 1: BASE MATEMÁTICA: INTRODUCCIÓN A LAS WAVELETS	
1 Escalamiento y traslación de funciones	11
1.1 Traslación de funciones	11
1.2 Escalamiento de funciones	12
1.2.1 Escalamiento con potencias de 2	13
1.3 La función escala	16
1.3.1 Propiedades de la función escala	16
1.4 La función wavelet	18
1.4.1 Relación entre la función escala y la wavelet	18
1.4.2 Relación entre $h(n)$ y $h_I(n)$	19
1.5 Sistemas wavelet	19
1.5.1 Sistemas wavelet de primera generación	19
1.5.2 Características de los sistemas wavelet	20
1.6 Expansión de funciones	20
1.7 Transformadas Wavelet continua y discreta	27
1.7.1 Transformada continua wavelet	27
1.7.2 Transformada discreta wavelet	27
CAPÍTULO 2: FUNDAMENTOS DEL PROCESAMIENTO DIGITAL DE SEÑALES	
2.1 Sistemas discretos en el tiempo	29
2.1.1 Función delta	29
2.1.2 Propiedades de la función delta	29
2.1.3 Sistemas discretos lineales e invariantes en el tiempo	30
2.1.4 Linealidad de los operadores DSP	30
2.1.5 Respuesta al Impulso	31
2.1.6 Convolución y salida del sistema	31
2.1.6.1 Convolución	31
2.1.7 La función delta y la respuesta al impulso	32
2.1.8 Invariancia en el tiempo	32
2.1.9 Estabilidad de los operadores DSP LTI	33
2.1.9.1 Operadores DSP causales	33
2.1.10 Ecuaciones de diferencias	33
2.2 Transformada Z	34
2.2.1 Introducción	34
2.2.2 La naturaleza de la transformada Z	34

2.2.3	Definición	35
2.2.4	Propiedades	37
2.3	Sistemas DSP LTI y la transformada Z	37
2.3.1	Polos y Ceros de $H(z)$	38

CAPÍTULO 3: FILTROS

3.1	Introducción y definición	40
3.2	Ventajas de los filtros digitales	40
3.3	Respuestas de un filtro	41
3.3.1	Respuesta a impulso	42
3.3.2	Respuesta al paso	42
3.3.3	Respuesta en frecuencia	42
3.4	Parámetros de un filtro	43
3.4.1	Parámetros de un filtro en el dominio del tiempo	43
3.4.2	Parámetros de un filtro en el dominio de la frecuencia	45
3.4.2.1	Respuestas en frecuencia	45
3.4.2.2	Parámetros	45
3.5	Clasificación de los filtros digitales	47
3.5.1	Características de los Filtros FIR e IIR	47

CAPÍTULO 4: DISEÑO DE FILTROS DIGITALES

4.1	Proceso de diseño de filtros digitales	48
4.1.1	Especificaciones del filtro digital	48
4.2	Análisis en Z de filtros digitales	49
4.2.1	Expresión de $H(z)$ en términos de sus Polos y Ceros	50
4.3	Descripción matemática del diseño de filtros FIR	51
4.3.1	Simetría y antisimetría en la fase lineal	53
4.3.1.1	Caso simétrico	53
4.3.1.2	Caso antisimétrico	53
4.4	Técnicas Especiales de Filtrado FIR	55
4.4.1	La convolución como técnica de filtrado	55
4.4.1.1	Una visión práctica de la convolución	56
4.4.1.2	Filtros implementados por convolución	57
4.4.1.3	Ejemplos implementados en MATLAB	58
4.4.2	Método MA	60
4.4.2.1	Implementación por convolución	60
4.4.2.2	Reducción del ruido v/s respuesta al paso	61
4.4.2.3	Respuesta en frecuencia	62
4.4.3	Método de enventanado sinc	63
4.4.3.1	Estrategia del enventanado sinc	63
4.4.3.2	Algunas consideraciones sobre ventanas	66
4.4.3.3	Diseño de filtros	67
4.5	Filtros recursivos	70
4.5.1	El método recursivo	70
4.5.2	Diseño de filtros digitales recursivos	71
4.5.2.1	Filtros “Single Pole”	71
4.5.2.2	Filtros “Notch”	75
4.5.3	Respuesta de fase de los filtros digitales	76

CAPÍTULO 5: WAVELETS Y FILTROS DIGITALES

5.1 Banco de filtros	81
5.1.1 Análisis	82
5.1.1.1 Filtrado y “Down-sampling”	83
5.1.2 Síntesis	86
5.1.2.1 Filtrado y “Up-sampling”	87
5.1.3 Interpretación de la transformada wavelet en términos de bancos de filtros	88
5.2 Condiciones de suficiencia	91
5.2.1 Transformaciones de Fourier	91
5.2.2 Condiciones necesarias en el dominio del tiempo	92
5.2.3 Condiciones necesarias en el dominio de la frecuencia	93
5.2.4 Condiciones suficientes	95
5.3 Diseño de Wavelets	97
5.3.1 Método de los momentos ceros de Daubechies	97
5.3.1.1 Momentos	98
5.3.1.2 Momentos wavelet desvanecidos	99
5.3.1.3 Método de diseño del momento wavelet cero	100
5.3.2 Wavelet Coiflet	102
5.3.2.1 Desvanecimiento de los momentos de las funciones escala	102
5.3.2.2 Aproximación de funciones mediante la proyección de la función escala	102
5.3.2.3 Aproximación de los coeficientes escala por términos de la señal	103
5.3.2.4 Coiflets	104
5.3.3 Simlets	105

CAPÍTULO 6: PARTE EXPERIMENTAL

6.1 Método de diseño de wavelets mediante filtros digitales FIR	106
6.1.1 Proposición y desarrollo del problema	106
6.1.2 Comparación del método con antecedentes teóricos previos	111
6.1.3 Síntesis del proceso de diseño	111
6.1.4 Algoritmo y solución del sistema no lineal	112
6.1.5 Implementación en MATLAB	113
6.2 Aplicación de la DWT a una señal con ruido	118
6.2.1 Reconstrucción de la señal filtrada	121
DISCUSIÓN Y PROYECCIONES DE TRABAJO	124
CONCLUSIONES	127
BIBLIOGRAFÍA	128
APENDICE A Gráficas de filtros digitales wavelet	
APENDICE B Implementaciones en MATLAB	
APENDICE C Tablas de coeficientes wavelet	

INTRODUCCIÓN

Una de las herramientas fundamentales en el análisis de señales en general, son los filtros, lo cual representa un importante tema de la ingeniería, que inicialmente se desarrolló de la electrónica, con el fin de modificar una señal eléctrica analógica. Posteriormente, con la aparición del computador, se hizo posible representar cualquier señal analógica como una colección de puntos, donde cada uno de ellos lleva asociado un número binario. La revolución que esto produjo fue enorme, pues a partir de ese momento cualquier señal podía ser representada por una cantidad finita de información, y ser procesada a través del lenguaje del computador, que en este caso corresponde al lenguaje binario. Invertir, sumar y separar señales se transformó en una tarea rápida y en extremo eficiente dejando atrás las limitaciones físicas de los filtros analógicos. Nace entonces, una nueva área de la ingeniería conocida como Procesamiento Digital de Señales o DSP (Digital Signal Processing). En esta nueva materia el filtro analógico encontró a su hermano gemelo: el filtro digital. Este corresponde a un importante tema en DSP, y ha sido motivo para muchas y variadas investigaciones. Una importante materia de investigación hasta ahora, ha sido la búsqueda de algoritmos matemáticos y computacionales que permitan implementar filtros digitales de calidad y más veloces. Sin duda, el más popular y tal vez el más usado sea la Transformada Rápida de Fourier o FFT (Fast Fourier Transform). La revolución que este algoritmo produjo fue enorme, pues aquellos procesos que utilizaban la transformada discreta de Fourier eran efectivos pero demasiado lentos. La Transformada Rápida de Fourier es muy eficiente cuando es aplicada a señales periódicas, debido a que las bases de esta son justamente las funciones seno y coseno, funciones que presentan periodicidad y representación individual en la frecuencia, lo cual las hace extremadamente útiles para el estudio de señales estacionarias o invariantes en el tiempo y fenómenos periódicos. Sin embargo, para señales que presentan discontinuidades o cambios abruptos de amplitud en el tiempo, tales como ráfagas de sonido o de luz, el contenido de frecuencia es variable en el tiempo. Una herramienta alternativa frente a esta situación es la teoría de Wavelets.

Los orígenes de las Wavelets no son tan nuevos como podría pensarse. De hecho, estos son muy cercanos al nacimiento de la teoría de Fourier. En 1909 Alfred Haar descubrió las primeras wavelets, que consisten de un breve impulso positivo, seguido de otro negativo. Tan solo en 1981, un descubrimiento hecho por Jean Morlet, geofísico francés e ingeniero petrolero, quién investigaba la propagación de ondas sísmicas a través de la corteza terrestre, permitiría formalizar y estructurar en una teoría matemática el descubrimiento hecho por Haar. Morlet descubre que versiones dilatadas y trasladadas de una "base" (la cual se conoce actualmente como Wavelet Morlet), permitían representar las señales de propagación en el tiempo como una combinación lineal de dichas funciones base con la misma robustez y versatilidad que las series de Fourier, pero sin sus limitaciones en periodos no estacionarios. La elegancia y simplicidad de esta nueva técnica matemática fue reconocida por el matemático francés Yves Meyer, quién descubrió que estos conjuntos de funciones generados mediante dos procesos llamados escalamiento y traslación constituían un espacio ortogonal entre sí, lo cual aseguraba el cálculo de los coeficientes como una tarea rápida y sencilla. En 1984, Morlet y el físico Alex Grossman introducen el término "wavelet" en el lenguaje

matemático. Pocos años más tarde, en 1985, Meyer, descubre las primeras wavelets ortogonales suaves. Un descubrimiento aislado hecho por Denis Gabor tendría un impacto importante en lo que sería después el análisis de señales. En 1946, Gabor adaptó la transformada de Fourier utilizando un proceso llamado "ventanamiento", el cual consiste en descomponer las señales en "paquetes de tiempo-frecuencia", también conocidas como frecuencias de Gabor. De esta manera, al aplicar una función en el tiempo, llamada "ventana", se hizo posible obtener el espectro de frecuencia para un intervalo de tiempo con una longitud previamente establecida, pudiendo incluso obtener un espectro casi instantáneo de la señal si se considera una longitud de la ventana lo suficientemente pequeña.

En 1986 Stéphane Mallat, por entonces en la Universidad de Pennsylvania, demuestra que la base de Haar, las octavas de Littlewood-Paley, las frecuencias de Gabor y los filtros subbanda de Galand y Esteban están todos relacionados con algoritmos basados en wavelets. Estos resultados permitieron modelar la transformada discreta wavelet como un filtro digital, estrechando aún más la relación entre las wavelets y el procesamiento digital de señales. Este constituye el tema principal de la presente tesis: explicar de una manera clara y coherente la relación entre las wavelets y el diseño de filtros digitales en base a ellas.

CAPÍTULO 1: BASE MATEMÁTICA

1. ESCALAMIENTO Y TRASLACIÓN DE FUNCIONES

1.1 Traslación de funciones

Sea $\varphi(t)$ en $L^2(\mathbb{R})$. Luego, esta función es de energía finita, es decir, esta acotada si ella está definida en un dominio finito [FAF2000]

Se dice que una función $\varphi(t)$ esta **trasladada** en k con $k \in \mathbb{Z}$, sí:

$$\phi(t-k)_{t=t+k} = \phi(t)_{t=t} \quad \forall t \in \mathfrak{R} \quad (1.1)$$

Por ejemplo, consideremos la función valor absoluto. Entonces, $\phi(t) = |x|$ y la correspondiente traslación en k será $\phi(t-k) = |x-k|$. En la figura 1 se ilustran algunas versiones trasladadas de esta función.

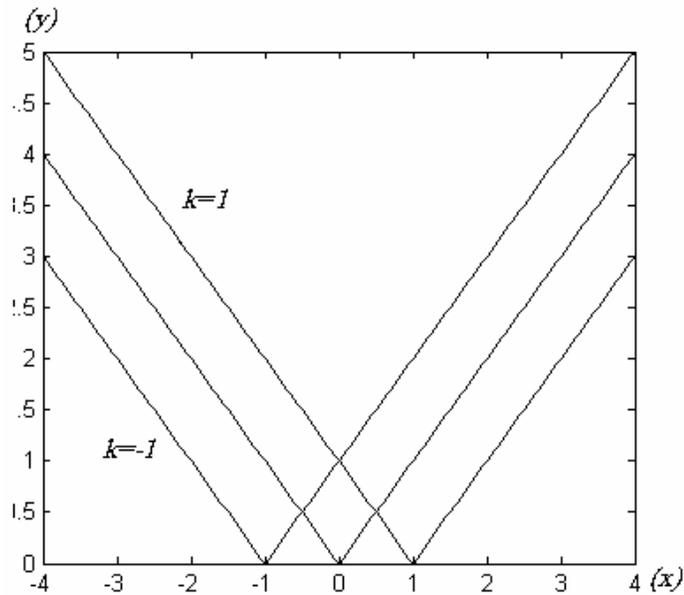


Figura 1. Versiones trasladadas de la función Valor Absoluto.

Puesto que la traslación es válida para todo k real, se puede hablar del conjunto de funciones generadas por traslación de la forma:

$$\{\phi_k(t) = \phi(t-k) | k \in \mathbb{Z}\}$$

1.2 Escalamiento de funciones

Consideremos una función $\varphi(t)$ en $L^2(\mathbb{R})$, y $a \in \mathbb{Z}$. Entonces se dice que la función $\varphi(at)$ corresponde a la versión escalada en un factor a de la función $\varphi(t)$. Matemáticamente, está escalada en a sí:

$$\varphi(at)_{t=t/a} = \varphi(t)$$

Usemos nuevamente como ejemplo la función Valor Absoluto.

Tenemos que $\phi(t) = |x|$ y $\phi(at) = |ax|$ con $a \in \mathbb{Z}$. Para algunos valores de a tendremos una nueva familia de funciones. En la figura 2 podemos ver algunas versiones escaladas.

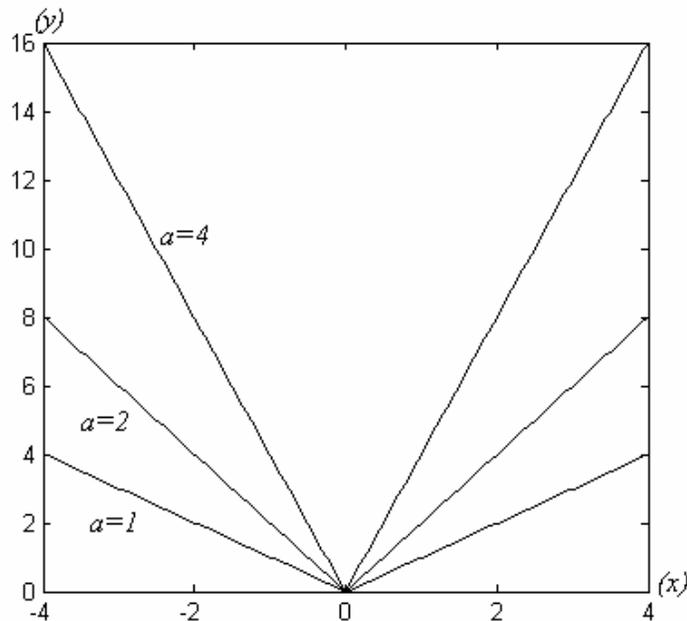


Figura 2. Versiones escaladas o dilatadas de la función Valor Absoluto.

Podemos decir que con los procesos de *escalamiento* y *traslación* de la función $\varphi(t)$ podemos generar una familia o *conjunto* de funciones. Tales conjuntos estarán definidos en cada caso por:

$$\{\phi_k(t) = \phi(t - k) | k \in \mathbb{Z}\}, \text{ en la traslación}$$

$$\{\phi_a(t) = \phi(at) | a \in \mathbb{Z}\}, \text{ para el escalamiento}$$

1.2.1 Escalamiento con potencias de 2

Consideremos el caso en que una función $\varphi(t)$ trasladada en k y escalada por el factor 2^j . Entonces, si j y k son arbitrarios, tendremos un conjunto generado de la siguiente forma:

$$\{\phi_{j,k}(t) = \phi(2^j t - k) \mid j, k \in \mathbb{Z}\}. \quad (1.2)$$

En la siguiente figura observamos algunos elementos de este conjunto así definido:

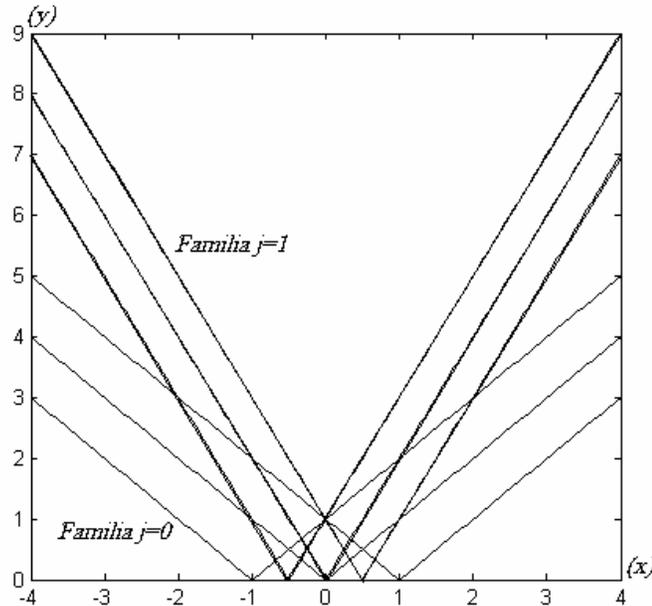


Figura 3. Versiones escaladas y trasladadas de la función $\varphi(t)$. Aquí se usa $\phi_{j,k}(t) = \phi(2^j t - k)$; $j, k \in \mathbb{Z}$

Observamos que este conjunto generado de funciones es bidimensional, en el sentido que dependen de dos variables, en este caso, j y k . Para cada valor de j (esto es, para un j fijo) tenemos una familia de funciones generadas solo por traslación. Hablaremos entonces del conjunto $\varphi(t)$ en el nivel j para el conjunto $\phi_{j,k}(t) = \phi(2^j t - k)$; $j, k \in \mathbb{Z}$. En la figura 3 se observan elementos de este conjunto para los niveles $j=0$ y $j=1$.

Es interesante notar que para un nivel más alto de j el conjunto generado del ejemplo es más denso en el sentido que es posible cubrir más puntos de la recta real (los vértices).

Para un número mayor de elementos podemos “ver” como son estas familias de funciones. En las figuras 4 a la 7 se muestran algunos ejemplos implementados en MATLAB 6.0.

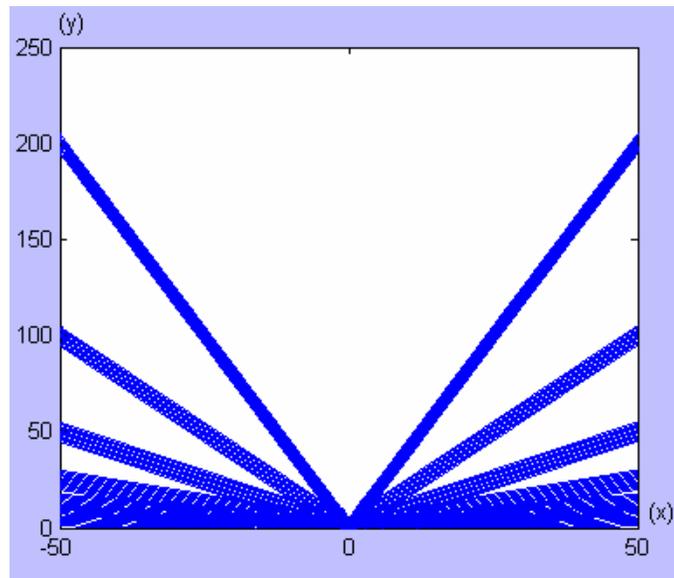


Figura 4. Familia de funciones para $abs(x)$

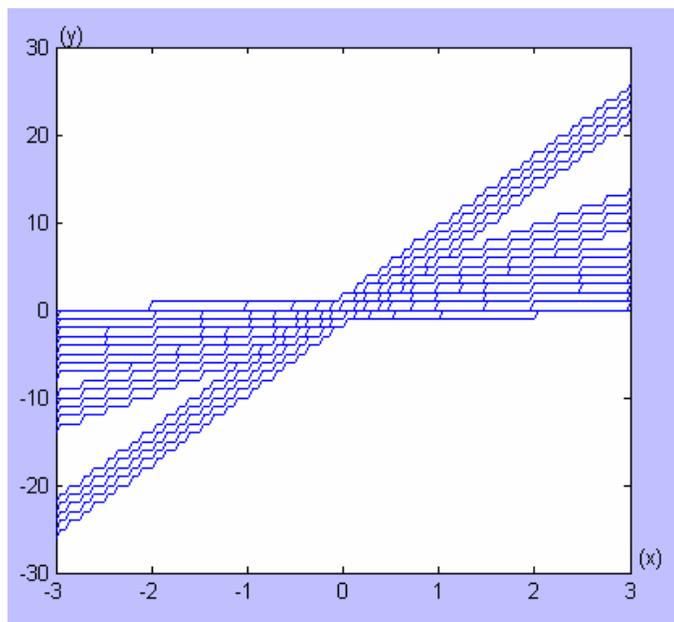


Figura 5. Familia de funciones para $fix(x)$

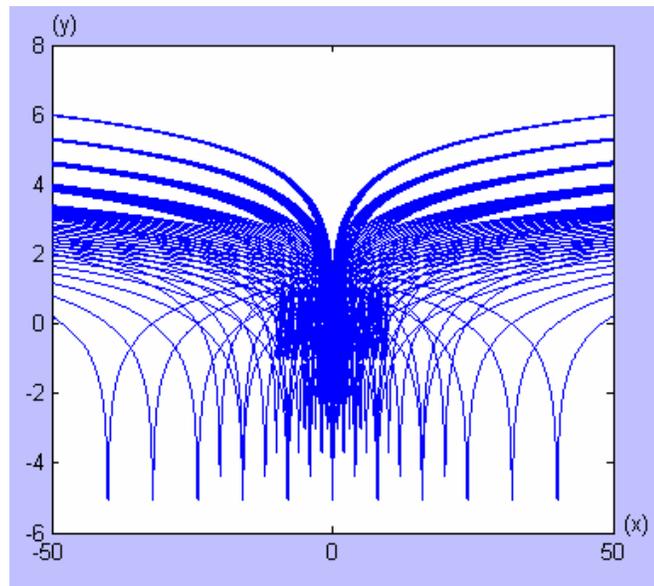


Figura 6. Familia de funciones para $\log(x)$

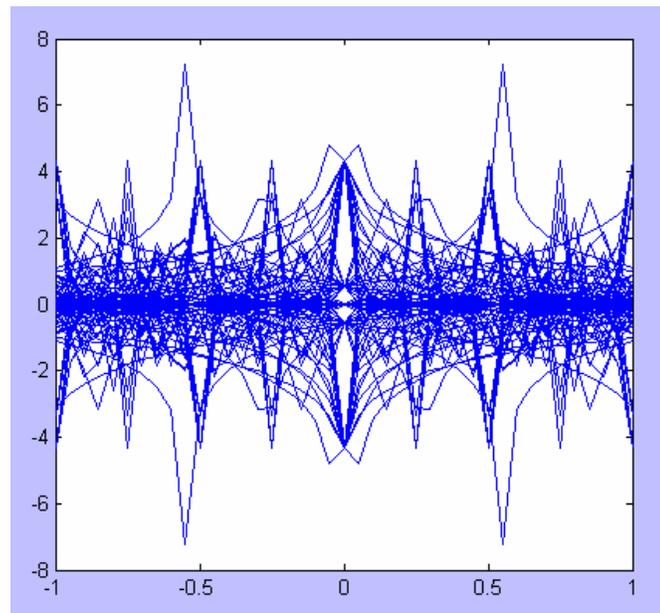


Figura 7. Familia de funciones para $\log(\cos(\tan(x)))$

1.3 LA FUNCIÓN ESCALA

Sea $\varphi(t)$ en $L^2(\mathbb{R})$, una función escala, que *trasladada* y *escalada* genera una familia de funciones de la siguiente forma:

$$\phi_{j,k}(t) = 2^{\frac{j}{2}} \phi(2^j t - k) \quad j, k \in \mathbb{Z}. \quad (1.3)$$

Para cada $j \in \mathbb{Z}$ esta función genera un subespacio ortonormal $V_j \subset L^2(\mathbb{R})$ definido por [FAF2000]:

$$V_j = \overline{\text{Span}_{k \in \mathbb{Z}} \phi_{j,k}(t)}. \quad (1.4)$$

Entonces una función $f(t)$ estará en V_j si puede escribirse como:

$$f(t) = \sum_{k \in \mathbb{Z}} c_{j,k} \phi_{j,k}(t), \quad (1.5)$$

donde

$$c_{j,k} = \langle f(t), \phi_{j,k}(t) \rangle = \int_{-\infty}^{\infty} |f(t) \phi_{j,k}(t)| dt. \quad (1.6)$$

1.3.1 Propiedades de la función escala

1. $f(t)$ pertenece a V_j si y solo si $f(2t)$ pertenece a V_{j+1} .
2. Para cada $j \in \mathbb{Z}$, $\phi_{j,k}$ forma una base ortonormal para el subespacio $V_j \subset L^2(\mathbb{R})$, $k \in \mathbb{Z}$.
3. Los subespacios V_j están anidados, es decir:

$$\forall j \in \mathbb{Z}; V_j \subset V_{j+1} \quad (1.7)$$

4. La función $\varphi(t)$ tiene soporte compacto. Esto es, existe un subconjunto del dominio de $\varphi(t)$ donde esta no es cero.

Proposición 1

El anidamiento de los espacios se puede expresar como[FAF2000]:

$$\forall j \in \mathbb{Z}; V_j \subset V_{j+1} \text{ Si y solo si } \varphi(t) \in V_j. \quad (1.8)$$

Como consecuencia de esta proposición, se desprende el hecho de que existen coeficientes $h(n)$, $n \in \mathbb{Z}$, tal que $\varphi(t)$ se puede expresar en términos de una suma ajustada y trasladada como:

$$\varphi(t) = \sum_n h(n) \sqrt{2} \varphi(2t-n), \quad (1.9)$$

llamada *ecuación básica de recursión* o ecuación de escala, donde $\varphi(t)$ tiene soporte compacto sobre el intervalo $[0, N-1]$. Los coeficientes $h(n)$ pueden ser una secuencia de números reales o complejos, llamados coeficientes de función escala (o filtro de escalamiento o vector de escalamiento) y $\sqrt{2}$ mantiene la norma de la función escala en V_1 .

Proposición 2

Si $\varphi(t)$ es una función escala como se define en la proposición 1, que tiene soporte compacto sobre $[0, N-1]$ y $\{\varphi(2t-k) | k \in \mathbb{Z}\}$ es base ortonormal para V_1 , entonces $h_1(n)$ también tiene soporte compacto sobre $[0, N-1]$, por lo que la máxima longitud de la secuencia $h(n)$ es N .

Proposición 3

Si $h(n)$ es un coeficiente de la ecuación (1.9) entonces:

$$h(n) = \frac{1}{\sqrt{2}} \int_{-\infty}^{\infty} \varphi(t) \varphi(2t-n) dt. \quad (1.10)$$

1.4 LA FUNCIÓN WAVELET

Definimos:

$$W_j = \overline{\text{Span}_{k \in \mathbb{Z}} \psi_{j,k}(t)}, \quad (1.11)$$

como el complemento ortogonal de V_j en V_{j+1} . Esto significa que todos los miembros de V_j son ortogonales a todos los miembros de W_j . Es decir:

$$c_{j,k} = \langle \phi_{j,k}(t), \psi_{j,k}(t) \rangle = \int_{-\infty}^{\infty} \phi_{j,k}(t) \psi_{j,k}(t) dt = 0 \quad j, k \in \mathbb{Z}, \quad (1.12)$$

$$\text{con } \psi_{j,k}(t) = 2^{\frac{j}{2}} \psi(2^j t - k) \quad j, k \in \mathbb{Z}, \quad (1.13)$$

y además

$$V_1 = V_0 \oplus W_0, \quad (1.14)$$

donde cualquier función $f(t) \in W_j$ puede ser representada como:

$$f(t) = \sum_{k \in \mathbb{Z}} d_{j,k} \psi_{j,k}(t). \quad (1.15)$$

La función wavelet básica $\psi(t) = \psi_{0,0}(t) \in W_0$, y $V_1 = V_0 \oplus W_0$, entonces $\psi(t) \in V_1$.

1.4.1 Relación entre la función escala y la wavelet

Del razonamiento anterior se deduce que:

$$\psi(t) = \sum_{k \in \mathbb{Z}} \sqrt{2} h_1(n) \phi(2t - k). \quad (1.16)$$

A través de esta ecuación podemos determinar la función wavelet, si conocemos la función escala.

Los coeficientes $h_1(n)$ pueden ser una secuencia de números reales o complejos, llamados **coeficientes de la función wavelet** (o filtro de escalamiento o vector de escalamiento) y $\sqrt{2}$ mantiene la norma de la función wavelet en V_1 .

1.4.2 Relación entre $h(n)$ y $h_1(n)$

Si la wavelet es el complemento ortogonal de la función escala, en cualquier nivel, entonces las secuencias $h(n)$ y $h_1(n)$ deben cumplir con la siguiente relación [BUR1998]:

$$h_1(n) = (-1)^n h(1-n). \quad (1.17)$$

Para una secuencia finita de largo N , tenemos:

$$h_1(n) = (-1)^n h(N-1-n). \quad (1.18)$$

Esta última ecuación es importante porque permite calcular los coeficientes wavelet si se conocen previamente los coeficientes de escala.

1.5 SISTEMAS WAVELET

1.5.1 Sistemas Wavelet de primera generación

Los sistemas Wavelet de primera generación son todos aquellos que son generados sólo por traslaciones enteras y escalamientos de una única función Wavelet $\psi(t)$, llamada *wavelet madre* o wavelet generadora, que da a luz a una familia de funciones de la forma:

$$\psi_{j,k}(t) = 2^{\frac{j}{2}} \psi(2^j t - k) \quad j, k \in Z, \quad (1.19)$$

donde el factor $2^{\frac{j}{2}}$ mantiene una norma constante e independiente de la escala j .

Esta familia de funciones es llamada el *set de expansiones wavelet*.

La wavelet madre $\psi(t)$, trae asociada consigo una función escala $\phi(t)$. Con estas dos funciones es posible aproximar cualquier función o señal $f(t)$ en $L^2(R)$, mediante:

$$f(t) = \sum_k \sum_j c_{j,k} \phi(t) + \sum_k \sum_j d_{j,k} \psi(t). \quad (1.20)$$

1.5.2 Características de los sistemas wavelet

Un sistema wavelet puede describirse como un conjunto de ladrillos (que para cada sistema pueden tener diferente forma), que sirven para reconstruir o representar una señal o función [FAF2000]. Este conjunto es una expresión bidimensional, que suele ser una base para alguna clase de señal de una o más dimensiones. En otras palabras, si el set de expansión está dado por $\{ \psi_{j,k}(t) | j, k \in Z \}$ una expansión lineal puede ser de la forma:

$$f(t) = \sum_k \sum_j d_{j,k} \psi(t).$$

La expansión wavelet entrega una localización tiempo frecuencia instantánea de la señal, lo que significa que un coeficiente wavelet nos indica cuando ocurre el tono y cual es su frecuencia. Esto quiere decir que la mayor parte de la energía de la señal es bien representada por unos pocos coeficientes $d_{j,k}$. Mientras que un coeficiente de Fourier representa un componente de frecuencia que dura todo el tiempo en que se extiende la señal, un coeficiente de expansión wavelet representa una componente bien definida en un intervalo de tiempo

Los sistemas wavelet satisfacen las condiciones de multiresolución [FAF2000]. Esto significa que si un conjunto de señales puede ser representado por una suma de $\varphi(t-k)$, con k en Z , un conjunto más amplio de señales (que incluye al conjunto original), puede ser representado por una suma $\varphi(2t-k)$, $k \in Z$.

Los coeficientes de más baja resolución pueden ser calculados a partir de los coeficientes de más alta resolución, mediante un algoritmo en forma de árbol llamado banco de filtros. Esto permite un buen cálculo de los coeficientes de la expansión (conocido como la Transformada Discreta wavelet). El tamaño de los coeficientes de la expansión wavelet disminuye rápidamente con j y k .

Los wavelets son ajustables y adaptables. Debido a que existen muchas wavelet, estos pueden ser diseñados para adaptarse a una aplicación particular.

1.6 EXPANSIÓN DE FUNCIONES

Veremos a continuación como es posible representar una función $f(t)$ a través de una expansión de funciones escala de la forma:

$$f(t) = \sum_k \sum_j c_{j,k} \phi(t).$$

Usemos como ejemplo, el **sistema Haar** [FAF2000, BUR1998]. Este es considerado como el más sencillo y simple de usar y está definido por:

$$\begin{aligned}\phi(t) &= 1, \text{ Si } 0 \leq t \leq 1 \\ \phi(t) &= 0 \text{ en otro caso.}\end{aligned}\tag{1.21}$$

Luego, una versión trasladada en k será:

$$\begin{aligned}\phi_k(t) &= \phi(t-k) = 1 \text{ Si } t_1 = k \leq t \leq k+1 = t_2 \\ \phi_k(t) &= 0 \text{ en caso contrario.}\end{aligned}$$

Si escalamos en el tiempo por un factor 2^j , entonces la versión escalada y trasladada será:

$$\begin{aligned}\phi_{j,k}(t) &= \phi(2^j t - k) = 1 \text{ si } t_1 = \frac{k}{2^j} \leq t \leq \frac{k+1}{2^j} = t_2 \\ \phi_{j,k}(t) &= 0 \text{ en otro caso.}\end{aligned}$$

Luego, la expansión de $f(t)$ en el nivel j , en términos de la función escala, tendrá la siguiente forma:

$$f(t) = \sum_k \sum_j c_{j,k} \phi_{j,k}(t) = c_{j,-\infty} \phi_{j,-\infty}(t) + \dots + c_{j,k-1} \phi_{j,k-1}(t) + c_{j,k} \phi_{j,k}(t) + c_{j,k+1} \phi_{j,k+1}(t) + \dots + c_{j,\infty} \phi_{j,\infty}(t)\tag{1.22}$$

Puesto que el conjunto $\{\phi_{j,k}(t) | j, k \in \mathbb{Z}\}$ forma una base ortogonal, los coeficientes $c_{j,k}$ pueden ser calculados a través del producto interno (1.6):

$$c_{j,k} = \langle f(t), \phi_{j,k}(t) \rangle = \int_{-\infty}^{\infty} f(t) \phi_{j,k}(t) dt.\tag{1.23}$$

Entonces, los coeficientes $c_{j,k}$ pueden ser calculados por:

$$c_{j,k} = \int_{-\infty}^{\infty} f(t) \phi(2^j t - k) dt.\tag{1.24}$$

Para el caso de la función Haar escala, tendremos:

$$c_{j,k} = \int_{k/2^j}^{(k+1)/2^j} f(t) * 1 dt.\tag{1.25}$$

Podemos notar claramente que el cálculo de los coeficientes escala se reduce a una integral sencilla, si usamos por ejemplo, el sistema Haar. Esto indica que un set de funciones que cumplan con la condición de ortogonalidad garantiza un fácil y rápido cálculo de los coeficientes de expansión.

Veamos como ejemplo la expansión finita de una función $f(t)$ utilizando el sistema Haar escala. Consideremos la función $f(t) = \text{sinc}(t)$. Esta está definida por:

$$\text{sinc}(t) = \frac{\text{sen}(t)}{t}. \quad (1.26)$$

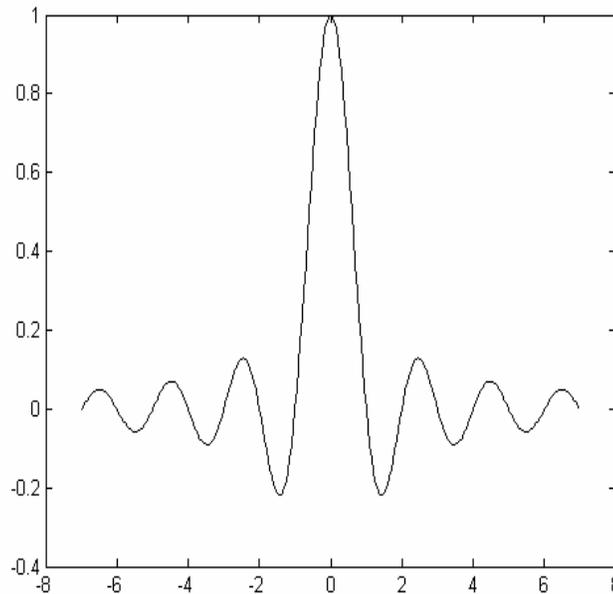


Figura 8. Gráfica de la función $\text{sinc}(t)$

A continuación mostramos diferentes versiones de $f(t) = \text{sinc}(t)$ para diferentes niveles de j . Las representaciones gráficas fueron obtenidos mediante una implementación de la última integral en el lenguaje de simulación MATLAB 6.0 [NAK1997].

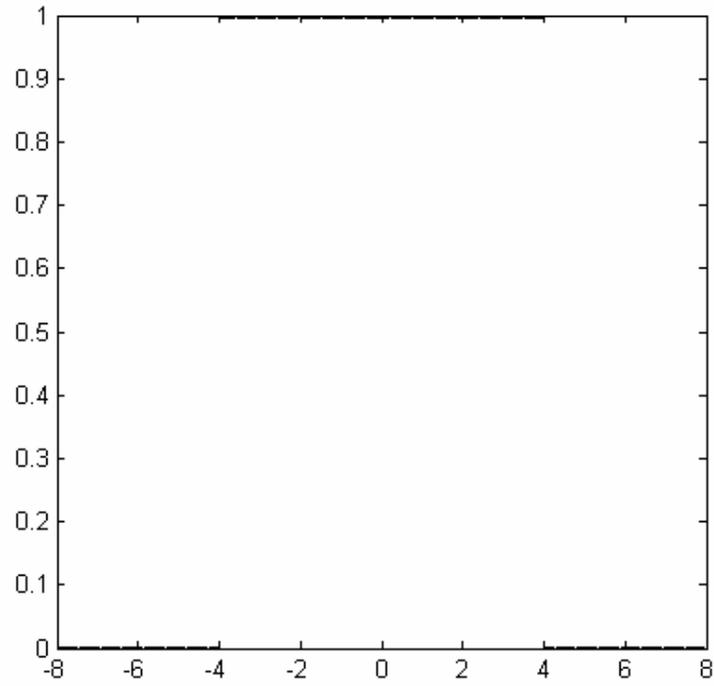


Figura 9. Nivel $j=-2$

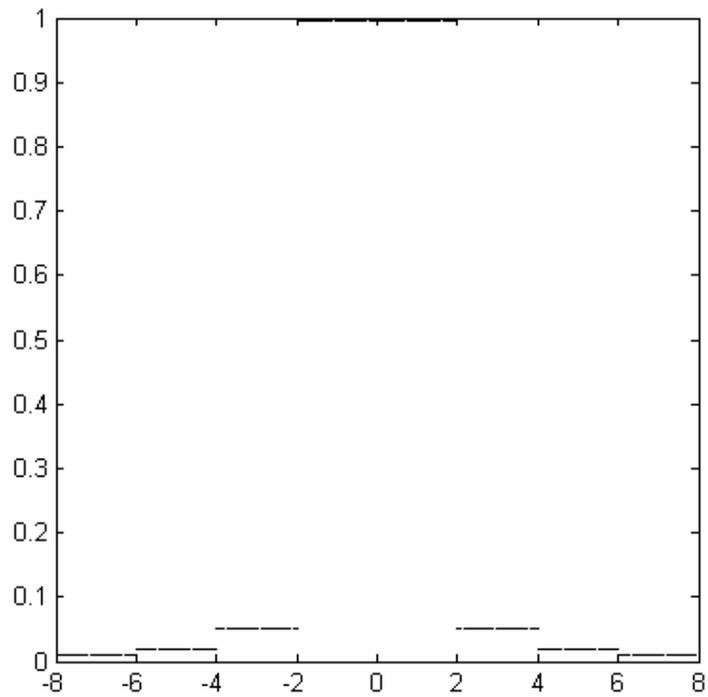


Figura 10. Nivel $j=-1$

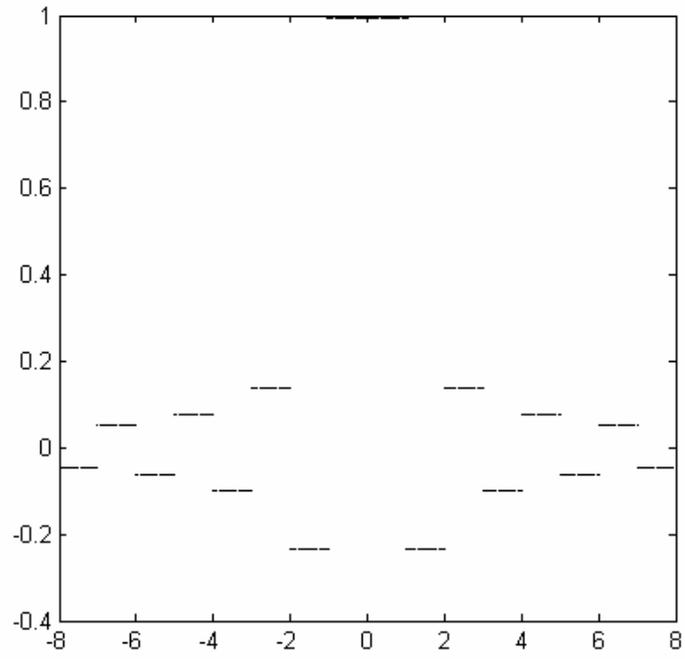


Figura 11. Nivel $j=0$

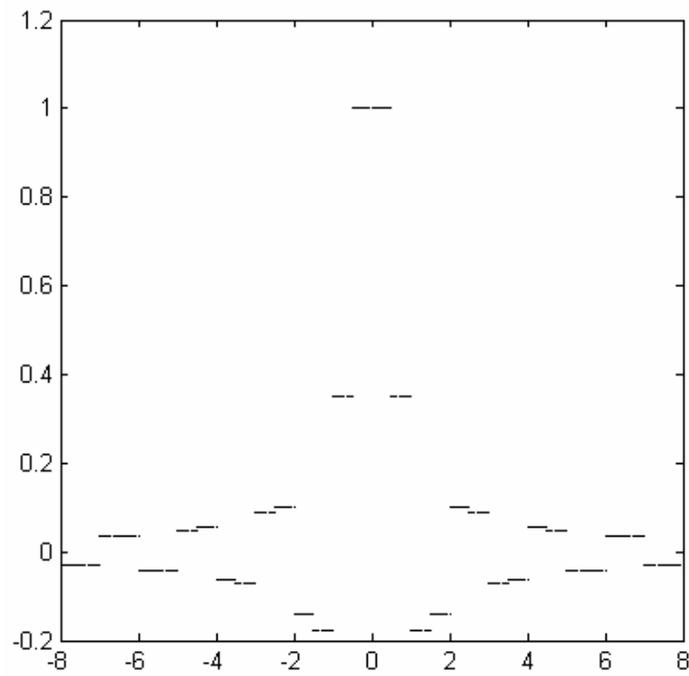


Figura 12. Nivel $j=1$

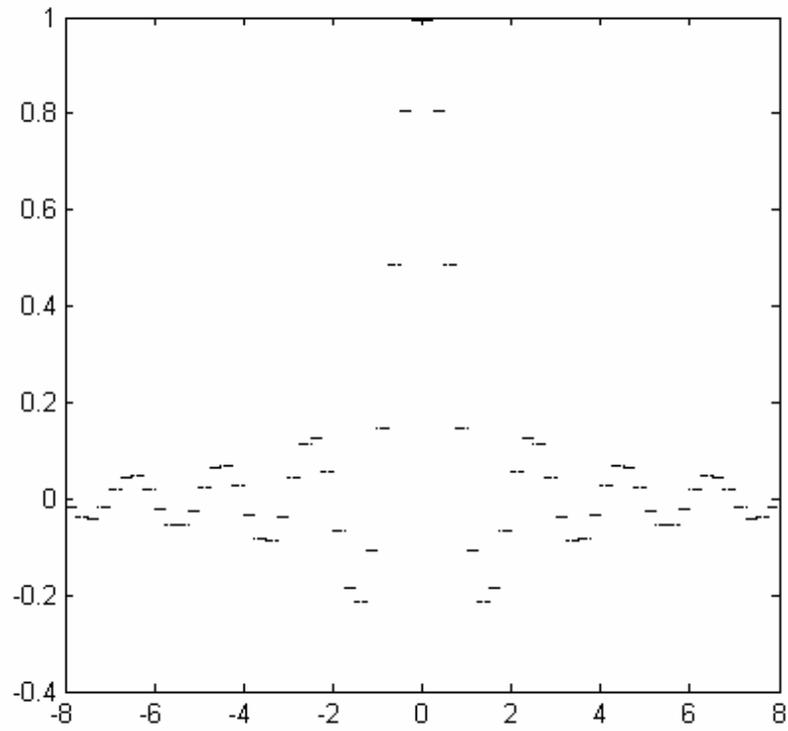


Figura 13. Nivel $j=2$

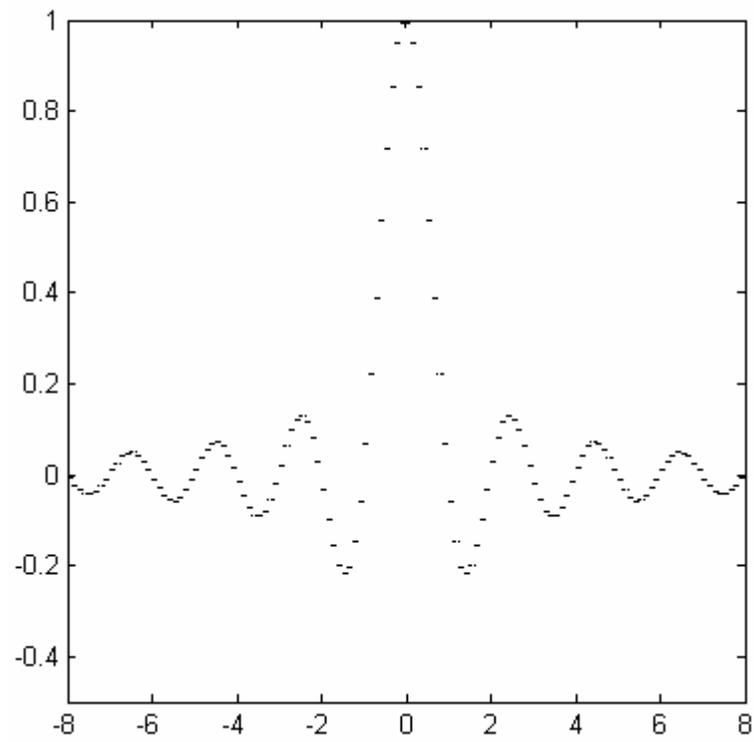


Figura 14. Nivel $j=3$

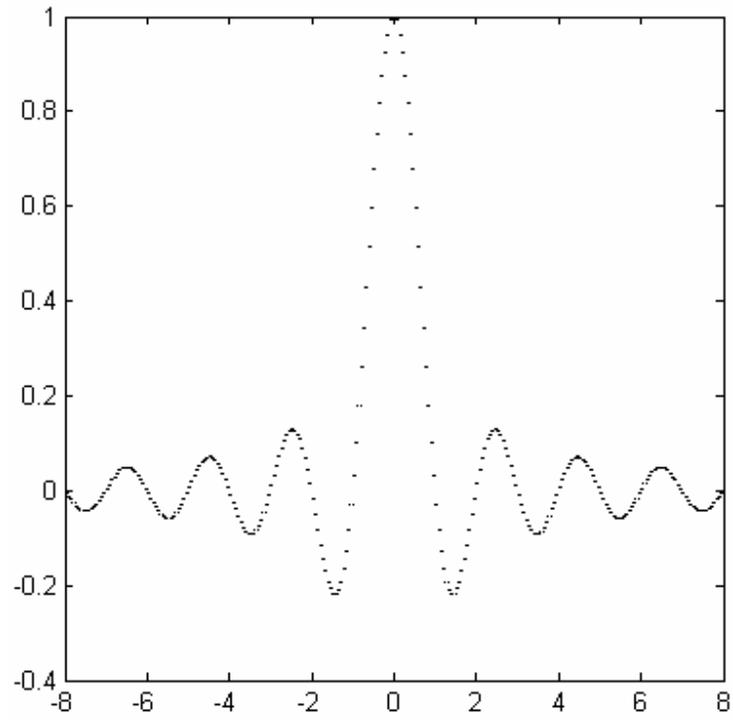


Figura 15. Nivel $j=4$

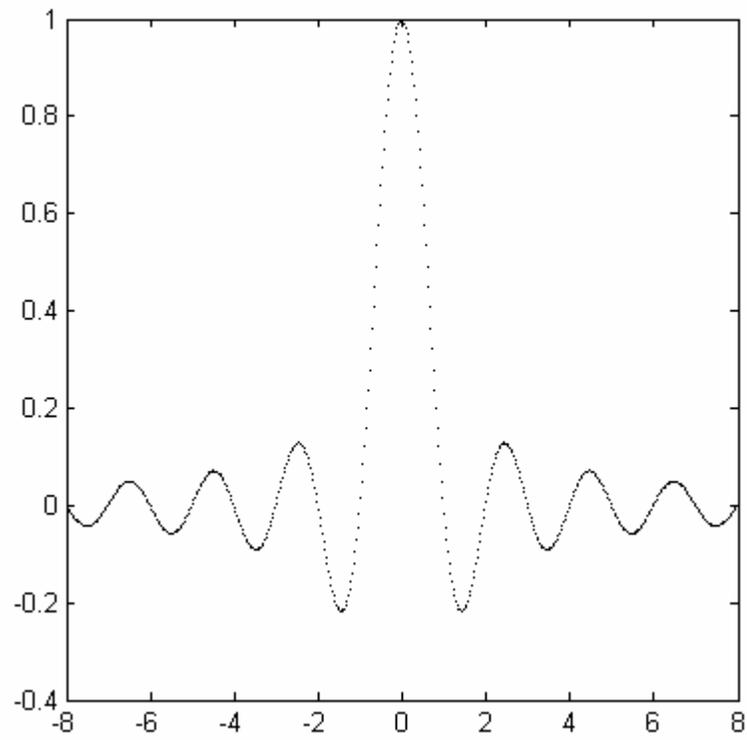


Figura 16. Nivel $j=5$

1.7 TRANSFORMADA CONTINUA WAVELET

La transformada continua Wavelet permite expresar una señal continua $x(t)$ en el dominio del tiempo, mediante una expansión de términos o coeficientes proporcionales al producto interno entre la señal y diferentes versiones escaladas y trasladadas de una función prototipo $\psi(t)$, también conocida como *wavelet madre*. Asumiremos además que la señal $x(t)$ y la función $\psi(t)$, son de energía finita [FAF2000].

Definición:

La Transformada Continua Wavelet (*Cotinuuous Wavelet Transformer*) está dada por:

$$CWT(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi\left(\frac{t-b}{a}\right) dt. \quad (1.27)$$

Observación.

El teorema de Parseval [HAS1973] relaciona la CWT con la transformada de Fourier por la siguiente expresión:

$$CWT(a, b) = \frac{1}{2\pi\sqrt{a}} \int_{-\infty}^{\infty} X(\omega) \Psi(a\omega) e^{-j\omega b} dt. \quad (1.28)$$

La continuidad de la CWT reside en que tanto la variable de escala como la variable de traslación varían en forma continua. Por lo mismo, si deseamos discretizar la transformada, es razonable pensar que las variables de escala y traslación deben pasarse al dominio discreto.

1.8 TRANSFORMADA DISCRETA WAVELET

Anteriormente vimos que la función wavelet continua para la transformada está dada por:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right).$$

Como a y b son arbitrarios, esto genera un conjunto infinito de funciones en un espacio bidimensional. Una manera conveniente de discretizar las variables a y b consiste en utilizar una *red diádica*, es decir:

$$\begin{aligned} a &= 2^j, \quad b = k2^j \\ \therefore \psi_{j,k}(t) &= 2^{\frac{j}{2}} \psi(2^j t - k), \end{aligned} \quad (1.29)$$

que nos entrega la wavelet madre del sistema de primera generación dado por la ecuación (1.1), pero ahora para la función wavelet.

Definición:

La transformada Discreta Wavelet o DWT (*Discrete Wavelet Transform*) está dada por [FAF2000, BUR1998]:

$$f(t) = \sum_{k=0}^{2^{j_0}-1} c_{j_0,k} \phi_{j_0,k}(t) + \sum_{j=j_0}^{N-1} \sum_{k=0}^{2^j-1} d_{j,k} \psi_{j,k}(t), \quad (1.30)$$

donde $c_{j,k}$ son los coeficientes escala, $d_{j,k}$ son los coeficientes wavelet, j_0 indica el espacio vectorial inicial V_{j_0} , el cual será el espacio de menor resolución y 2^N es la longitud de la señal $f(n)$, que es la versión discreta de $f(t)$.

CAPÍTULO 2: FUNDAMENTOS DEL PROCESAMIENTO DIGITAL DE SEÑALES

2.1 SISTEMAS DISCRETOS EN EL TIEMPO

Tal como lo indica su nombre, los sistemas discretos son aquellos que pueden representar la información de manera “segmentada”, o a través de cantidades finitas. Este concepto es de suma importancia pues la información que puede procesar una computadora debe estar necesariamente “discretizada” o “digitalizada”.

2.1.1 Función Delta

La función “delta de Dirac” [RIC1975] está definida por:

$$\delta(t) = 0; t \neq 0 \quad (2.1)$$

$$\delta(t) = \infty; t = 0 .$$

2.1.2 Propiedades de la función Delta

$$\int_{-\infty}^{\infty} f(t) * \delta(t - \tau) dt = f(\tau) \text{ y } \int_{-\infty}^{\infty} \delta(t - \tau) dt = 1 . \quad (2.2)$$

La función $\delta(t)$ es la encargada de pasar una señal desde el dominio continuo (análogo) al dominio discreto (digital).

El proceso de muestreo de una señal continua se lleva a cabo multiplicando ésta por una secuencia infinita de funciones $\delta(t)$, a intervalos de T unidades de tiempo.

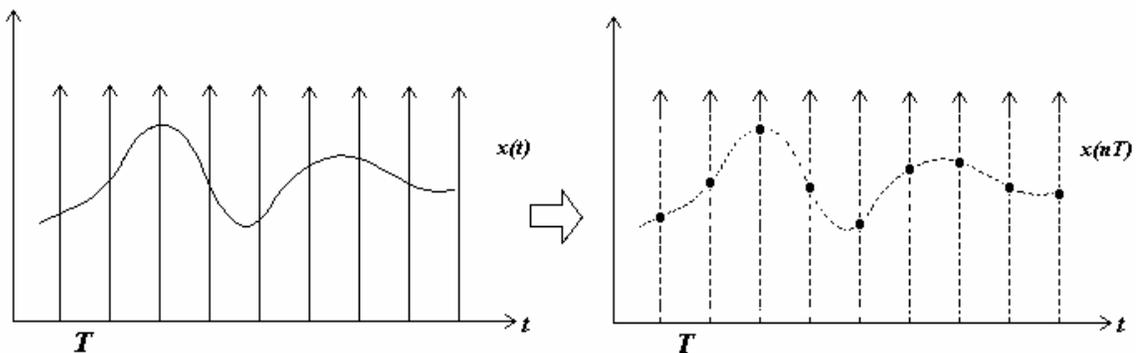


Figura 17. Proceso de muestreo de una señal.

Luego,

$$x_d(t) = \sum_{n=-\infty}^{\infty} x(t) * \delta(t - nT). \quad (2.3)$$

Esta claro que solo interesan los valores de $x(t)$ evaluados para $t = nT$, ya que los demás términos se harán necesariamente cero. Luego tendremos que:

$$x_d(t) = x_s(t) = \sum_{n=-\infty}^{\infty} x(nT) * \delta(t - nT). \quad (2.4)$$

La secuencia de números $x(nT)$ permitirá la representación discreta de la señal de interés. En esta operación debemos tomar en cuenta lo siguiente:

1. Se obtendrán valores finitos de amplitud en puntos discretos del tiempo.
2. El valor de n varía en un rango finito.
3. Sé cuantizará la señal a valores discretos de amplitud, estableciéndose la fidelidad o rango dinámico en función de la precisión matemática del hardware y/o software disponible.

2.1.3 Sistemas Discretos Lineales e Invariantes en el tiempo

Se define un operador DSP (*Digital Signal Processing*) como un operador matemático $T\{ \}$, el cual convierte una secuencia numérica $x(n)$ (entrada) en otra secuencia $y(n)$ (salida):

$$y(n) = T\{x(n)\}.$$

Estos operadores son diseñados para cumplir con el propósito particular de alterar la naturaleza de una secuencia.

2.1.4 Linealidad de los operadores DSP

Consideremos dos secuencias arbitrarias $x_1(n)$ y $x_2(n)$, y sus respectivas salidas: $y_1(n)$ e $y_2(n)$. El operador $T\{ \}$ satisface la condición de linealidad:

$$T\{ax_1(n) + bx_2(n)\} = ay_1(n) + by_2(n),$$

donde a y b son constantes arbitrarias.

2.1.5 Respuesta al impulso

La función delta es un impulso normalizado, esto es, el término cero tiene un valor de uno, mientras que todos los otros términos tienen un valor cero. Por esta razón la función delta es frecuentemente llamada el **impulso unitario**.

Consideremos ahora un sistema dado por el operador $T\{ \}$. Si aplicamos a la entrada de este sistema un *impulso unitario*, la salida del sistema será la secuencia $h(n)$.

$h(n)$:= Respuesta a impulso unitario.

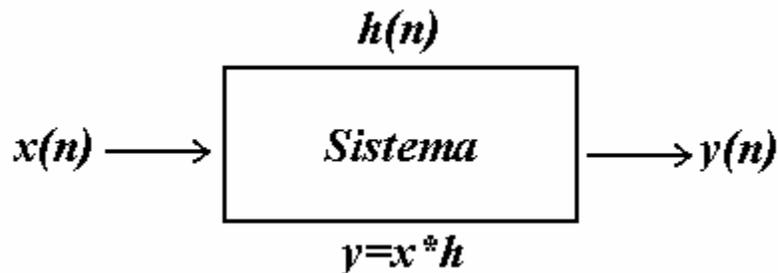


Figura 18. Representación esquemática del operador $T\{ \}$.

Como el nombre sugiere, la respuesta al impulso es la señal que aparece en un sistema cuando una función delta es la entrada. Si dos sistemas lineales son diferentes, entonces tendrán diferentes respuestas al impulso. La respuesta al impulso $h(n)$, es una secuencia cuyos términos no dependen de la secuencia $x(n)$, sino que es una respuesta intrínseca del sistema.

La respuesta al impulso de un sistema es importante, pues si la conocemos *a priori*, podremos saber como reaccionará dicho sistema ante *cualquier* impulso.

2.1.6 Convolución y salida del sistema

2.1.6.1 Convolución

La convolución es un método matemático para combinar dos señales y formar una tercera señal. Esta es una de las técnicas más importantes en DSP. Usando la estrategia de *descomposición de impulso*, los sistemas son descritos por la llamada *respuesta a impulso*. La convolución es importante porque esta involucra las tres señales de interés: la señal de entrada, la señal de salida y la respuesta a impulso.

Técnicamente hablando, la convolución es un operador matemático formal, tal como la multiplicación, adición o integración. La adición toma dos *números* y produce un tercer *número*, mientras que la convolución toma dos *señales* y produce una tercera *señal*. La convolución es usada en diversos campos de las matemáticas, tales como la probabilidad y la estadística.

2.1.7 La función Delta y la respuesta al impulso

Un impulso es una señal compuesta de ceros, excepto por un único punto que es distinto de cero. En efecto, la descomposición en impulsos provee una manera de analizar señales a un término por tiempo. La entrada es descompuesta en simples componentes aditivas, donde cada una de estas componentes son pasadas a través de un sistema lineal y las componentes de salida resultantes son sumadas. La señal resultante de este procedimiento, obtenida a través de la estrategia “Divide y Conquista” es idéntico a aquel que pasa directamente la señal original a través del sistema. Existen varias maneras de descomposición, y sólo dos conforman la espina dorsal del procesamiento de señales: la descomposición de impulso y la descomposición de Fourier. Cuando usemos la descomposición de impulso, el procedimiento será descrito por la **convolución**.

En lo presente usaremos la convolución para señales discretas. Sin embargo, la convolución también puede ser aplicada a señales continuas.

La respuesta $y(n)$ de un operador DSP a un estímulo $x(n)$ resulta de la *convolución discreta* entre la entrada $x(n)$ y la respuesta a impulso $h(n)$ del sistema.

Esta operación se denota como:

$$y[n] = x[n] * h[n], \quad (2.5)$$

Lo cual, en lenguaje matemático significa que:

$$y[n] = \sum_{k=-\infty}^{\infty} h(k)x(n-k). \quad (2.6)$$

El orden en el cual las secuencias sean convolucionadas no importa. La salida del sistema es la misma en cualquier caso. Un operador DSP LTI con entrada $x(n)$ y respuesta a impulso $h(n)$ tendrá la misma salida $y(n)$ que si la entrada es $h(n)$ y la respuesta al impulso es la secuencia $x(n)$.

2.1.8 Invariancia en el tiempo

Si $y(n)$ es la respuesta de un operador DSP a una entrada $x(n)$, e $y(n-k)$ es la respuesta a $x(n-k)$, con k entero, se dice que T es un operador DSP Lineal e Invariante en el Tiempo o **LTI** (Linear-Time-Invariant) [VER1997].

2.1.9 Estabilidad de los operadores DSP LTI

Un operador se dice estable si para cada $x(n) < \infty$ existe $y(n) < \infty$ tal que:

$$y(n) = \left| \sum_{k=-\infty}^{\infty} h(k)x(n-k) \right| < \infty.$$

Si $x(n) \leq M$, donde M es un número real, se cumple que:

$$|y(n)| \leq M \sum_{k=-\infty}^{\infty} h(k).$$

Además, se cumplirá que $|y(n)| < \infty$ sí y sólo sí:

$$\sum_{k=-\infty}^{\infty} |h(k)| < \infty.$$

2.1.9.1 Operadores DSP causales

Un operador DSP LTI se dice que es *causal* si su respuesta al impulso unitario es cero para $n < 0$. Esto es, sí:

$$h(n) = 0, \quad n < 0.$$

Observación:

Una secuencia cualquiera $x(n)$, cuyos términos son iguales a cero para $n < 0$, se dice que es causal, teniendo en cuenta que ésta pudiese corresponder a la respuesta impulso unitario de un operador causal.

2.1.10 Ecuaciones de diferencias

Análogamente a como las ecuaciones diferenciales describen la naturaleza de un problema físico continuo en el tiempo, es posible establecer una ecuación lineal de coeficientes constantes que describa completamente a un operador DSP LTI causal.

Una ecuación de diferencias esta dada por:

$$\sum_{k=0}^N a_k y(n-k) = \sum_{k=0}^M b_k x(n-k). \quad (2.7)$$

Normalizando los coeficientes de la ecuación (2.7) por el valor del coeficiente a_0 , se obtiene la expresión general de la siguiente forma:

$$y(n) + \sum_{k=1}^N a_k y(n-k) = \sum_{r=0}^M b_r x(n-r). \quad (2.8)$$

2.2 LA TRANSFORMADA Z

2.2.1 Introducción

La transformada Z constituye una herramienta que permite tratar ecuaciones de diferencias, de la misma manera que la transformada de Laplace analiza ecuaciones diferenciales en el dominio S . Sin embargo las dos técnicas no son la imagen gemela de la otra; el plano S tiene asociado un sistema de coordenadas rectangulares, mientras que el plano Z un sistema de coordenadas polares.

2.2.2 La naturaleza de la transformada Z

Para recalcar que la transformada Z y la transformada de Laplace son técnicas paralelas, comenzaremos con la transformada de Laplace, y mostraremos como esta puede ser cambiada a la transformada Z . La transformada de Laplace está definida por la relación entre el dominio del tiempo y el dominio s de las señales por:

$$X(s) = \int_{-\infty}^{\infty} x(t)e^{-st} dt. \quad (2.9)$$

La transformada de Laplace puede ser cambiada a la transformada Z en tres pasos. El primero, consiste en cambiar de una señal continua a una discreta. Esto se logra reemplazando la variable de tiempo, t , por un número entero, n , y cambiando la integral por la sumatoria:

$$X(\sigma, \omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-\sigma t} e^{-j\omega t}. \quad (2.10)$$

Aunque ahora tratamos con una señal discreta en el dominio del tiempo, $x(n)$, los parámetros σ y ω todavía pueden tomar un rango de valores continuos. El segundo paso es reescribir el término exponencial. Una señal exponencial puede ser matemáticamente representada de dos maneras:

$$y[n] = e^{-\sigma n} \quad (2.11)$$

$$y[n] = r^{-n}. \quad (2.12)$$

Ambas expresiones generan una curva exponencial. La primera controla el decaimiento de la señal a través del parámetro σ . Si σ es positivo, la forma de onda decaerá en forma discreta. En caso contrario, si σ es negativo, la curva crecerá progresivamente. Si σ tiene el valor cero, $x(n)$ tendrá un valor constante igual a *uno*.

La segunda expresión usa el parámetro r , que controla el decaimiento de la forma de onda. La forma de onda se decrementará si $r > 1$ y se incrementará si $r < 1$. La onda tendrá un valor constante cuando $r = 1$. Las ecuaciones (2.11) y (2.12) son diferentes maneras de expresar la misma idea.

El segundo paso para convertir la transformada de Laplace en la transformada Z es completar usando otra exponencial de la forma:

$$X(r, \omega) = \sum_{n=-\infty}^{\infty} x[n] r^{-n} e^{-j\omega n} \quad (2.13)$$

Aunque esta es una correcta expresión de la transformada Z , no está escrita en su forma más compacta para la notación compleja. Este problema fue superado en la transformada de Laplace, introduciendo una nueva variable compleja, s , definida como $s = \sigma + j\omega$. De manera similar, definiremos una nueva variable para la transformada Z :

$$Z = r e^{j\omega} . \quad (2.14)$$

Esta es la definición de la variable compleja, z , en la notación polar, como la combinación de dos variables, r y ω . Esto produce la forma estándar de la transformada Z :

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n} . \quad (2.15)$$

2.2.3 Definición de la Transformada Z

Para un sistema discreto o digital $x(n)$, se define la **Transformada Z** por:

$$Z\{x(n)\} = \sum_{n=-\infty}^{\infty} x[n] z^{-n} ,$$

donde Z , simboliza a la transformada Z y z es la variable compleja del plano bidimensional en donde Z está definida.

Observaciones:

La cantidad compleja z , generaliza el concepto de frecuencia al dominio complejo. De hecho, la respuesta en frecuencia en el dominio z es hallada a lo largo del círculo unitario. Esto puede ser visto evaluando en la transformada z , $r=1$, con lo que la transformada Z se reducirá a la transformada discreta de Fourier (DTFT).

El valor de z^{-1} corresponde al *retardo unitario*.

Ya que $X(z) = Z\{x(n)\}$ corresponde a una serie de potencias, podría no converger para todo z . Los valores de z para los cuales $X(z)$ converge definen la región de convergencia (ROC).

Toda $X(z)$ lleva asociada una ROC, pues podría ocurrir que dos secuencias distintas produzcan una $X(z)$ idéntica con diferente ROC.

La figura 19 ilustra la diferencia que hay entre la transformada de Laplace en el plano s , y la transformada Z en el plano z . Un punto en el plano S esta identificado por dos parámetros: el decaimiento exponencial σ , que es la variable a lo largo del eje horizontal, y la frecuencia ω , que es la variable a lo largo del eje vertical. En otras palabras, estas dos variables están definidas en un sistema de coordenadas rectangulares. Esta geometría resulta de la definición de s , la variable compleja representando la posición en el plano s , a través de la relación $s = \sigma + j\omega$.

En comparación, el dominio z usa las variables r y ω , las cuales forman un sistema de coordenadas polares. La distancia desde el origen r , es el valor del decaimiento exponencial. La distancia angular medida desde el eje horizontal positivo ω , es la frecuencia. Esta geometría resulta de la definición de z por $z = re^{j\omega}$. En otras palabras, la variable compleja que representa la posición en el plano z esta formada por la combinación de dos parámetros reales en la forma polar.

De estas diferencias resultan *líneas* verticales en el plano s , lo que corresponderá *círculos* unitarios en el plano z . Por ejemplo, el plano s de la figura 19 muestra un patrón polo-cero, donde los polos y ceros corresponden a líneas verticales. Los equivalentes polos y ceros en el plano z corresponden a círculos concéntricos con respecto al origen.

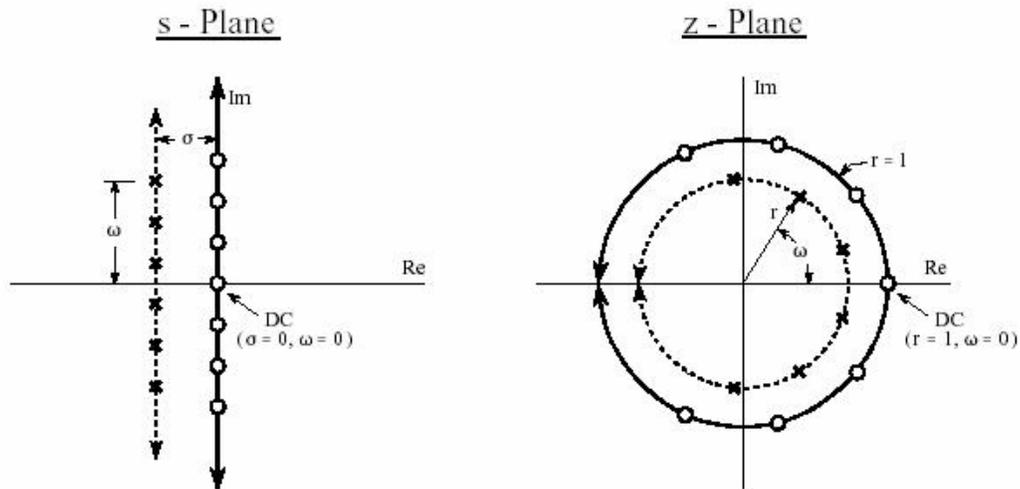


Figura 19. Geometrías de la Transformada de Laplace y la Transformada Z .

Las líneas verticales en la mitad izquierda del plano s corresponden a círculos dentro del círculo unitario en el plano z . De igual manera, las líneas verticales en la mitad derecha del plano s corresponden a círculos fuera del círculo unitario en el plano z . En otras palabras, los lados izquierdo y derecho del plano s corresponden al interior y exterior del círculo unitario, respectivamente.

Esta geometría circular también corresponde al espectro de frecuencia de la señal discreta, la cual es periódica. Esto es, cuando ω es incrementado en π , son hallados los mismos valores entre 0 y π .

Es importante notar como la variable ω , es utilizada en las dos transformaciones. Una senoide *continua* puede tener una frecuencia entre DC e infinito. Esto significa que el plano s puede permitir a ω recorrer desde menos infinito hasta más infinito. En comparación, una senoide *discreta* solo puede tener una frecuencia entre DC y la mitad de la razón de muestreo. Esto es, la frecuencia debe estar entre 0 y 0.5 cuando es expresada como una fracción de la frecuencia de muestreo, o entre 0 y π cuando es expresada como una frecuencia natural.

En el plano z , las frecuencias positivas corresponden a los ángulos entre 0 y π radianes, mientras que las frecuencias negativas corresponden de 0 a $-\pi$ radianes.

2.2.4 Propiedades de la Transformada Z

$$Z\{ax(n)+by(n)\} = aX(z)+bY(z)$$

$$Z\{x(n-1)\} = z^{-1}X(z)+x(-1)$$

$$Z\{x(n-N)\} = z^{-N}X(z)+z^{-(N-1)}x(-1)+\dots+x(-N)$$

$$Z\{x(n+N)\} = z^N X(z)-z^N x(0)-z^{N-1}x(1)-\dots-zx(N-1)$$

$$Z\{a^n x(n)\} = X(z/a)$$

$$x(n)*y(n) = X(z)Y(z) \quad (\text{Propiedad de Convención})$$

2.3 Sistemas DSP LTI y la Transformada Z

De la ecuación anterior, y haciendo $k=m$ y $r=k$, y aplicando la transformada Z, tenemos:

$$Z\left\{y(n) + \sum_{m=0}^{\infty} a_m y(n-m)\right\} = Z\left\{\sum_{k=0}^{\infty} b_k x(n-k)\right\}. \quad (2.12)$$

Utilizando las propiedades de retardo y linealidad, tendremos que:

$$Y(z)\left\{1 + \sum_{m=1}^{\infty} a_m z^{-m}\right\} = \sum_{k=0}^{\infty} b_k z^{-k} X(z). \quad (2.13)$$

Reagrupando términos en la última ecuación, nos queda lo que se conoce como *Función de Transferencia* del filtro, la cual, quedará en términos de la variable compleja z como:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{\infty} b_k z^{-k}}{1 + \sum_{m=1}^{\infty} a_m z^{-m}}. \quad (2.14)$$

2.3.1 Polos y Ceros de $H(z)$

La ecuación de transferencia puede reescribirse de la siguiente manera:

$$H(z) = \frac{\sum_{k=0}^{\infty} b_k z^{-k}}{1 + \sum_{m=0}^{\infty} a_m z^{-m}}, \text{ con } a_0=1. \quad (2.15)$$

Multiplicando y dividiendo por $z^{(N-M)}$ nos queda:

$$H(z) = z^{N-M} \frac{\sum_{k=0}^{\infty} b_k z^{-k+M}}{\sum_{m=0}^{\infty} a_m z^{-m+N}}. \quad (2.16)$$

Suponiendo que $H(z)$ ha sido expresada en su forma irreducible, entonces podemos decir que:

$$H(z) = (a_0/b_0) z^{N-M} \frac{\prod_{k=1}^M (z - \varepsilon_k)}{\prod_{m=1}^N (z - \lambda_m)}. \quad (2.17)$$

El conjunto de valores ε_k para los cuales $H(\varepsilon_k)=0$ se conocen como **ceros** de $H(z)$. A su vez, los valores λ_m para los cuales $H(\lambda_m) \rightarrow \infty$ se conocen como **polos** de $H(z)$.

Para encontrar los ceros $H(z)$ basta determinar las raíces del polinomio numerador, mientras que los polos de $H(z)$ se encuentran al determinar las raíces del polinomio denominador. Además, existe un polo o cero adicional en $z=0$. Este será polo o cero dependiendo de si $M>N$ o $N>M$.

Observación.

Un sistema continuo es inestable cuando los polos ocupan la izquierda y derecha del plano s . De manera similar, un sistema discreto es inestable cuando los polos están *fuera* del círculo unitario en el plano z . Cuando la señal en el dominio del tiempo es completamente real (que es el caso más común), las mitades superior e inferior del plano z son imágenes especulares una de la otra.

CAPÍTULO 3: FILTROS

3.1 Introducción

El término *Filtro* es comúnmente usado para describir un dispositivo que discrimina información, acorde a las características que se le piden a la señal de entrada que pasa a través de este. Por ejemplo, en el caso de un filtro de aire, este discriminará el polvo y las impurezas del aire "no deseado" que ingresa por la entrada del filtro, para obtener un aire "puro" en la salida.

El término *filtro digital* lo entenderemos como cualquier procesamiento realizado en una señal digital. Técnicamente hablando, un filtro digital es la implementación en hardware o software de una ecuación de diferencias.

Un sistema LTI también realiza un tipo de discriminación o filtrado entre las diferentes componentes de frecuencia de la entrada. Por lo mismo, todas las consideraciones que se han hecho hasta ahora sobre los sistemas DSP LTI son igualmente válidas para un filtro digital, por lo que se hablará de ahora en adelante de filtro digital.

3.2 Ventajas de los filtros digitales

1. Alta inmunidad al ruido
2. Alta precisión (limitada por los errores de redondeo en la aritmética empleada)
3. Fácil modificación de las características del filtro
4. Muy bajo costo

Por estas razones, los filtros digitales están reemplazando rápidamente a los filtros analógicos.

Los filtros digitales son usados para dos propósitos generales: (1) separación de señales que han sido combinadas, y (2) restauración de señales que han sido distorsionadas de alguna manera. Los filtros analógicos pueden ser usados para estas tareas, sin embargo, los filtros digitales pueden lograr resultados muy superiores.

El camino más directo para implementar un filtro digital es a través de la convolución de la señal de entrada con la respuesta al impulso del filtro digital. Todos los filtros lineales pueden ser diseñados de esta manera. Cuando se usa la respuesta a impulso para este fin, el filtro diseñado tiene un nombre especial: **filtro Kernel**. En particular, esta tesis tratará con los filtros que pueden ser implementados a través de una convolución.

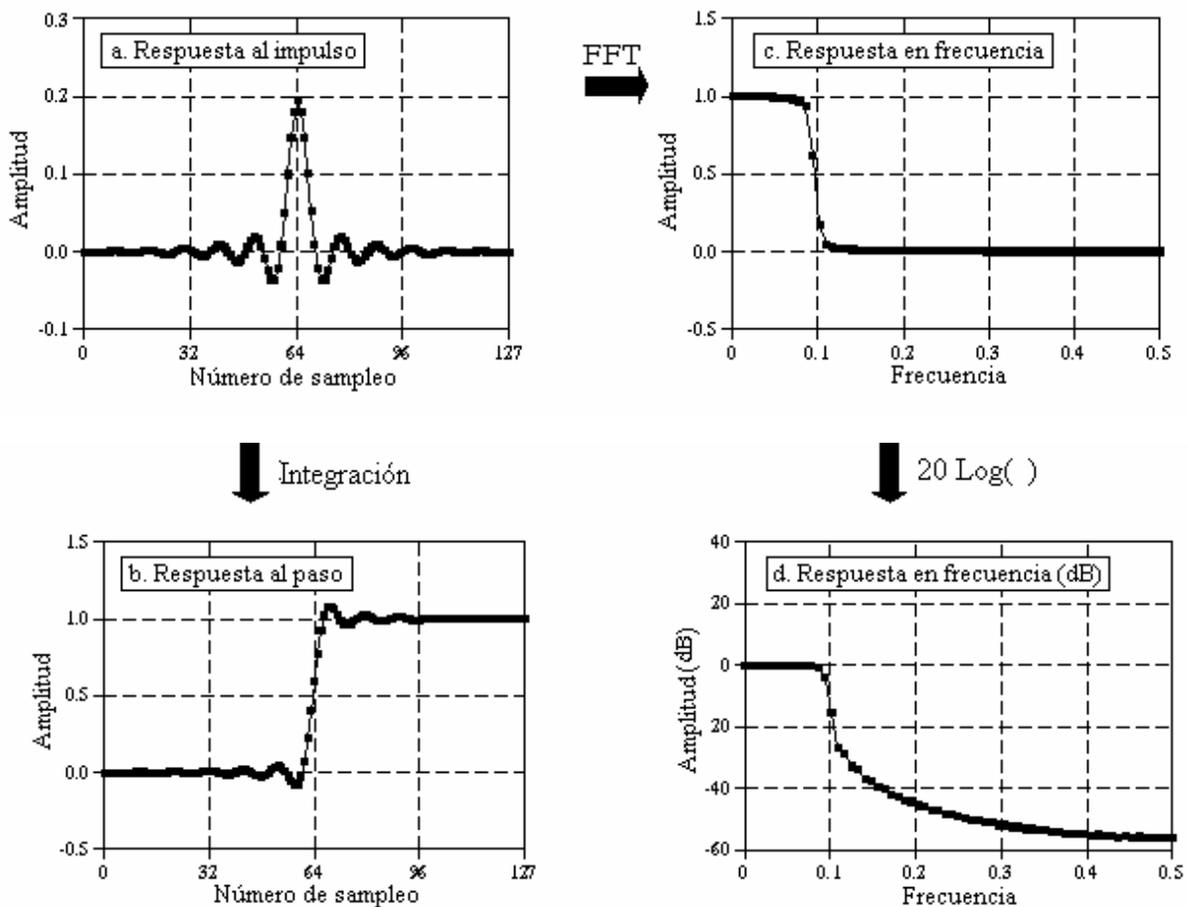
También existe otro camino para implementar filtros digitales llamado **recursión**. Cuando un filtro es implementado por convolución, cada término de la salida es calculado escalando los términos de la entrada para después sumar todas las contribuciones. Los filtros recursivos son una extensión de estos, usando los valores previamente calculados de la salida, además de los valores de entrada.

En vez de usar un filtro Kernel, los filtros recursivos son definidos usando un set de coeficientes de recursión. De las diferencias de estos dos caminos hablaremos más adelante, clasificando a los filtros en dos grandes familias, según su respuesta al impulso.

3.3 Respuestas de un filtro

Cualquier filtro digital o analógico esta caracterizado por las siguientes respuestas:

1. Respuesta al impulso.
2. Respuesta al paso.
3. Respuesta en frecuencia.



La figura 20 muestra las tres respuestas mencionadas antes. Cada una de estas respuestas contiene información completa acerca del filtro, pero de diferente manera. Si uno de estos tres es especificado, los otros dos quedan fijos y pueden ser directamente especificados. Las tres representaciones son igualmente importantes porque ellas describen como reaccionará el filtro bajo diferentes circunstancias.

3.3.1 Respuesta al Impulso

Cualquier filtro lineal tiene una respuesta al impulso, la cual puede ser usada para diseñar un filtro digital. La *respuesta al impulso* de un filtro, que está definida por la secuencia $h(n)$, corresponde a la salida del sistema cuando se ingresa en la entrada un impulso unitario.

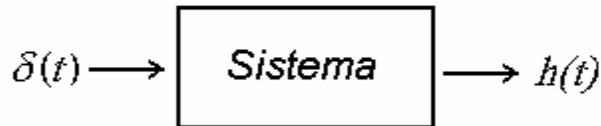


Figura 21: Esquema que ilustra la obtención de la respuesta al impulso de un sistema o filtro.

3.3.2 Respuesta al paso

Del mismo modo en que la respuesta al impulso es la salida de un sistema cuando la entrada es un impulso, la **respuesta al paso** corresponde a la salida del sistema cuando la entrada es un paso (o escalón). Puesto que el paso es la integral del impulso [SMI1997], la respuesta al paso es la integral de la respuesta al impulso. Esto nos provee de dos caminos para hallar la respuesta al paso:

1. Ingresar una forma de onda paso (escalón) a la entrada, y observar la salida.
2. Integración de la respuesta al impulso.

3.3.3 Respuesta en frecuencia

La respuesta en frecuencia del filtro puede ser hallada tomando la transformada discreta de Fourier (DFT) [FAF2000, PRO1996] o aplicando el algoritmo FFT a la respuesta al impulso.

Debemos mencionar que la respuesta al paso describe cómo la información representada en el dominio del tiempo es modificada por el sistema, mientras que la respuesta en frecuencia muestra como se modifica la información en el dominio de la frecuencia. Esta distinción es sumamente importante en el diseño de filtros porque no es posible optimizar un filtro para ambas aplicaciones. De un buen rendimiento en el dominio del tiempo resulta un mal rendimiento en el dominio de la frecuencia y viceversa.

3.4 Parámetros de un filtro

3.4.1 Parámetros en el dominio del tiempo

Vale la pena mencionar que las respuestas al impulso, al paso y en frecuencia contienen la misma información, sólo que de maneras diferentes. Para los casos en que se requiera el análisis en el dominio del tiempo se usará la respuesta al paso.

Los parámetros para evaluar el rendimiento en el dominio del tiempo son estos:

- *Velocidad de transición* (risetime)
- *Overshoot*
- Linealidad de fase (simetría entre las mitades superior e inferior de un paso)

Para distinguir cambios en una señal, la duración de la respuesta al paso debe ser breve, de tal manera que se puedan distinguir los cambios o fluctuaciones. Esto indica que la respuesta al paso debe ser *fast* (significa *veloz* en la jerga DSP). La figura 22 muestra lo recién mencionado. La manera más común de especificar la velocidad de transición es considerar el número de términos entre el 10% y el 90% de nivel de amplitud máxima. Sin embargo, no siempre es posible obtener una rápida transición. Esto se debe a varias razones: reducción de ruido, limitaciones inherentes a la adquisición de datos del sistema, aliasing, etc.

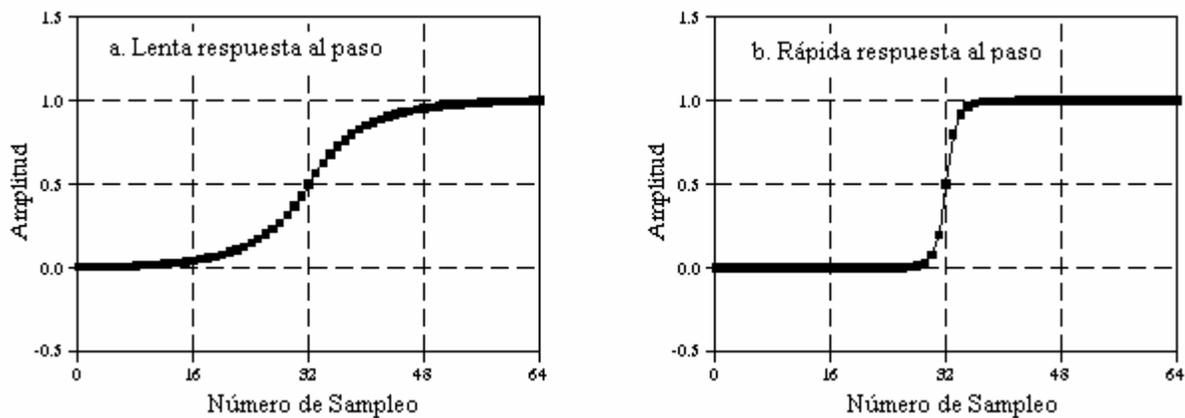


Figura 22. Respuesta al paso.

Las figura 23 muestra el siguiente parámetro: El **overshoot** en la respuesta al paso. El overshoot corresponde a una fluctuación en la respuesta al paso del filtro y generalmente debe ser eliminado, pues este cambia la amplitud de los términos de la señal. Esto es una distorsión básica de la información contenida en el tiempo.

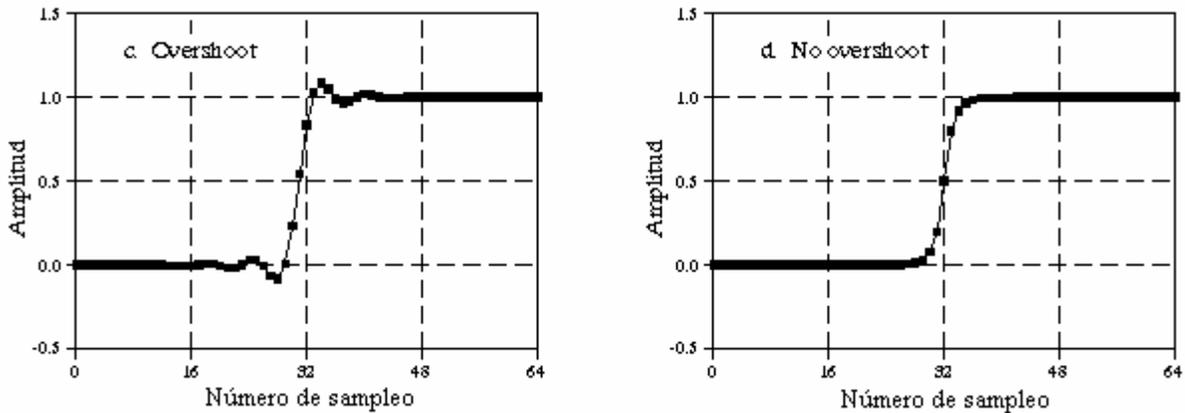


Figura 23. Segundo parámetro: Overshoot

Por último, con frecuencia se desea que la mitad superior de la respuesta al paso sea simétrica con la mitad inferior, como se ilustra en la figura 34. Esta simetría es necesaria para hacer que la forma en que se asciende por el borde sea igual a la forma de descenso y es llamada **fase lineal**.

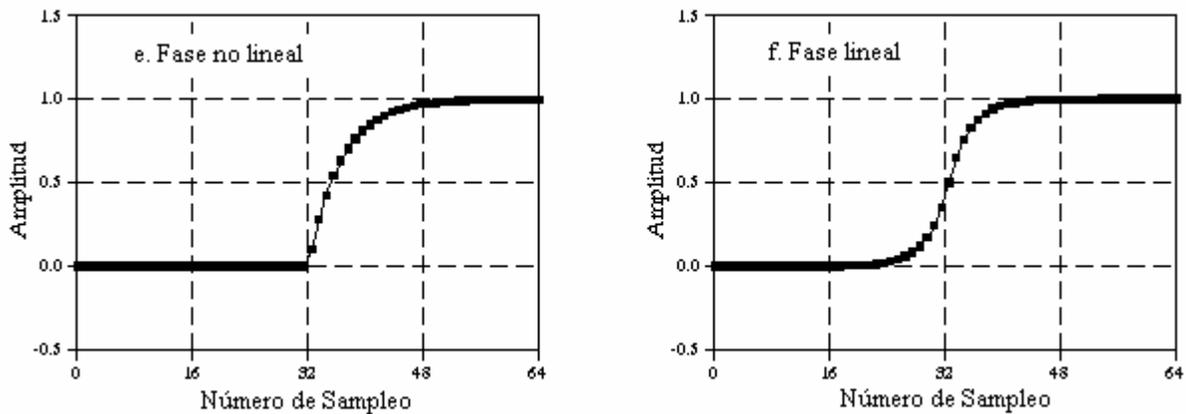


Figura 24: Linealidad de fase.

3.4.2 Parámetros en el dominio de la frecuencia

3.4.2.1 Respuestas en frecuencia

La figura 25 muestra las cuatro respuestas básicas en frecuencia.

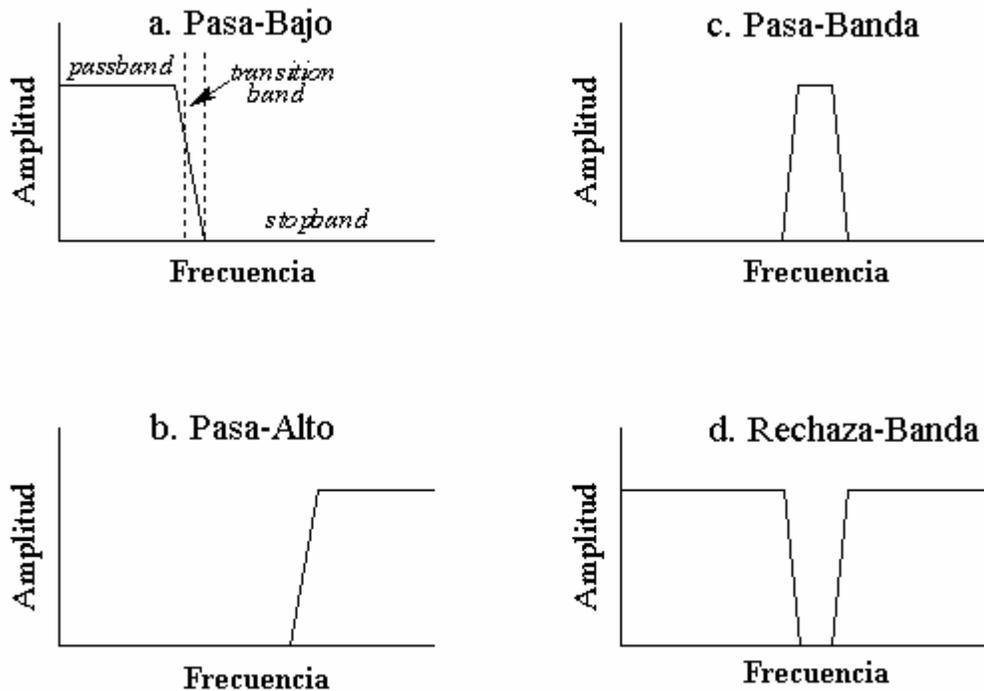


Figura 25. Las cuatro respuestas básicas en frecuencia.

El propósito de estos filtros es permitir el paso a algunas frecuencias sin ser alteradas, mientras que otras frecuencias son totalmente bloqueadas. La banda de paso está referida a aquellas frecuencias que son consideradas, mientras que la banda de parada contiene todas aquellas frecuencias que son bloqueadas. La banda de transición está al medio de estas dos.

3.4.2.2 Parámetros

El primer parámetro de un filtro en el dominio de la frecuencia es el **roll-off**, y está asociado directamente a la pendiente de decaimiento entre la banda de paso y la banda de parada. Un rápido roll-off significa que la banda de transición es muy estrecha.

La división entre la banda de paso y la banda de transición es llamada la **frecuencia de corte**, que corresponde al segundo parámetro del filtro. En el diseño de filtros analógicos, la frecuencia de corte es usualmente definida donde la amplitud se reduce a 0.707 (esto es, -3dB) [SMI1997]. Los filtros digitales son menos estandarizados, y es común ver 99%, 90%, 70.7% y 50% de nivel de amplitud para la frecuencia de corte.

Para que las frecuencias que pasan a través de la banda de paso no sean alteradas por el filtro, este no debe tener **ripple de pasa-banda**, como se muestra en la figura 26. Finalmente, para un adecuado bloqueo de las frecuencias en la banda de paso, es necesario tener una muy buena **atenuación en la banda de parada**, como se ve en (e) y (f).

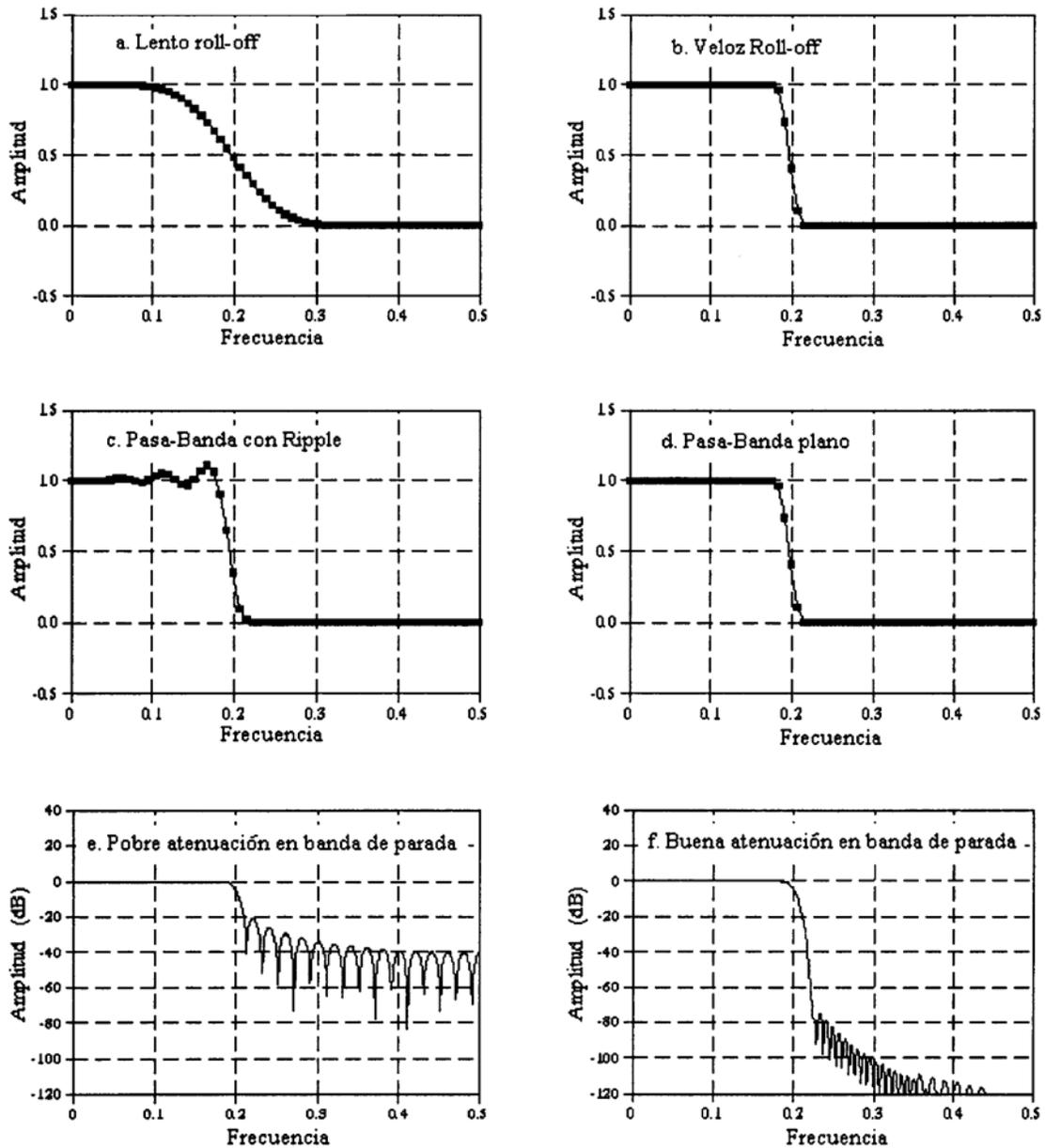


Figura 26: Parámetros para evaluar el rendimiento en el dominio de la frecuencia. Las respuestas en frecuencia mostradas son para un filtro pasa-bajo. Tres parámetros son importantes: (1) la forma del roll-off, (2) el ripple de pasa-banda y (3) la banda de atenuación.

3.5 CLASIFICACIÓN DE LOS FILTROS DIGITALES

Los filtros digitales pueden ser clasificados según la duración de su respuesta al impulso. Como se mencionó antes, los filtros pueden ser implementados por dos caminos: convolución y recursión. Cuando se implementa un filtro por el método recursivo, su respuesta a impulso está compuesta de sinusoides que decaen exponencialmente en amplitud. Esto hace que la respuesta impulso sea infinitamente larga. Sin embargo, la amplitud eventualmente decae hasta el nivel del ruido de fondo del sistema, por lo que los términos remanentes pueden ser ignorados. Debido a estas características, los filtros recursivos son también llamados filtros **de respuesta a impulso infinita** o **filtros IIR** (*Infinite Impulse Response*). En comparación, los filtros que son creados por convolución, son **de respuesta a impulso finita** o son llamados **filtros FIR** (*Finite Impulse Response*).

3.5.1 Características de los filtros FIR e IIR

En ciertas ocasiones se prefiere el uso de filtros FIR al uso de filtros IIR, debido a que presentan algunas ventajas en cuanto a diseño e implementación con aritmética finita. Aquí tenemos algunas de ellas:

- Los filtros FIR tienen una fase perfectamente lineal y son siempre estables.
- La implementación en Hardware de un filtro FIR es mucho más sencilla que la de un filtro IIR.
- Existen eficientes estructuras matemáticas para su implementación, incluyendo formas no recursivas y recursivas.
- Pueden ser diseñados para cumplir especificaciones arbitrarias (varios pasa-banda y rechaza-banda con diferentes atenuaciones en distintas regiones)
- Pueden ser implementados con aritmética finita de bajo orden (por ejemplo, 8 bits)

Los filtros IIR, al presentar distorsión de fase producen una respuesta de fase no lineal. Además, la presencia de polos en estos filtros produce inestabilidad. Sin embargo, los filtros IIR tienen una gran ventaja: al tener polos y ceros se requiere un orden muchísimo menor del filtro para una misma aplicación que un filtro FIR.

CAPÍTULO 4: DISEÑO DE FILTROS DIGITALES

4.1 PROCESO DE DISEÑO DE FILTROS DIGITALES

El proceso de diseño de cualquier filtro digital requiere de los siguientes pasos:

1. Establecer las especificaciones del filtro para ciertos requerimientos. Estos son los mismos que los necesarios para un filtro analógico: frecuencias de parabanada y pasa-banda, atenuaciones, ganancia DC, frecuencia de corte y otras.
2. Establecer un margen de error aceptable en relación a la respuesta encontrada y la respuesta deseada para el filtro digital.
3. Determinar la función de transferencia que cumpla con las especificaciones.
4. Realizar la función de transferencia en Hardware o Software

4.1.1 Especificaciones del filtro digital

Por lo general, las especificaciones que debe cumplir un filtro digital están dadas en función de una respuesta en frecuencia, determinando así las regiones en dónde las componentes espectrales de la señal de entrada al sistema deben conservarse y dónde éstas deben rechazarse.

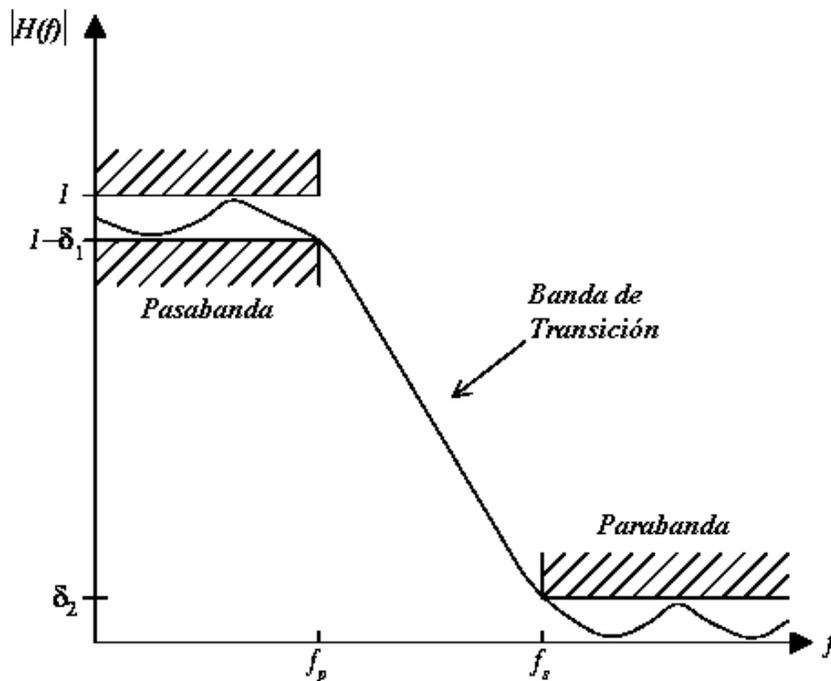


Figura 27: Especificaciones principales de un filtro digital

Puesto que la función respuesta en frecuencia se obtiene al evaluar $H(z)$ sobre el círculo unitario $z = e^{j\Omega}$, las especificaciones se dan en función de la frecuencia en radianes entre 0 y π , en donde π se hace corresponder a la mitad de la frecuencia de muestreo (en Hertz) con la cual las formas de onda continuas en tiempo son muestreadas. Tales requerimientos u especificaciones solo pueden satisfacerse por medio de un filtro pasa-bajos ideal, el cual es imposible de implementar con aritmética finita. Así pues, se hace necesario considerar una respuesta en frecuencia **realizable**, como la mostrada en la figura 27, que se aproxime a la respuesta ideal.

4.2 Análisis en Z de los filtros digitales

Recordemos la ecuación de diferencias para un sistema LTI:

$$\sum_{k=0}^N a_k y(n-k) = \sum_{k=0}^M b_k x(n-k),$$

donde a y b son los coeficientes que describen el sistema o, dado el caso, el filtro. El uso de esta ecuación describe como se puede implementar este filtro.

Si aplicamos la transformada Z a la ecuación de diferencias, obtendremos la función de transferencia del sistema, dado por:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{\infty} b_k z^{-k}}{1 + \sum_{m=1}^{\infty} a_m z^{-m}}. \quad (4.1)$$

Esta forma de expresar la función de transferencia es importante porque contiene de manera directa los coeficientes del filtro.

Observaciones.

Debe tenerse especial cuidado con la convención de signos de los coeficientes. En algunos textos se utilizan signos menos para “ a ”. En este caso serán los coeficientes “ a ” con signo opuesto. Si no se observa este detalle, el filtro puede volverse inestable.

Otro detalle es que, como el filtro tiene un número finito de coeficientes, entonces es posible hacer que el signo de los exponentes de las potencias sean positivos en vez de negativos. Sin embargo, notar que no es posible determinar por que potencia amplificar la función de transferencia si no se conoce *a priori* cual será el número de coeficientes del filtro que se utilizarán.

La función de transferencia de un filtro es importante porque esta es manejable en métodos donde los coeficientes del filtro no pueden hacerlo de manera directa, o es muy complicada su manipulación. Esto incluye tareas como la combinación de etapas cascada y paralelo en un único sistema, diseño de filtros especificando la localización de polos y ceros, conversión de filtros análogos en filtros digitales, etc.

4.2.1 Expresión de $H(z)$ en términos de sus polos y ceros

La estrategia de la transformada Z estriba en probar la respuesta al impulso con sinusoides y exponenciales para encontrar el sistema de polos y ceros.

Mientras que el método anterior involucra una tremenda cantidad de álgebra, expresar la función de transferencia en términos de sus polos y ceros es sencillo y directo, y puede ser fácilmente implementado en una computadora. La representación polo-cero está dada por:

$$H(z) = (a_0/b_0)z^{N-M} \frac{\prod_{k=1}^M (z - \varepsilon_k)}{\prod_{m=1}^N (z - \lambda_m)}, \quad (4.2)$$

La dificultad para pasar de la ecuación (4.1) a la ecuación (4.2) es considerable, pues requiere una factorización de orden arbitrario. Como no es siempre posible usar métodos algebraicos para factorizar polinomios de elevado orden, deben usarse métodos numéricos. Afortunadamente, esto no será necesario porque el diseño de filtros digitales comienza con la localización *polo-cero* y termina con los coeficientes del filtro.

Como se trata de números complejos, los polos y ceros pueden ser representados en la forma polar o rectangular. La notación polar tiene la ventaja de ser más consistente con la organización natural del plano z . En cambio, el formato rectangular es preferido para el trabajo matemático, pues es más fácil de manipular $\sigma + j\omega$ que $re^{j\omega}$.

Así pues podemos implementar un filtro digital usando los siguientes pasos:

- Especificando la localización de polos y ceros en el plano Z .
- Escribir la función de transferencia en el formato polo-cero.
- Pasar a la función de transferencia en el formato polinomial.
- De este último, identificar los coeficientes del filtro.

Para hallar la respuesta en frecuencia del filtro implementado existen dos métodos:

Un método matemático, el cual está basado en hallar los valores en el plano z que yacen en el círculo unitario. Esto se hace evaluando la función de transferencia para $r=1$. Para ello es necesario escribir la función de transferencia en las formas mencionadas por las ecuaciones (4.1) y (4.2).

Esto provee una ecuación matemática de la respuesta en frecuencia $H(\omega)$. El problema está en que la ecuación resultante requiere una tremenda cantidad de álgebra para obtener alguna organización acerca de la magnitud y la fase.

El segundo método para encontrar la respuesta en frecuencia también usa el hecho de evaluar el plano z en el círculo unitario. La diferencia es que este calcula solo términos aislados de la respuesta en frecuencia y no la solución matemática entera de la curva. La magnitud y fase de la respuesta en frecuencia son halladas para cada una de estas localizaciones evaluando la función de transferencia. Este sistema en general trabaja bien. Sin embargo, la mayor desventaja estriba en que no es posible visualizar como el *ruido de round-off* afecta las características del sistema, de manera que la respuesta en frecuencia puede verse muy bien pero la implementación del sistema puede ser completamente inestable.

El método más seguro parece ser el siguiente: hallar la respuesta en frecuencia desde los coeficientes del filtro que son usados actualmente para implementar el filtro. Para comenzar, hallaremos la respuesta a impulso pasando un impulso unitario a través del sistema. En el segundo paso, tomaremos la DFT de la respuesta a impulso (usando por supuesto, la FFT) para hallar la respuesta en frecuencia del sistema. El único inconveniente del método es que deben tomarse bastantes términos de la respuesta al impulso para que los términos descartados sean insignificantes en su efecto.

4.3. Descripción matemática del diseño de filtros FIR

Un filtro FIR de longitud M con entrada $x(n)$ y salida $y(n)$ se describe mediante la siguiente ecuación de diferencias:

$$y(n) = b_0x(n) + b_1x(n-1) + \dots + b_{M-1}x(n-M+1) = \sum_{k=0}^{M-1} b_kx(n-k), \quad (4.3)$$

donde b_k son los coeficientes del filtro.

Por otro lado podemos expresar la salida del filtro $y(n)$ como una convolución de la entrada $x(n)$ con la respuesta al impulso del filtro $h(n)$:

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k). \quad (4.4)$$

Como estas dos ecuaciones son idénticas, entonces se cumple que $b_k=h(k)$.

Se puede demostrar que la respuesta de un filtro FIR es de fase lineal si los coeficientes $h(n)$ cumplen con [PRO1996]:

$$h(n) = \pm h(M-1-n) \quad n = 0, 1, \dots, M-1. \quad (4.5)$$

La función de transferencia Z del filtro FIR, aplicando esta condición es:

$$\begin{aligned}
 H(z) &= \sum_{k=0}^{M-1} h(k)z^{-k} = h(0) + h(1)z^{-1} + h(2)z^{-2} + \dots + h(M-2)z^{-(M-2)} + h(M-1)z^{-(M-1)} \\
 &= \begin{cases} z^{-(M-1)/2} \left\{ h\left(\frac{M-1}{2}\right) + \sum_{k=0}^{(M-3)/2} h(k) \left(z^{(M-1-2k)/2} \pm z^{-(M-1-2k)/2} \right) \right\} & \text{Para } M \text{ impar} \\ z^{-(M-1)/2} \sum_{k=0}^{(M-3)/2} h(k) \left(z^{(M-1-2k)/2} \pm z^{-(M-1-2k)/2} \right) & \text{Para } M \text{ par} \end{cases}
 \end{aligned} \tag{4.6}$$

De esta última expresión se deduce que:

$$z^{-(M-1)} H(z^{-1}) = \pm H(z), \tag{4.7}$$

lo que significa que las raíces de $H(z)$ son las mismas que las de $H(z^{-1})$.

Es decir, las raíces (en este caso los ceros) ocurren en pares recíprocos. Si z_1 es un cero de $H(z)$, $\frac{1}{z_1}$ también es un cero de $H(z)$. Además, si z_1 es un cero complejo, el conjugado z_1^* también lo es, así como $1/z_1^*$, como se indica en la figura 28.

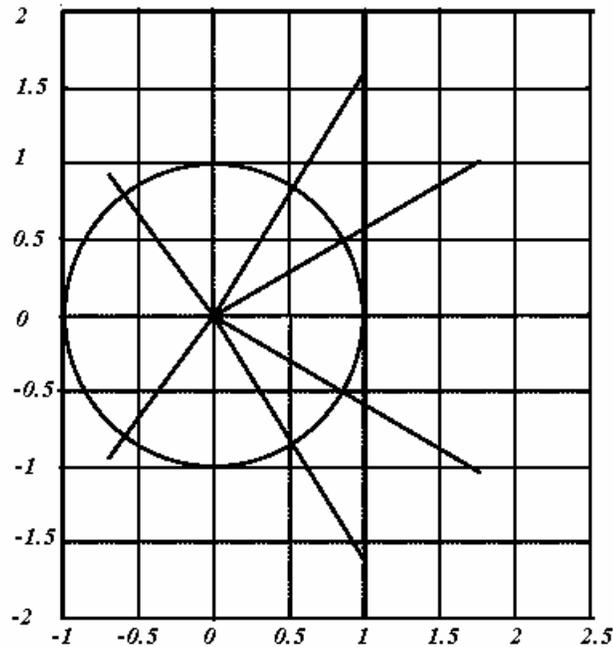


Figura 28: Simetría de ceros en $H(z)$ para un filtro FIR.

Las características de la respuesta en frecuencia de la fase lineal de los filtros FIR son obtenidas evaluando $H(z)$ en el círculo unitario. Tal sustitución nos entregará $H(\omega)$.

4.3.1 Simetría y antisimetría en la fase lineal

4.3.1.1 Caso simétrico

Cuando $h(n) = h(M-1-n)$, $H(\omega)$ puede ser expresado como:

$$H(\omega) = H_r(\omega)e^{-j\omega(M-1)/2}, \quad (4.8)$$

donde $H_r(\omega)$ es una función real de ω y puede ser expresada como:

$$H_r(\omega) = h\left(\frac{M-1}{2}\right) + 2 \sum_{n=0}^{(M-3)/2} h(n) \cos \omega \left(\frac{M-1}{2} - n\right), \text{ si } M \text{ es impar} \quad (4.9)$$

$$H_r(\omega) = 2 \sum_{n=0}^{(M/2)-1} h(n) \cos \omega \left(\frac{M-1}{2} - n\right), \text{ Si } M \text{ es par.} \quad (4.10)$$

La fase característica del filtro para ambos casos de M es:

$$\Theta(\omega) = -\omega \left(\frac{M-1}{2}\right), \quad \text{Si } H_r(\omega) > 0 \quad (4.11)$$

$$\Theta(\omega) = -\omega \left(\frac{M-1}{2}\right) + \pi, \quad \text{Si } H_r(\omega) < 0. \quad (4.12)$$

4.3.1.2 Caso antisimétrico

Cuando $h(n) = -h(M-1-n)$, entonces la respuesta al impulso unitario es *antisimétrica*. Para M impar, el punto central de $h(n)$ es $n=(M-1)/2$. Consecuentemente,

$$h\left(\frac{M-1}{2}\right) = 0.$$

Sin embargo, cuando M es par, cada término de $h(n)$ tiene su opuesto idéntico, pero con signo contrario.

Una manera directa de representar la respuesta en frecuencia de un filtro FIR con una respuesta a impulso antisimétrica es expresándola como:

$$H(\omega) = H_r(\omega)e^{j[-\omega(M-1)/2 + \pi/2]}, \quad (4.13)$$

donde

$$H_r(\omega) = 2 \sum_{n=0}^{(M-3)/2} h(n) \operatorname{sen} \omega \left(\frac{M-1}{2} - n \right), \text{ si } M \text{ es impar} \quad (4.14)$$

$$H_r(\omega) = 2 \sum_{n=0}^{(M/2)-1} h(n) \sin \omega \left(\frac{M-1}{2} - n \right), \text{ Si } M \text{ es par.} \quad (4.15)$$

La fase característica del filtro para ambos casos de M es:

$$\Theta(\omega) = \frac{\pi}{2} - \omega \left(\frac{M-1}{2} \right), \quad \text{Si } H_r(\omega) > 0 \quad (4.16)$$

$$\Theta(\omega) = \frac{3\pi}{2} - \omega \left(\frac{M-1}{2} \right) + \pi, \quad \text{Si } H_r(\omega) < 0. \quad (4.17)$$

Estas fórmulas generales de respuesta en frecuencia pueden ser usadas para diseñar la fase lineal de filtros FIR con respuestas a impulso simétrica y antisimétrica.

Escoger una respuesta al impulso unitaria simétrica o antisimétrica depende de la aplicación del filtro. Por ejemplo, si $h(n) = -h(M-1-n)$ y M es impar, esto implica que $H_r(0) = 0$ y $H_r(\pi) = 0$. Consecuentemente, esto significa que no es posible diseñar un pasa-alto o un pasa-bajo con esta especificación [PRO1996]. Similarmente, de una respuesta al impulso unitaria antisimétrica con M par también resulta $H_r(0) = 0$. Consecuentemente, no podemos usar la condición antisimétrica en el diseño de un filtro pasa-bajo FIR de fase lineal.

En resumen, el problema del diseño de un filtro FIR consiste en determinar los M coeficientes de $h(n)$ para una especificación de una respuesta en frecuencia deseada.

4.4 TÉCNICAS ESPECIALES DE FILTRADO FIR

4.4.1 La convolución como técnica de filtrado

La manera más directa de implementar un filtro digital es a través de la convolución de la entrada digital con una respuesta al impulso. De hecho, todos los filtros lineales pueden ser construidos de esta manera. Cuando la respuesta al impulso es utilizada para tal fin, el filtro resultante recibe el nombre de **filtro Kernel** [SMI1997].

Comprenderemos ahora como un sistema cambia una señal de entrada en una señal de salida. Primero, la señal de entrada es descompuesta en un set de impulsos, cada uno de los cuales puede ser visto como una función delta escalada y trasladada. Segundo, la salida resultante de cada impulso es una versión escalada y trasladada de la respuesta al impulso. Tercero, la señal de salida completa puede ser hallada sumando todas estas versiones escaladas y trasladadas de la respuesta al impulso. En otras palabras, *si conocemos la respuesta al impulso de un sistema, entonces podremos saber cual será la señal de salida para cualquier señal de entrada*. Esto significa conocer cualquier cosa acerca del sistema. Para ello, se necesitan comprender las características lineales del sistema.

La respuesta al impulso recibirá diferentes nombres según sus aplicaciones. Si el sistema es considerado como un *filtro*, la respuesta al impulso es llamada un **filtro Kernel**, la **convolución Kernel** o simplemente, el **Kernel**. En procesamiento de imágenes, la respuesta al impulso es llamada *función punto de propagación* [SMI1997]. Aunque estos términos se emplean para diferentes áreas, ambos significan lo mismo: la señal producida por un sistema cuando la entrada es una función delta, como se muestra en la figura 29.

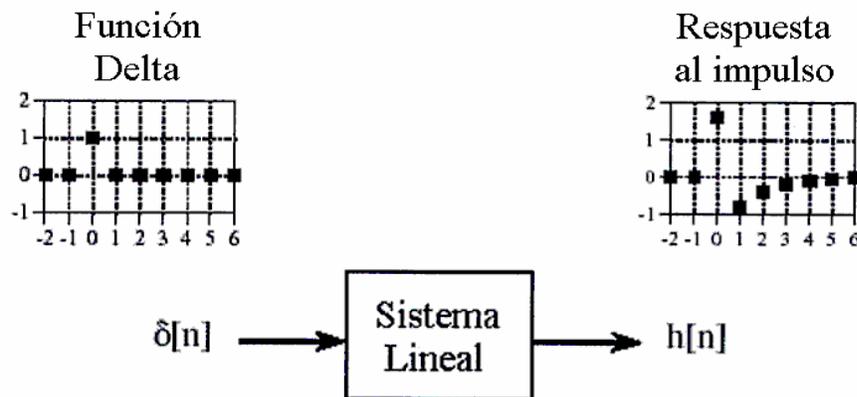


Figura 29: Esquema de la obtención de la respuesta al impulso de un filtro.

En sistemas lineales, la *convolución* es utilizada para describir la relación entre tres señales de interés: la señal de entrada, la respuesta al impulso y la señal de salida. Una señal $x(n)$, entra al sistema lineal con una respuesta al impulso $h(n)$, resultando una señal de salida $y(n)$. En forma de ecuación: $x(n)*h(n)=y(n)$ o por $y[n]=\sum_{k=-\infty}^{\infty} h(k)x(n-k)$.

Expresado en palabras, la señal de entrada convolucionada con la respuesta al impulso es igual a la señal de salida.

4.4.1.1 Una visión práctica de la convolución

La convolución de señales, está basada en el concepto fundamental del DSP, descomponer la entrada, pasar las componentes a través del sistema, y analizar la salida. En el siguiente ejemplo, cada uno de los términos en la señal de entrada contribuirá una versión escalada y trasladada de la respuesta al impulso a la señal de salida.

En el siguiente caso, la señal de entrada consta de 9 puntos y una respuesta al impulso de 4 puntos. Estos términos se muestran en la figura 30. Sumando cada una de estas nueve señales, se produce la señal de salida, $y(n)$.

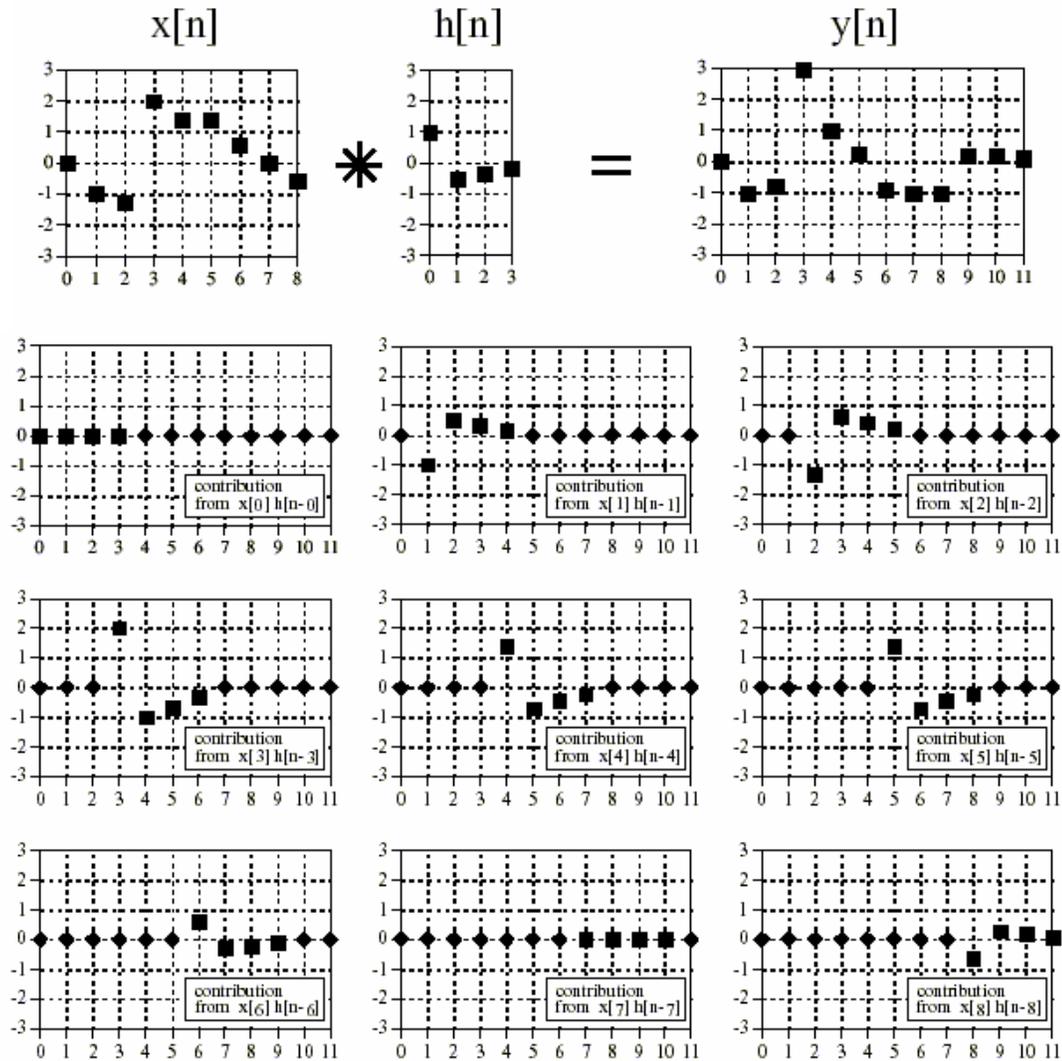


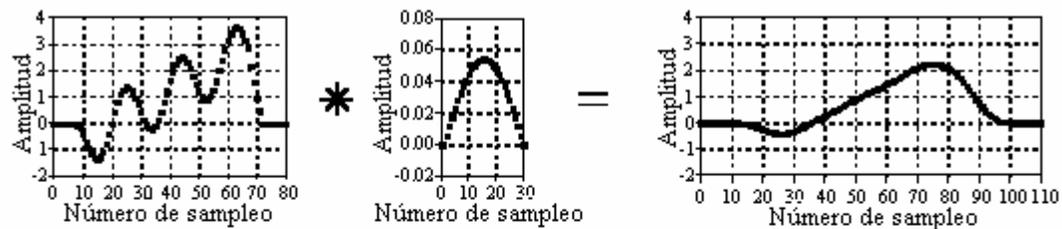
Figura 30: Ilustración de la convolución aplicada a un conjunto de datos. Cada punto de la entrada es convolucionado por la respuesta del sistema, y la salida corresponde a la suma de todas las contribuciones.

Observemos como cada punto de la señal de entrada contribuye a muchos puntos en la señal de salida. Este punto de vista es importante porque provee un entendimiento conceptual de como la convolución pertenece al DSP.

4.4.1.2 Filtros implementados por convolución

La figura 31 muestra como se usa la convolución para filtrar altas y bajas frecuencias. La señal de entrada del ejemplo es la suma de dos componentes: tres ciclos de una onda senoidal (representando una onda de alta frecuencia) y una rampa de ascenso lento (compuesta de frecuencias bajas). En (a), la respuesta al impulso para un filtro pasa-bajo es un suave arco, resultando solamente la forma de onda de rampa. Similarmente, el filtro pasa-alto en (b), deja pasar solamente la onda senoidal.

a. Filtro pasa-bajos



b. Filtro pasa-altos

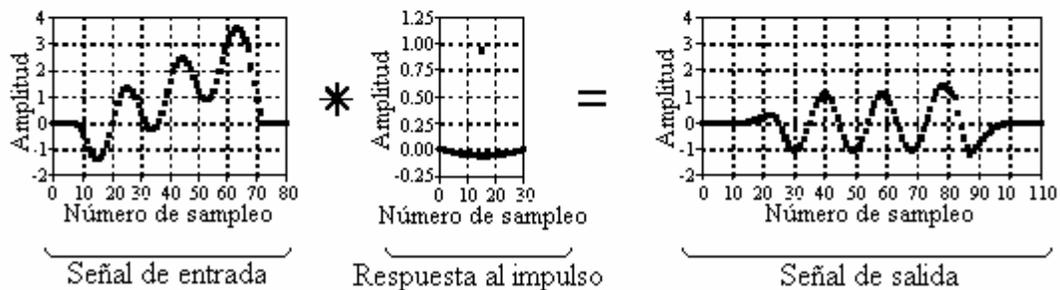


Figura 31. Filtros pasa-bajo y pasa-alto, ambos implementados por convolución.

En la figura 31, las entradas tienen 81 términos en el dominio del tiempo, mientras que cada respuesta al impulso está compuesta de 31 términos. En diversas aplicaciones del DSP, los términos de la señal de entrada son cientos, miles o millones. La respuesta al impulso es usualmente mucho más corta, de unos cuantos puntos a unos cuantos cientos de puntos. Las matemáticas asociadas a la convolución no son restrictivas en cuanto al largo que deben tener las señales. De todos modos, esto si especifica el largo de la señal de salida. La longitud de la señal de salida es igual a la longitud de la señal de entrada, más la longitud de la respuesta al impulso menos uno.

Para las señales de la figura 31, cada salida tiene $81+31-1$ términos de largo. La señal de entrada corre desde el término 0 hasta el término 80, la respuesta al impulso desde el término 0 al 30, y la señal de salida desde el término 0 al 110.

Esto nos advierte que la convolución es una poderosa operación en el sentido que podemos implementar un filtro con muy pocos puntos y la señal de salida será muy diferente a la señal de entrada.

4.4.1.3 Ejemplos implementados en MATLAB

Veamos un sencillo caso de como podemos modificar una señal de entrada si esta es convolucionada con una respuesta al impulso de muy pocos puntos.

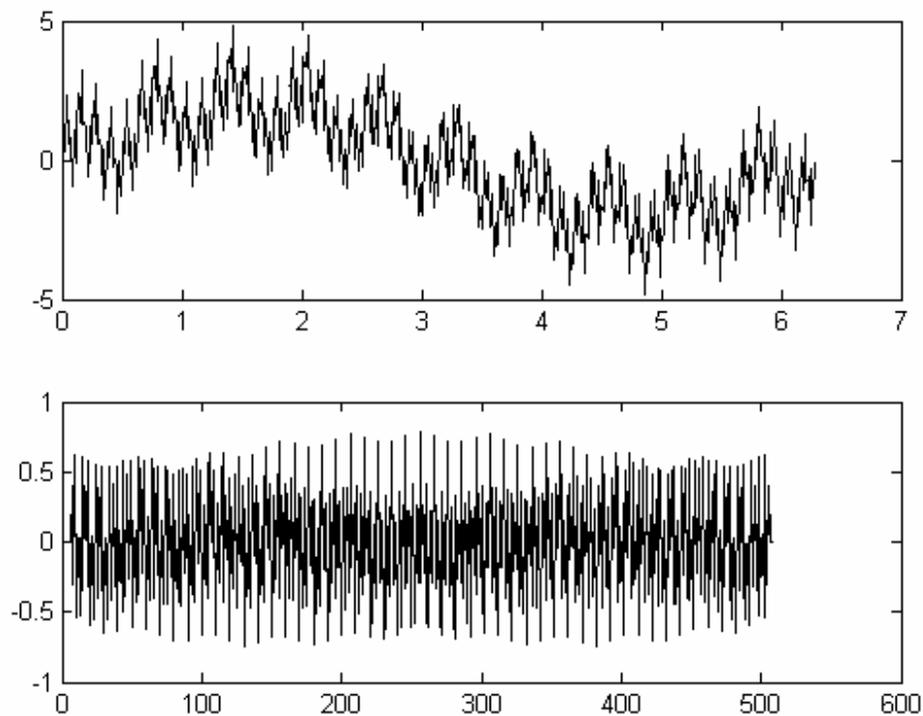


Figura 32: Implementación en MATLAB de un filtro por convolución. Aquí se eliminó la onda moduladora casi por completo.

En la figura 32 se muestran los resultados al usar la secuencia:

$$\mathbf{h}=[0;0;0;0;0;0.3;-0.3;0;0;0;0]$$

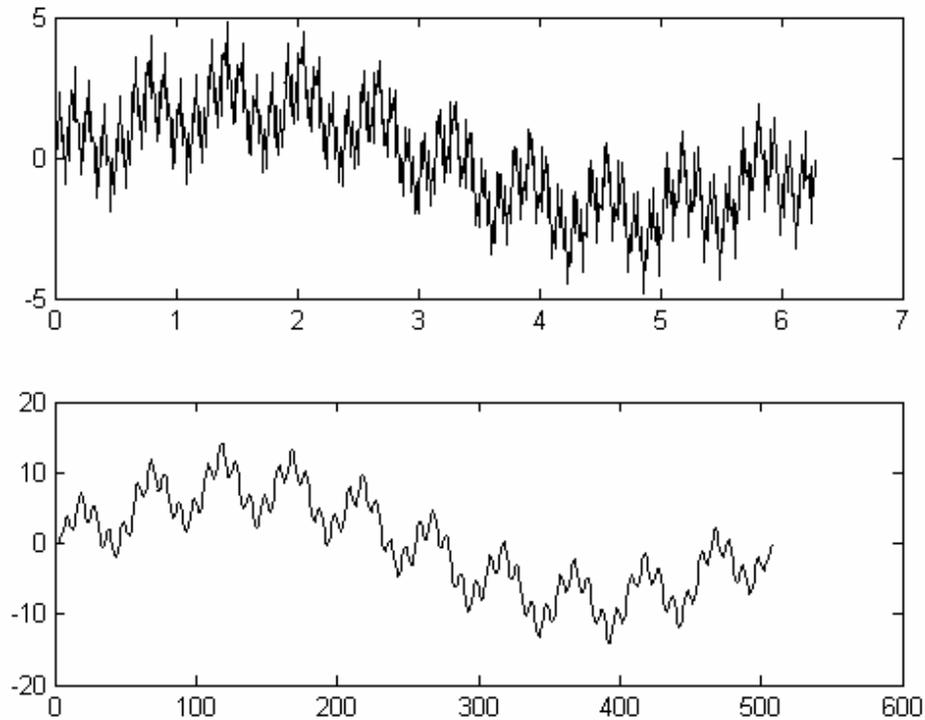


Figura 33: Implementación en MATLAB de un filtro por convolución. Notar la reducción del ruido de alta frecuencia.

En la figura 33 se muestra el resultado cuando la secuencia es:

$$\mathbf{h}=[0;0.1;0.3;0.5;0.7;0.9;0.7;0.5;0.3;0.1];$$

Como se puede observar, con distintas secuencias de la respuesta al impulso podemos obtener diferentes salidas después de efectuar la convolución. A continuación veremos métodos más específicos de filtrado de señales, en los cuales la convolución está involucrada de manera importante.

4.4.2 FILTROS MOVING AVERAGE (Media en movimiento)

El filtro Moving Average (media en movimiento) o filtro MA, es uno de los más usados en DSP, principalmente porque este es fácil de comprender y usar. Se considera muy bueno para reducir el ruido randómico mientras se retiene la forma de la respuesta al paso. Esto lo hace el mejor filtro para tratar señales en el dominio del tiempo. Sin embargo, el MA es el peor filtro para tratar señales en el dominio de la frecuencia, debido su poca eficiencia para separar una banda de frecuencias de otra. Existen variantes del filtro MA, tales como el Gaussiano, Blackman y múltiple-pass MA [SMI1997]. Estos tienen mejor rendimiento en el dominio de la frecuencia, a expensas de un incremento del tiempo de cálculo.

4.4.2.1 Implementación por Convolución

Como el nombre indica, un filtro MA opera promediando un número de puntos de la señal de entrada para producir cada punto de la salida. La ecuación que describe lo anterior esta dada por:

$$y_i = \frac{1}{M} \sum_{j=0}^{M-1} x_{i+j} , \quad (4.14)$$

donde x_i es la señal de entrada, y_i es la señal de salida y M es el número de puntos en la media. Por ejemplo, en un filtro MA de 5 puntos, el punto 80 en la señal de salida esta dado por:

$$y_{80} = \frac{x_{80} + x_{81} + x_{82} + x_{83} + x_{84}}{5} .$$

Como una alternativa, para obtener la salida, la elección de los puntos de la entrada puede ser de la siguiente manera:

$$y_{80} = \frac{x_{78} + x_{79} + x_{80} + x_{81} + x_{82}}{5} .$$

De estos dos ejemplos, vemos que tenemos dos casos para obtener la salida: uno *simétrico* y otro *antisimétrico*.

Podemos reconocer que el filtro MA es en realidad una convolución. Para ello, usemos un filtro Kernel. Por ejemplo, un filtro de 5 puntos tiene el siguiente Kernel:

$$\dots 0, 0, 1/5, 1/5, 1/5, 1/5, 1/5, 0, 0 \dots$$

esto es, el filtro MA es una convolución de la señal de entrada con un pulso rectangular de área 1 y amplitud 1/5.

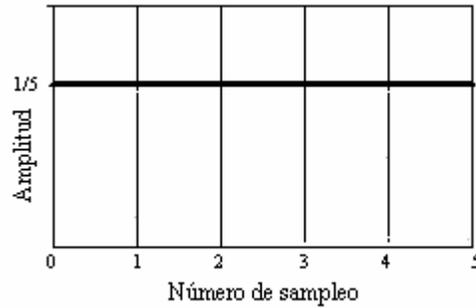


Figura 34: Respuesta al impulso de un filtro Kernel MA.

4.4.2.2 Reducción de Ruido v/s Respuesta al Paso

Debido a que el filtro MA es el más fácil de implementar, este es a menudo el primero que se considera al momento de encarar un problema. Además de ser muy bueno para muchas aplicaciones, es óptimo para un problema común: Reducir el ruido blanco mientras se mantiene la forma de la respuesta al paso, como se ilustra en la figura 35.

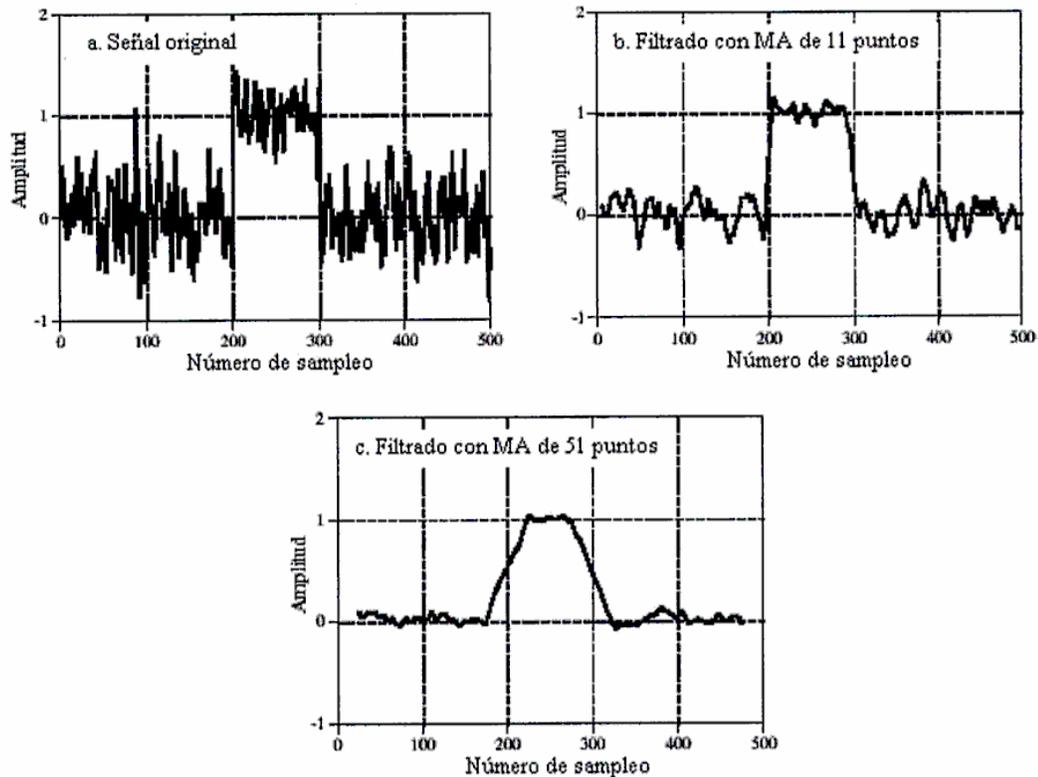


Figura 35: Reducción de ruido randómico en una señal, utilizando un filtro MA.

La figura 35 muestra un ejemplo de como trabajan estos filtros. La señal en (a) es un pulso que contiene ruido randómico. En (b) y (c) la acción del filtro MA, decremента la amplitud del ruido (esto es bueno), pero también cambia la forma de los bordes (lo cual es malo). De todos los posibles filtros lineales que podemos usar, el MA produce una reducción de ruido a expensas de un cambio de la forma de los bordes. La cantidad de ruido reducido es igual a la raíz cuadrada de la cantidad de puntos en la media. Por ejemplo, un MA de 100 puntos reduce el ruido en un factor de 10 [SMI1997].

$$NR \propto \sqrt{M} \text{ (dB)}. \quad (4.15)$$

Para comprender por que el filtro MA es la mejor solución en el dominio del tiempo, imaginemos que debemos diseñar un filtro con una forma fija de los bordes. Agreguemos a esto la especificación de que son 11 puntos en la subida de la respuesta en paso. Esto requiere que el filtro Kernel tenga 11 puntos. Para lograr la solución óptima, debemos preguntarnos como escoger los 11 valores en el filtro Kernel para minimizar el ruido en la señal de salida. Puesto que el ruido que estamos tratando de reducir es randómico, ninguno de los puntos de entrada es especial, cada uno esta exactamente tan ruidoso como su vecino. Por consiguiente, no debemos entregar un tratamiento preferencial a ninguno de los puntos de entrada. Esto se logra asignando un gran coeficiente en el filtro Kernel que actúe sobre todos los puntos. La disminución de ruido es obtenida cuando todos los términos de la entrada son tratados de igual manera. En este principio esta basado el filtro MA.

4.4.2.3 Respuesta en Frecuencia

La respuesta en frecuencia de un filtro MA está matemáticamente descrita por la Transformada de Fourier de un pulso rectangular:

$$H(f) = \frac{\sin(\pi f M)}{M \sin(\pi f)}. \quad (4.16)$$

Notamos que la función de transferencia del filtro corresponde solamente a la *Función de Dirichlet* [APO1972].

En la figura 36 observamos algunas respuestas en frecuencia de filtros MA. Podemos ver que el roll-off es lento y la atenuación de la banda de parada es pésima. Claramente, el filtro MA no puede separar una banda de frecuencias de otra. Recordemos que, a mejor rendimiento en el dominio del tiempo, peor rendimiento en el dominio de la frecuencia y viceversa. En resumen, el MA es un muy buen filtro suavizador (dominio del tiempo), y a la vez, un pésimo filtro pasa-bajo (dominio de la frecuencia)

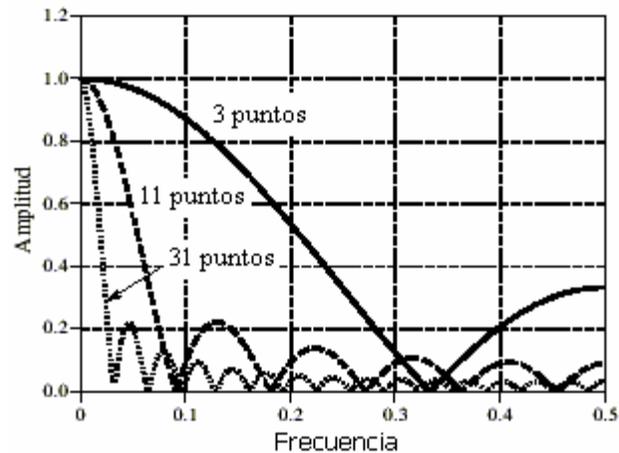


Figura 36: Respuesta en frecuencia de un filtro MA.

El método que veremos a continuación es exactamente lo contrario al filtro M.A., en el sentido que será un muy buen filtro en el dominio de la frecuencia, y no tan bueno en el dominio del tiempo.

4.4.3 FILTROS DE ENVENTANADO-SINC

Los filtros de enventanado-sinc son utilizados para separar una banda de frecuencias de otra. Ellos son muy estables, producen pocas molestias y pueden ser llevados a increíbles niveles de rendimiento. Estas excepcionales características en el dominio de la frecuencia son obtenidas a las expensas de un pobre rendimiento en el dominio del tiempo, incluyendo un excesivo Ripple y Overshoot en la respuesta al paso. Los filtros WS (*Windowed sinc*) son fáciles de programar, pero lentos en su ejecución. Sin embargo, el algoritmo FFT puede incrementar la velocidad de ejecución de manera notable.

4.4.3.1 Estrategia del enventanado-sinc

En la parte (a) de la figura 37, se muestra la respuesta en frecuencia de un filtro pasabajo ideal. Todas las frecuencias por debajo de la frecuencia de corte, son pasadas con amplitud unitaria, mientras que todas las frecuencias superiores a esta, son bloqueadas. La banda de paso es perfectamente plana, la atenuación en la banda de parada es infinita, y la transición entre las dos es infinitamente pequeña. Tomando la transformada inversa de Fourier de esta respuesta ideal en frecuencia, aparece el filtro Kernel ideal (respuesta al impulso), mostrada en la parte (b). Esta curva está dada matemáticamente por la función Sinc:

$$\text{Sinc}(x) = \frac{\sin(x)}{x},$$

y la función de transferencia por:

$$H(n) = \frac{\sin(2\pi fcn)}{n\pi}. \quad (4.17)$$

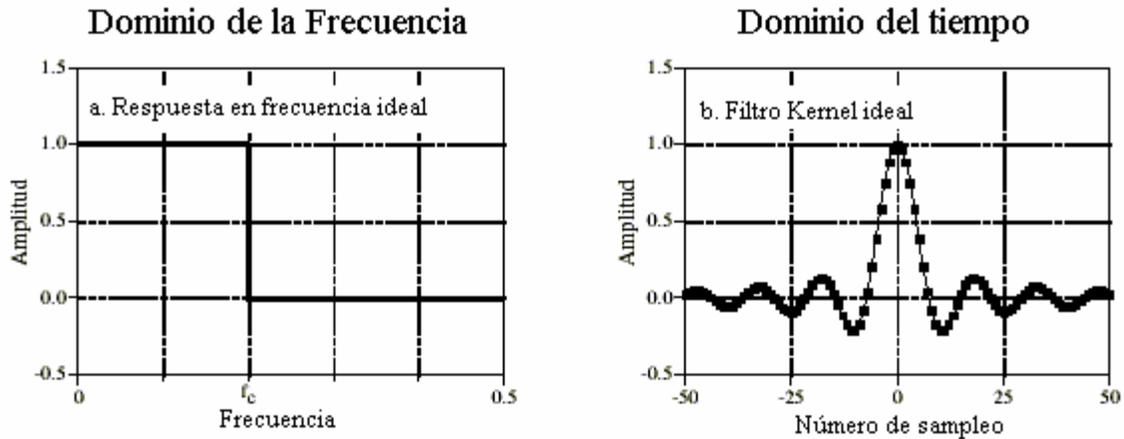


Figura 37: Representación en la frecuencia y el tiempo de un filtro pasa-bajo ideal.

Al convolucionar una señal de entrada con este filtro Kernel se realiza un *filtrado pasa-bajo perfecto*. Notar sin embargo que la función *sinc* contiene un lado negativo y otro positivo con decaimiento asintótico (infinito) a cero. Aquí surge un problema, puesto que ningún computador puede procesar una cantidad infinita de datos..

Es posible hacer algunas modificaciones a la función *sinc*, cuyo resultado se ilustra en la figura 38-c. Primero, esta es truncada en puntos, los cuales son sistemáticamente escogidos en torno del lóbulo principal, donde M es un número par. Todos los términos fuera de estos puntos son llevados a cero, o simplemente son ignorados. Segundo, la secuencia entera es trasladada hacia la derecha, corriendo desde 0 hasta M . Esto permite representar al filtro Kernel usando solo índices positivos. El único efecto de estos cambios en el filtro Kernel es que la señal de salida es trasladada en $M/2$.

Puesto que el filtro Kernel modificado es sólo una aproximación del filtro Kernel ideal, este no tendrá una respuesta en frecuencia ideal. Para hallar la respuesta en frecuencia, se puede tomar la transformada de Fourier de la señal en (c), resultando la curva que se ve en (d). Notamos entonces un excesivo Ripple en la banda de paso y una pobre atenuación en la banda de parada (conocido como efecto Gibbs) [HSU1973]. Estos problemas resultan de la abrupta discontinuidad de la función *sinc* al ser esta truncada. Incrementando el largo del filtro Kernel no se soluciona este problema: la discontinuidad aparecerá no importando que largo M asignemos al filtro.

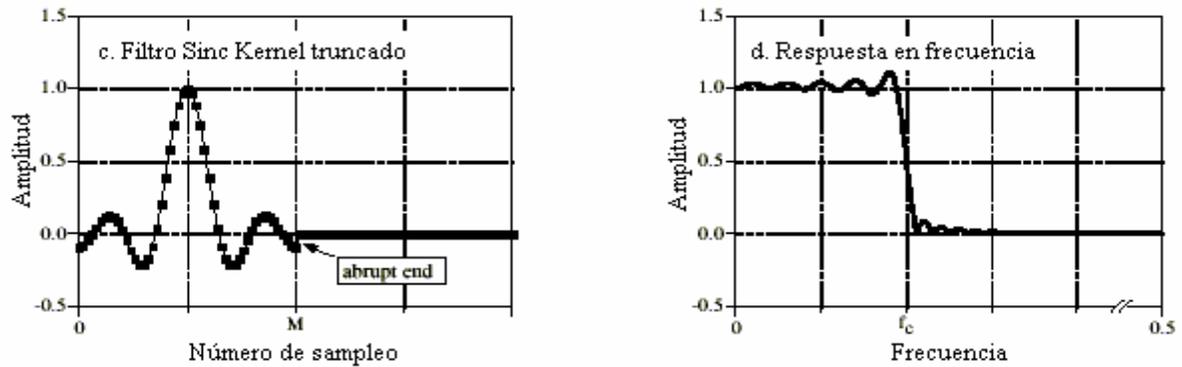


Figura 38: Respuesta al impulso truncada de un filtro pasa-bajo ideal y su respectiva respuesta en frecuencia.

Sin embargo, existe un método sencillo para mejorar esta situación: la figura 39 (e) muestra una suave curva de una función llamada **ventana**. Multiplicando la función truncada sinc de la figura 38(c) con la ventana, resulta el **filtro Kernel WS** mostrado en la figura 39 (f). La idea de esto es reducir el abrupto final de la función truncada y de este modo mejorar la respuesta en frecuencia. La figura 39 (g) muestra este mejoramiento. La banda de paso es ahora plana, y la atenuación de la banda de parada es buena pero no puede ser apreciada en este gráfico.

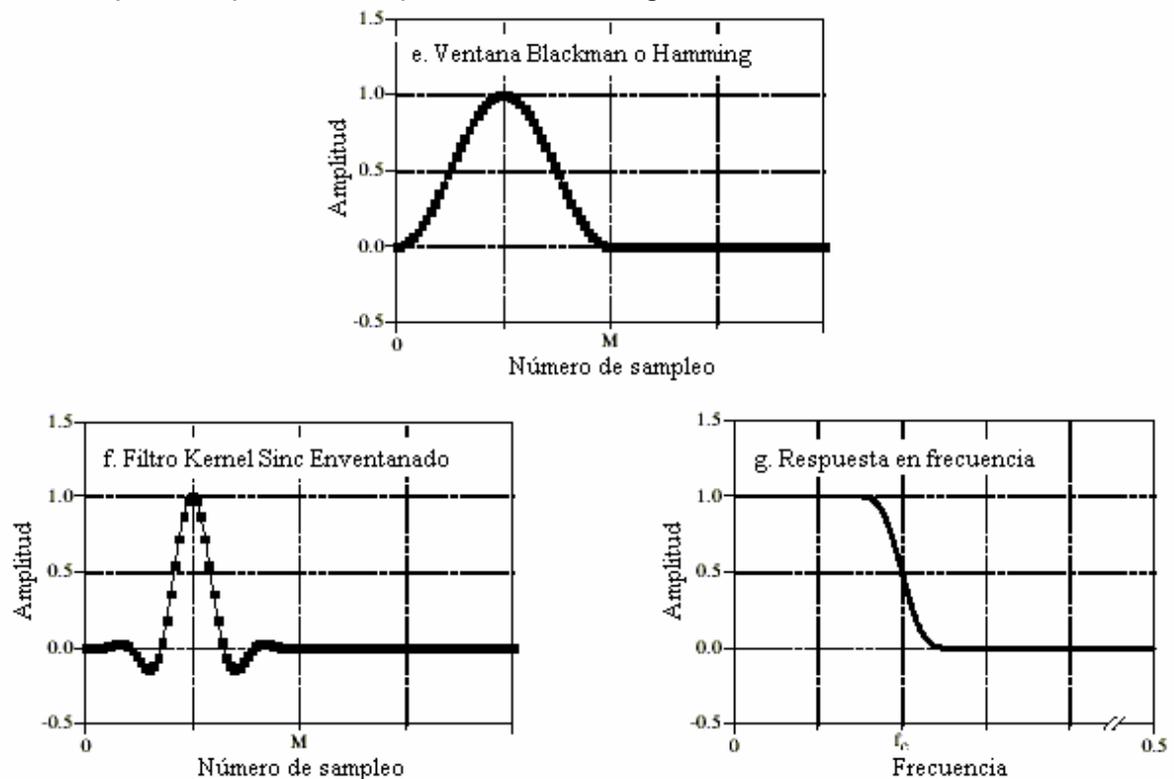


Figura 39: En la parte (e) se muestra la gráfica de la función ventana "Blackman". En (f) se muestra la función sinc truncada, pero ahora más suavizada en la discontinuidad. (g) muestra su respuesta en frecuencia.

Existen diversas y diferentes ventanas. De ellas sólo dos son las más utilizadas: la **Hamming** y la **Blackman** [SMI1997]. Estas están dadas por las siguientes ecuaciones:

Ecuaciones hamming y Blackman.

$$w(n) = 0.54 - 0.46 \cos(2\pi n/M), \quad (4.18)$$

$$w(n) = 0.42 - 0.5 \cos(2\pi n/M) + 0.08 \cos(4\pi n/M) \quad (4.19)$$

La figura 40 muestra la forma de estas dos ventanas.

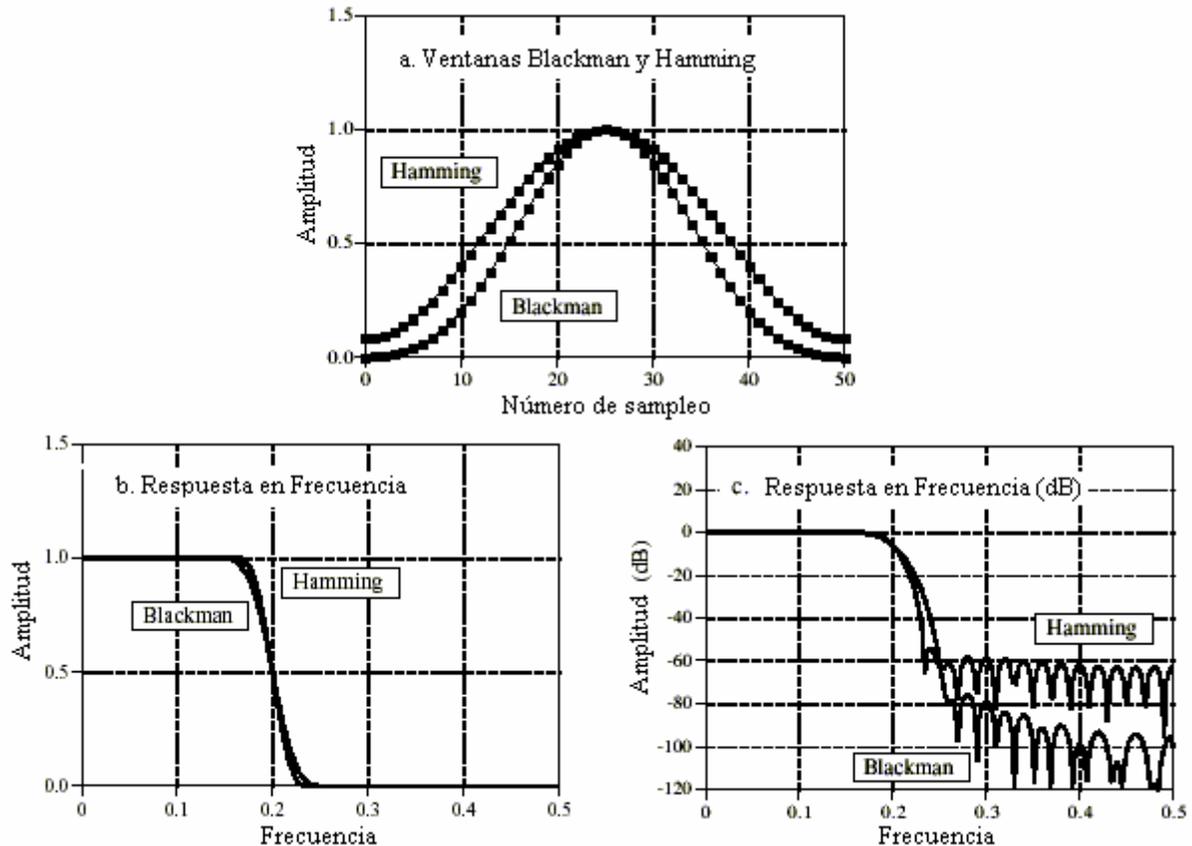


Figura 40: Ventanas Hamming y Blackman y sus respuestas en frecuencia en escala lineal y logarítmica.

4.4.3.2 Consideraciones sobre ventanas

Notamos que la ventana Hamming decae aproximadamente un 20% más rápido que la Blackman. Como se muestra en la figura 40(b), la ventana Hamming tiene más o menos un 20% de roll-off que la Blackman. Sin embargo, en la figura 40(c) se ve que la Blackman tiene una mejor *atenuación en la banda de parada*. Para ser exactos, la atenuación de la banda de parada para la Blackman es -74dB (-0.02%), mientras que la Hamming es solo -53dB (-0.2%). Aunque esto no puede ser visto en estos gráficos, la Blackman tiene un ripple de pasa-banda de solo aproximadamente 0.02%, mientras que la Hamming es típicamente 0.2%. En general, la Blackman puede ser la primera elección, pues un lento roll-off es más fácil de manipular que una pobre atenuación de la banda de parada.

Estas son otras ventanas que también podrían ser utilizadas, aunque ellas decaen más rápido que la Hamming o Blackman. La ventana Bartlett es un triángulo. La ventana Hanning esta dada por la siguiente ecuación:

Ecuación de la Ventana Hanning:

$$w(n) = 0.5 - 0.5 \cos(2\pi n/M). \quad (4.20)$$

Estas dos ventanas tienen aproximadamente la misma velocidad de roll-off que la Hamming, pero peor atenuación en la banda de parada (*bartlett: -25dB o 5.6%, Hanning: -44dB o 0.63%*) [SMI1997].

También está la ventana rectangular. Esta, más que una ventana, corresponde a un truncamiento de los extremos. Mientras que el roll-off es -2.5 veces más rápido que la Blackman, la atenuación de la banda de parada es solo de -21dB (8.9%).

4.4.3.3 Diseño de Filtros WS

Para diseñar un filtro WS, se deben seleccionar dos parámetros: la frecuencia de corte f_c y el largo del filtro Kernel M . La frecuencia de corte es expresada como una fracción de la razón de muestreo, y por lo tanto debe estar entre 0 y 0.5. El valor de M establece el roll-off más adecuado a la aproximación. Este está dado por:

$$M = \frac{4}{BW}, \quad (4.21)$$

donde BW es el ancho de la banda de transición, medido a partir del punto donde la curva deja exactamente la unidad, hasta cuando alcanza el valor cero (es decir, 99% a 1% de la curva) La banda de transición es también expresada como una fracción de la frecuencia de muestreo, y debe estar entre 0 y 0.5. La figura 41 muestra un ejemplo de como se usa esta aproximación. Las tres curvas mostradas son generadas con el filtro Kernel, con $M= 20, 40$ y 200 ; y los anchos de banda son 0.2, 0.1, y 0.02 respectivamente. Estos últimos muestran que la respuesta en frecuencia no depende de la frecuencia de corte seleccionada.

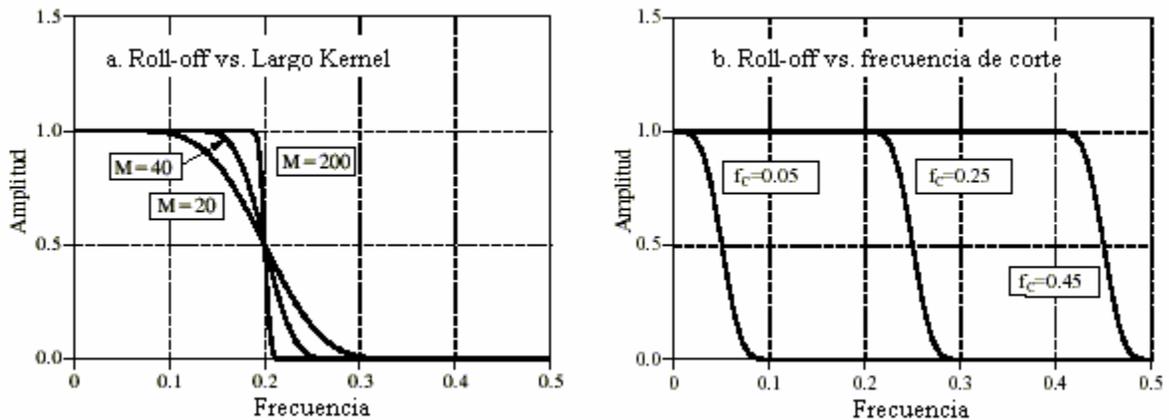


Figura 41: La parte (a) muestra la relación entre la pendiente en la banda de transición y el largo del filtro WS. La parte (b) muestra que no hay dependencia entre la pendiente de la banda de transición y la frecuencia de corte.

Puesto que el tiempo requerido para la convolución es proporcional al largo de las señales, la ecuación anterior expresa el trade-off entre el *tiempo de computo* (que depende de M) y la forma del filtro (que depende del valor de BW)

Por ejemplo, el 20% más lento de roll-off de la ventana Blackman (en comparación con la Hamming) puede ser compensado usando un filtro Kernel 20% mas largo. En otras palabras, se puede ver que la ventana Blackman es un 20% más lenta a ejecutar que el equivalente roll-off de la ventana Hamming. Esto es importante porque *la velocidad de ejecución de los filtros WS es extremadamente lenta*.

Como vemos en la figura, la frecuencia de corte de los WS es medida a partir del punto medio de amplitud. Observemos que de hecho se usan 0.5 en comparación de los 0.707 (-3dB) usados en filtros analógicos y otros filtros digitales. Esto se debe a que la respuesta en frecuencia de los filtros WS es simétrica entre la banda de paso y la banda de parada. Por ejemplo, la ventana Hamming tiene un ripple de banda de paso de 0.2%, y la misma atenuación en la banda de parada. (Esto es, ripple en la banda de parada) de 0.2 %. Otros filtros no muestran esta simetría, y por lo tanto no tienen ventaja en usar el punto de amplitud media que marca la frecuencia de corte. Esta simetría hace al filtro WS ideal para la *inversión espectral* [SMI1997].

Después que han sido seleccionados f_c y M , el filtro Kernel es determinado de la siguiente relación:

$$h(n) = k \frac{\sin(2\pi f_c(n - M/2))}{n - M/2} \left[0.42 - 0.5 \cos\left(\frac{2\pi n}{M}\right) + 0.08 \cos\left(\frac{4\pi n}{M}\right) \right]. \quad (4.22)$$

Notemos que en la ecuación (4.22) aparece la función sinc, la forma de $M/2$ y la ecuación de la ventana Blackman. Para que el filtro tenga ganancia unitaria en DC, la constante k debe ser escogida de tal manera que la suma de todos los términos sea igual a uno.

La figura 42 muestra ejemplos de filtros Kernel WS, y sus correspondientes respuestas al paso. Los términos para el comienzo y final de los filtros Kernel son tan pequeños que no pueden ser vistos en los gráficos. Estos términos pueden ser pequeños en valor, no obstante, ellos colectivamente tendrán un gran efecto en el rendimiento del filtro. El rendimiento de este filtro en el dominio del tiempo es pobre. La respuesta en paso tiene overshoot y ringing. Claramente su rendimiento es mejor en el dominio de la frecuencia.

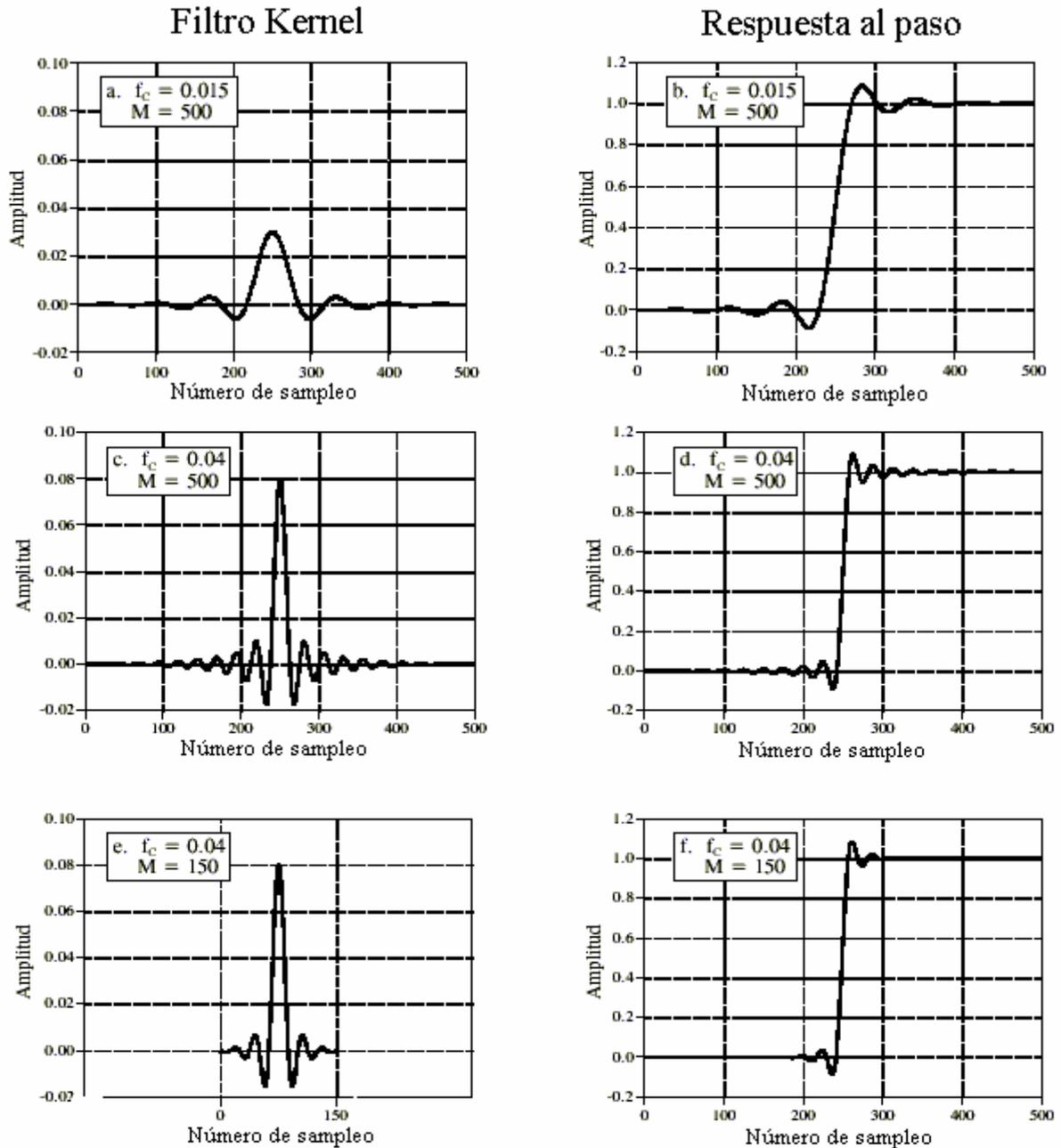


Figura 42: Ejemplos de filtros WS y sus respectivas respuestas en paso. Como se observa, la respuesta de estos filtros en el dominio del tiempo es deficiente.

4.5 FILTROS RECURSIVOS

Los filtros recursivos son un camino eficiente para lograr una muy buena respuesta al impulso, sin necesidad del enorme rendimiento que implica la convolución. Ellos se ejecutan *muy* rápidamente, pero tienen menor flexibilidad que otros filtros digitales. A continuación veremos cual es el fundamento con el que se desarrolla este tipo de filtros.

4.5.1 El método recursivo

Para comprender el método recursivo, imaginemos una señal de entrada $x(n)$, la cual al pasar por nuestro filtro nos dará la salida $y(n)$. Debemos preguntarnos que información utilizaremos para calcular $y(n)$, siendo n un número fijo.

La fuente de información más obvia es *la señal de entrada*, esto es, los valores $x(n)$, $x(n-1)$, $x(n-2)$, etc. Se podría multiplicar cada valor de la entrada por un coeficiente y sumar los productos:

$$y[n] = a_0x[n] + a_1x[n-1] + a_2x[n-2] + \dots \quad (4.23)$$

Esto no es más que una convolución, con los coeficientes a_0 , a_1 , a_2 , etc., formando la convolución Kernel. Si esto es todo lo que se puede hacer, entonces no estaríamos sino diseñando un filtro FIR. Sin embargo, existe otra fuente de información a la cual se tiene acceso: los valores *previamente calculados* de la señal de salida; estos son: $y(n-1)$, $y(n-2)$, etc. Usando esta información adicional, el algoritmo para obtener la salida calculado será el siguiente:

$$y[n] = a_0x[n] + a_1x[n-1] + a_2x[n-2] + \dots + b_1y[n-1] + b_2y[n-2] + \dots \quad (4.24)$$

Notemos que esta no es más que la ecuación de diferencias para un sistema lineal LTI. En palabras, cada punto de la señal de salida es hallado por la multiplicación de los valores de la señal de entrada con un coeficiente "a", y la multiplicación de los valores de la salida previamente calculados, con un coeficiente "b", y sumando todos estos productos. Notar que no está el término b_0 pues este corresponde al término que se está calculando. La ecuación (4.24) es llamada **ecuación de recursión** y los filtros que utilizan esta son llamados **filtros recursivos**. Los valores "a" y "b" que definen el filtro son llamados **coeficientes de recursión**. En la práctica actual, no pueden usarse más de una docena de coeficientes de recursión, aproximadamente, pues de lo contrario el filtro se vuelve inestable.

Los filtros recursivos son útiles porque ellos evitan una gran convolución. Por el momento, consideremos qué pasa cuando una función delta pasa a través de un filtro recursivo. La salida corresponde a la respuesta al impulso del filtro, y típicamente será una oscilación sinusoidal cuya amplitud decaerá exponencialmente en el tiempo.

Puesto que la respuesta al impulso es infinitamente larga, los filtros recursivos son llamados de *respuesta al impulso infinita* o **filtros IIR**. En efecto, los filtros recursivos convolucionan la señal de entrada con un enorme filtro Kernel, de lo que se considerarán sólo unos pocos términos.

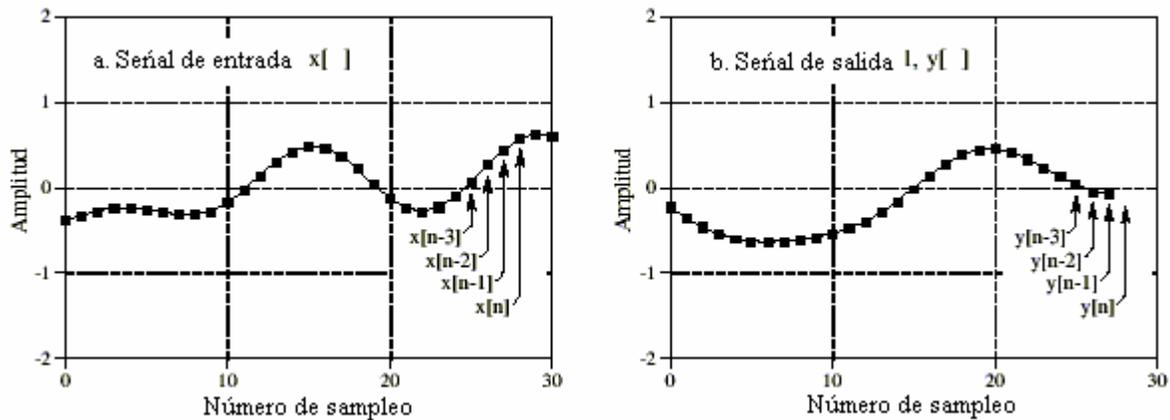


Figura 43: Ilustración de como la información es utilizada para encontrar la salida de un sistema recursivo.

La relación entre los coeficientes de recursión y la respuesta del filtro está dada por la transformada Z [SMI1997].

4.5.2 DISEÑO DE FILTROS DIGITALES RECURSIVOS

El diseño de los filtros digitales recursivos se obtiene comenzando con un modelo de filtro analógico, tales como Butterworth, Cheychev o elíptico [VER1997]. Una serie de conversiones matemáticas son aplicadas en el modelo análogo para obtener el filtro deseado. La transformada Z es una herramienta adecuada en este caso.

4.5.2.1 Filtros Recursivos “Single Pole”

La figura 44 muestra lo que es llamado un *Filtro pasa-bajo Single-Pole*. Este filtro recursivo usa solamente dos coeficientes, $a_0=0.15$, y $b_1= 0.85$. Para este ejemplo la función de entrada es una función de paso, mejor conocida como escalón unitario. Como se puede esperar de un filtro pasa-bajo, la salida es un suave rizo con un nivel estable. La figura también muestra el mismo filtro, pero en versión analógica. Esta puede hacerse fácilmente mediante la combinación de un condensador y una resistencia.

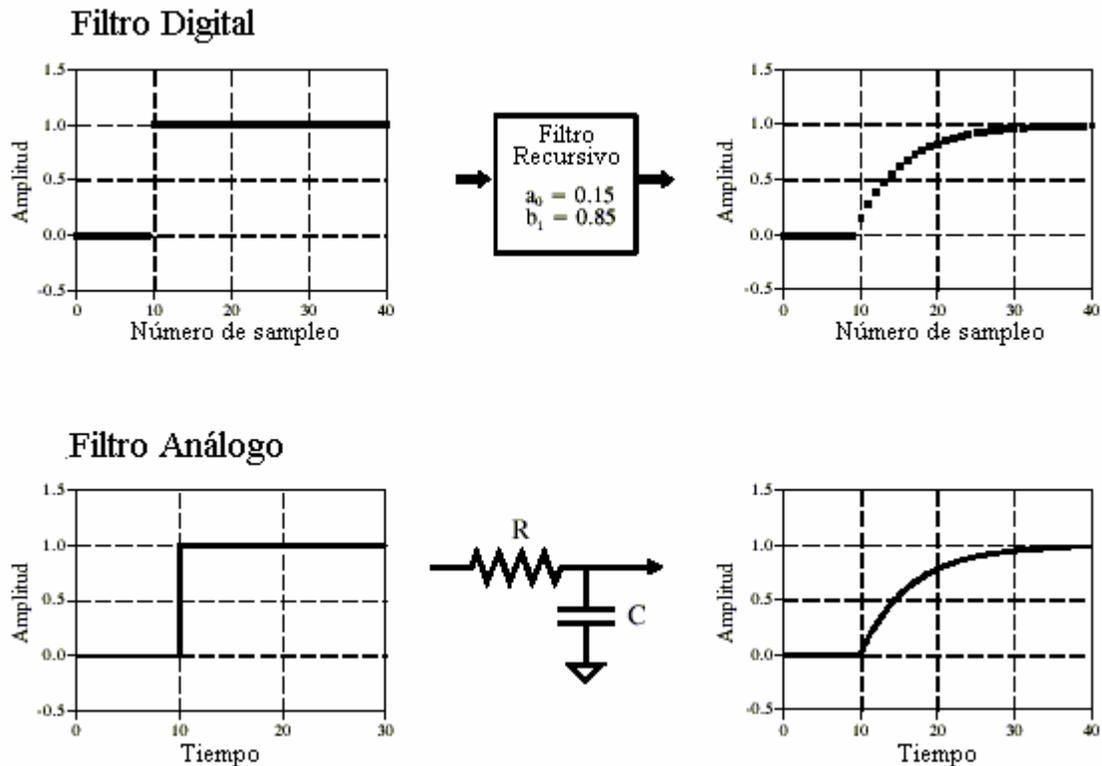


Figura 44: Ejemplo de un filtro recursivo pasa-alto y su equivalente filtro analógico.

La belleza del método recursivo está en su facilidad para crear una gran variedad de respuestas con sólo cambiar algunos parámetros. Por ejemplo, la figura 45 muestra un filtro con tres coeficientes: $a_0 = 0.93$, $a_1 = -0.93$ y $b_1 = 0.86$. Como se puede apreciar, el filtro digital imita la respuesta de un filtro analógico pasa-alto.

Los coeficientes de estos filtros Single Pole son hallados a través de sencillas ecuaciones:

Filtro Single Pole pasa-bajo

$$a_0 = 1 - x$$

$$b_1 = x$$

Filtro Single Pole pasa-alto

$$a_0 = (1 + x)/2$$

$$a_1 = -(1 + x)/2$$

$$b_1 = x$$

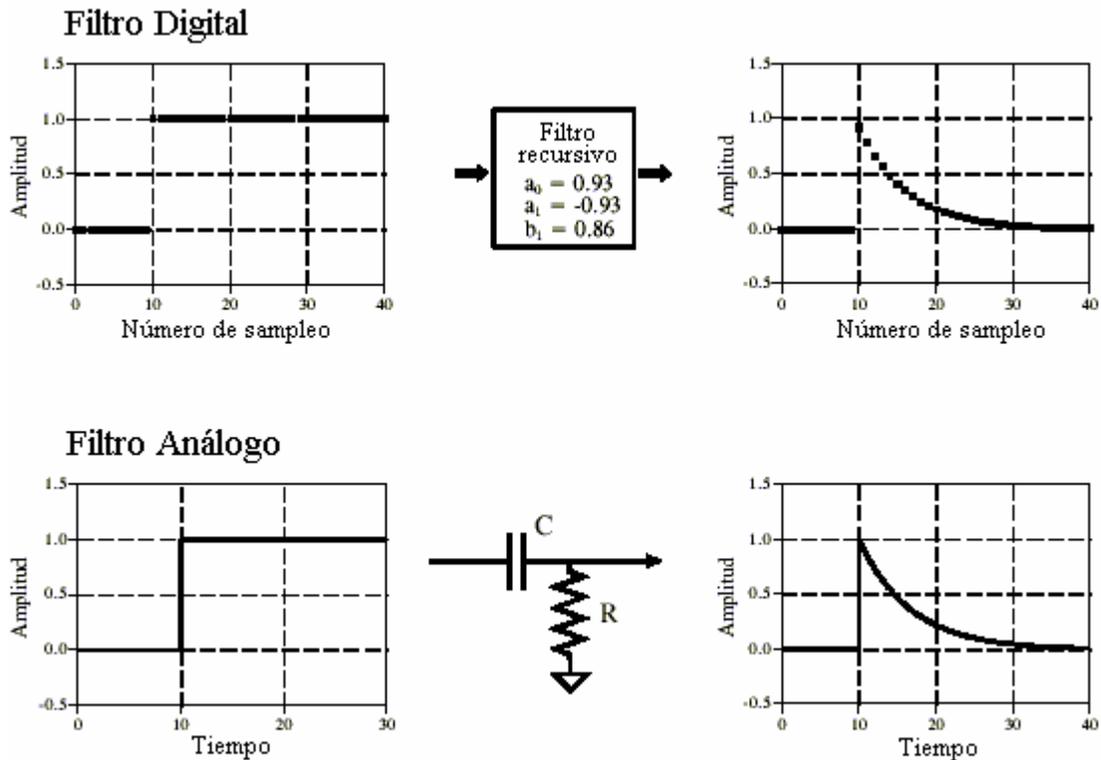


Figura 45: Respuesta en frecuencia de un filtro recursivo pasa-bajo y su equivalente filtro analógico.

Las características de estos filtros son controladas por el parámetro x , que está definido entre 0 y 1. Físicamente, x es la cantidad de decaimiento entre términos adyacentes. Por ejemplo, x es 0.86 en la figura 3, significando que cada punto en la señal de salida es 0.86 veces el valor anterior.

Notar que el filtro se puede volver inestable si x es más grande que 1. Esto es, cualquier valor distinto de cero en la entrada hará que la salida se incremente hasta que ocurran fluctuaciones (oscilaciones inestables).

El valor para x puede ser especificado directamente, o de la *constante de tiempo* del filtro [SAV2000]. RC es el tiempo que le toma a un circuito RC en decaer hasta el 36% de su valor inicial, y d es el número de tiempos que le toma a un filtro recursivo decaer a este valor.

$$x = e^{-1/d}$$

Por ejemplo, un decaimiento término a término de $x = 0.86$ corresponde a una constante de tiempo de $d = 6.63$ tiempos (como se muestra en la figura 46). También esta es una relación fija entre x y los -3dB de la frecuencia de corte, f_c para el filtro digital.

Esto nos proporciona tres caminos para hallar los coeficientes “ a y b ”, comenzando con la constante de tiempo, la frecuencia de corte, o solamente eligiendo x .

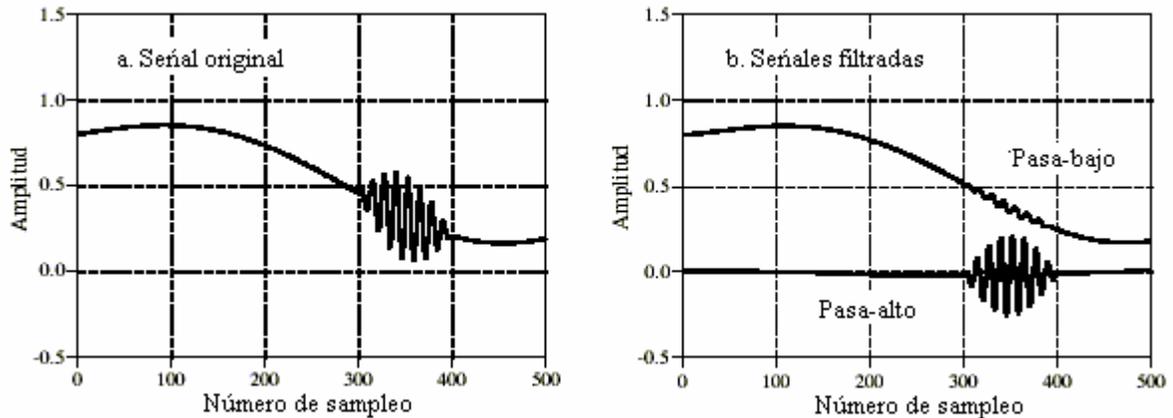


Figura 46: Ejemplo de como actúan dos filtros recursivos Single-Pole pasa-alto y pasa-bajo simultáneamente.

La figura 46 muestra el uso de un filtro Single Pole. En (a) la señal original es una curva suave excepto por una fluctuación de una onda sinusoidal de alta frecuencia. La figura (b) muestra a la señal después de haber pasado por un filtro pasa-bajo y un pasa-alto. Las señales han sido separadas satisfactoriamente pero no en forma perfecta. Solo como si se hubiera utilizado un circuito RC en una señal analógica.

Algunas respuestas en frecuencia de filtros Single Pole se muestran en la figura 47. Estas curvas son obtenidas al pasar una función delta a través del filtro, obteniendo con ello la respuesta a impulso del filtro. La FFT se utilizará aquí para convertir la respuesta al impulso en una respuesta en frecuencia. En principio, la respuesta al impulso es infinitamente larga. Sin embargo, esta decae por debajo del ruido de fondo luego de unas 15 a 20 constantes de tiempo. Por ejemplo, cuando la constante de tiempo del filtro es $d= 6.63$, la respuesta al impulso puede ser obtenida después de aproximadamente 128 samples.

La característica primordial de estos filtros es que tienen poca facilidad para separar una banda de frecuencias de otra. En otras palabras, ellos tienen un buen rendimiento en el dominio del tiempo, y por ende, un mal rendimiento en el dominio de la frecuencia. La respuesta en frecuencia puede ser mejorada utilizando etapas en cascada. Esto puede ser llevado a cabo de dos maneras: Primero, la señal puede ser pasada a través de un filtro varias veces. Segundo, la transformada Z puede ser utilizada para encontrar los coeficientes de recursión que combinan la cascada en una única etapa. La figura (c) muestra la respuesta en frecuencia de una cascada de cuatro etapas.

Aunque la atenuación de la banda de parada ha sido significativamente mejorada, el roll-off es pésimo. Si se desea mejorar el rendimiento en el dominio de la frecuencia, se puede recurrir a los llamados *Filtros Chebyshev* [SMI1997], [SAV2000].

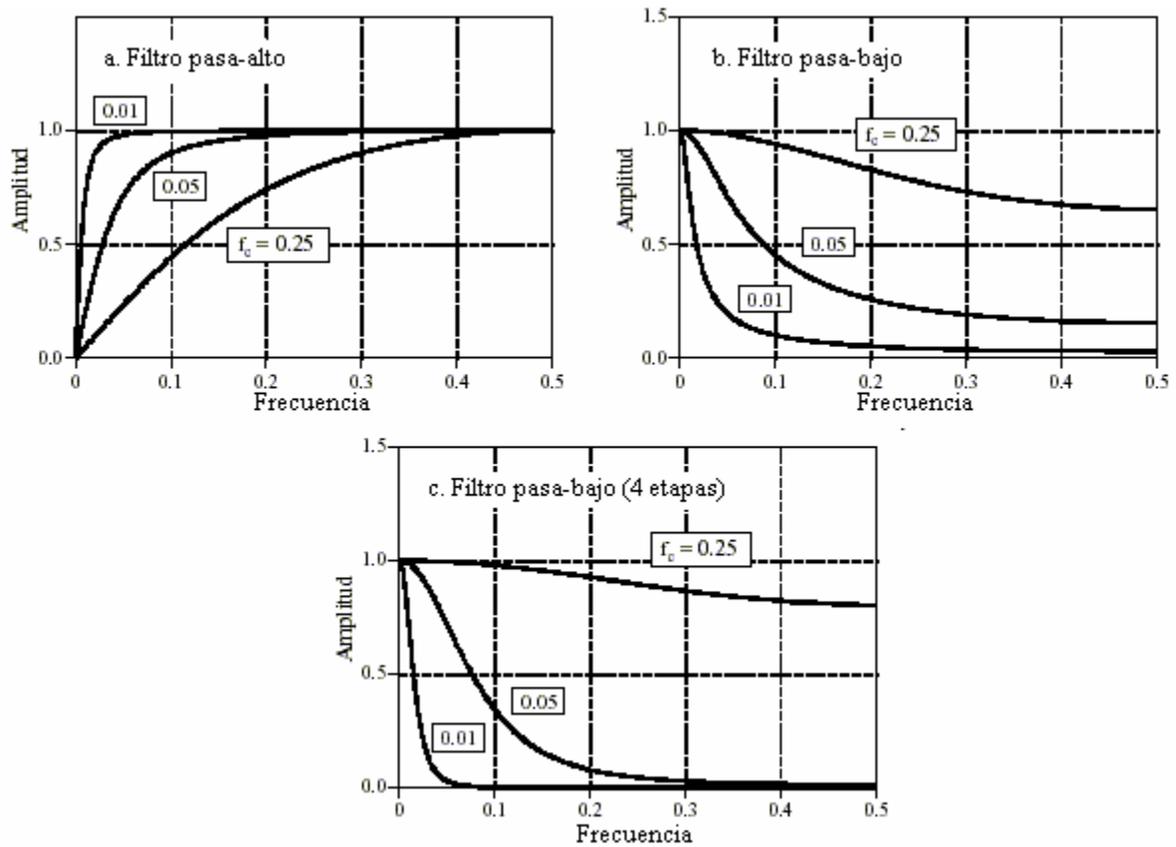


Figura 47: Ejemplos de respuestas en frecuencia de filtros recursivos Single-Pole.

4.5.2.2 Filtros “Notch”

Una de las necesidades que tienen en común la electrónica y el DSP es aislar una pequeña banda de frecuencias de una señal con ancho de banda extenso. Por ejemplo, se pueden aislar 60 Hz de una señal de interferencia de un equipo de instrumentación electrónico. Están disponibles dos tipos de respuesta en frecuencia: El *pasa-banda* y el *rechaza-banda* (también llamado **Filtro Notch**).

Se deben seleccionar dos parámetros para poder usar estas ecuaciones: f_c la frecuencia central, y BW , el ancho de banda (medido a partir de los 0.707 de amplitud) Ambos son expresados como una fracción de la frecuencia de muestreo, y por lo tanto debe estar entre 0 y 0.5

Como se muestra en la figura 48(a), el filtro pasa-banda tiene una superficie relativamente ancha, lo cual es indeseable para efectos de aislar una frecuencia en particular. Esto puede ser mejorado por etapas de cascada. Puesto que el diseño de estas ecuaciones es relativamente grande, se puede pasar la señal varias veces por el filtro, lo cual es preferible a tratar de hallar los coeficientes de un filtro con esas características.

La figura 48(b) muestra ejemplos de un filtro rechaza-banda. El estrechamiento del ancho de banda puede ser obtenido con precisión única y ser aproximadamente de 0.0003 de la frecuencia de muestreo [SMI1997]. Cuando se va más allá de este límite, la atenuación del Notch se degradará. La figura 48(c) muestra la respuesta al paso del filtro rechaza-banda. Notamos que tiene overshoot y ringing, pero su amplitud es bastante menor. Esto le permite al filtro remover interferencia de banda estrecha con una menor distorsión de la forma de onda en el dominio de la frecuencia.

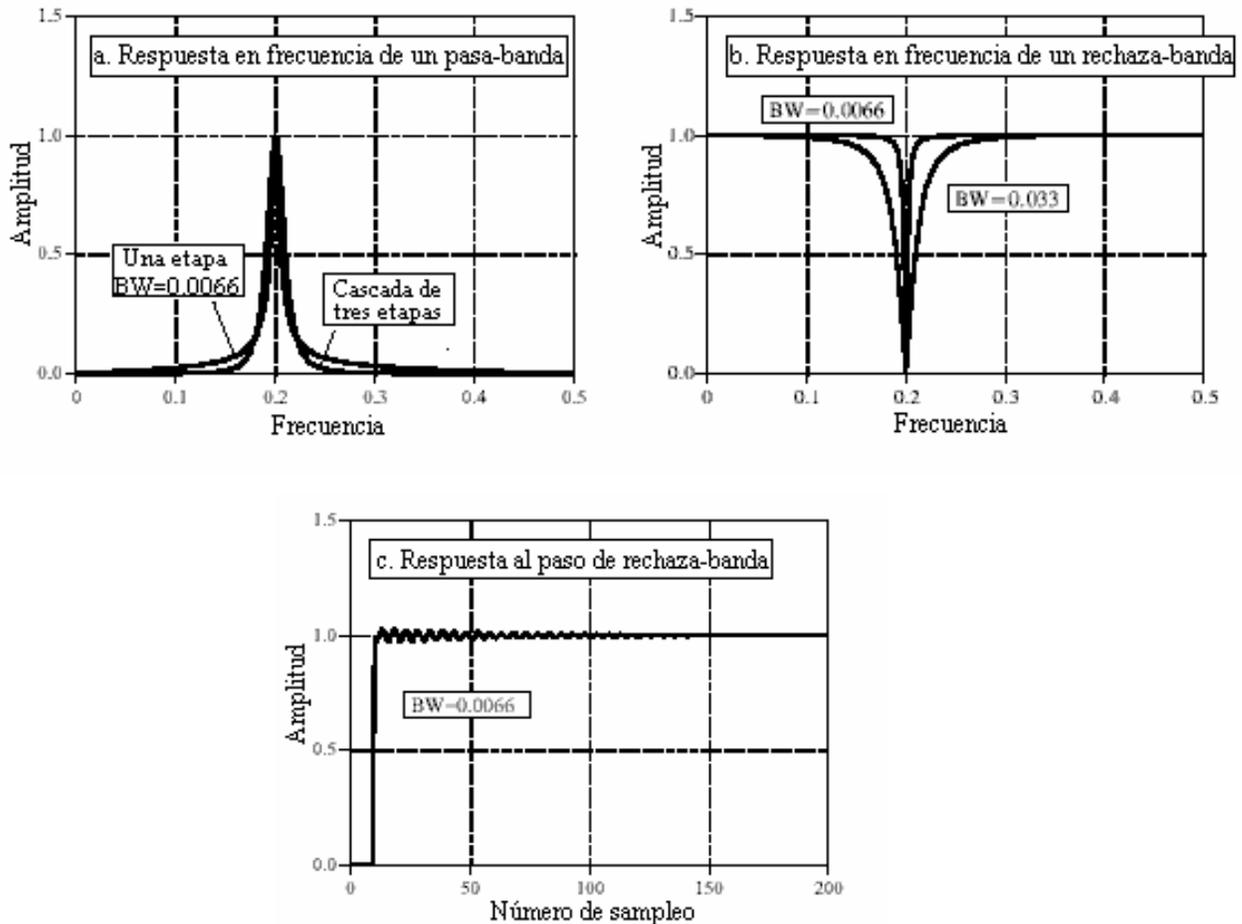


Figura 48: Respuestas en frecuencia de filtros de banda estrecha, implementados por recursión.

4.5.3 Respuestas de Fase de filtros los digitales

Existen tres respuestas de fase que cualquier filtro digital puede tener: **fase cero**, **fase lineal** y **fase no lineal**. Un ejemplo de cada uno de estos se muestra en la figura 49. Como se muestra en la figura 49(a), un filtro de fase cero se caracteriza por una respuesta al impulso que es simétrica respecto a cero. En este caso los términos negativos son el reflejo de los términos positivos. Cuando la transformada de Fourier se

aplica a este tipo de onda simétrica, el resultado será que la fase es cero, como se muestra en 49(b).

La desventaja de un filtro de fase cero es que al tener índices negativos, produce inconvenientes para su manipulación a nivel de programación. El filtro de **fase lineal** es un camino alternativo para esto. La respuesta al pulso mostrada en la figura 49(d) es idéntica a la mostrada en la figura 49(a), sólo que ha sido trasladada de modo que solamente utiliza términos positivos. La respuesta al pulso es poco menos simétrica entre la izquierda y la derecha, y de todas formas, la localización de simetría ha sido trasladada desde cero.

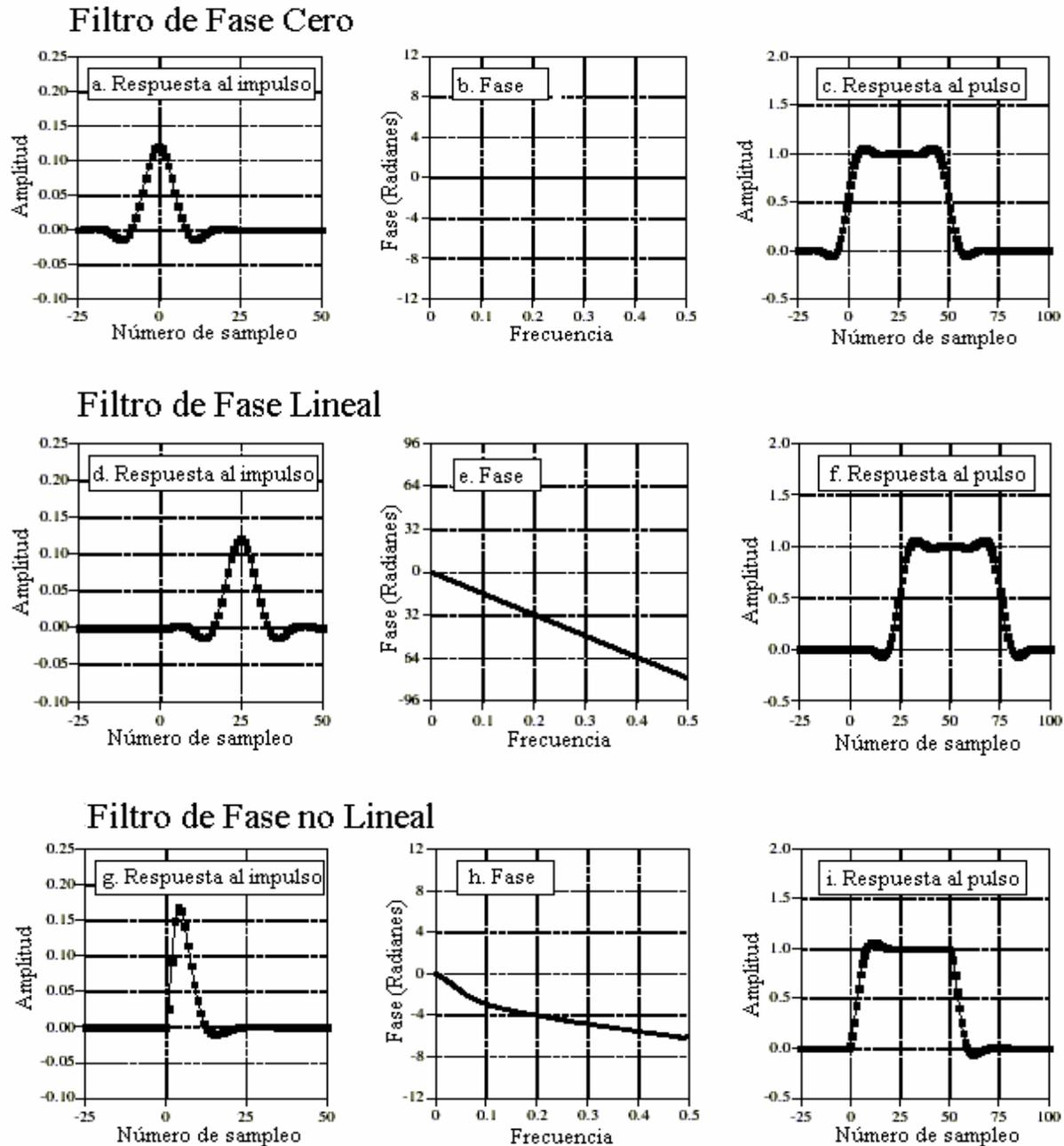


Figura 49: Características de un filtro de fase cero, lineal y no lineal respectivamente.

De estas traslaciones resultará un cambio en la fase que serán acordes a una *línea recta*, de ahí que sea de *fase lineal*. La pendiente de esta línea recta es directamente proporcional a la cantidad de la traslación. La traslación no afecta a la respuesta al pulso en cuanto a su forma, pero produce una traslación en ella. Por esta razón, el filtro de fase lineal es equivalente al filtro de fase cero para la mayor parte de los propósitos.

La figura 49(g) muestra una respuesta al impulso que no es simétrica entre la izquierda y la derecha, correspondientemente, la fase, en 49(h) no es una línea recta. En otras palabras, tiene una *fase no lineal*.

La importancia de que la fase sea lineal o no radica en lo siguiente: Las figuras 49(c), 49(f) y 49(i) muestran lo que se conoce como **respuesta al pulso** de cada uno de estos tres filtros. La respuesta al pulso no es más que la respuesta al paso hacia el lado positivo, seguida de la respuesta al paso hacia el lado negativo. La respuesta al pulso es útil porque esta muestra que ocurre con el rising y falling en la señal. Aquí viene la parte importante: Los filtros de fase cero y fase lineal tienen bordes izquierdo y derecho que se ven semejantes, mientras que los filtros de fase no lineal tienen bordes izquierdo y derecho que se ven diferentes. Algunas aplicaciones no pueden tolerar que los bordes izquierdo y derecho sean diferentes. Un ejemplo de esto se ve en el procesamiento de video: si los bordes fuesen distintos, ocurriría que una misma imagen vista con el ojo izquierdo se vería distinta con el ojo derecho.

Para un filtro FIR es sencillo que este tenga una respuesta de fase lineal. Esto es porque la respuesta a impulso (filtro Kernel) está directamente *especificada* en el proceso de diseño. Hacer que el filtro Kernel tenga simetría derecha-izquierda es todo lo que se requiere. Este no es el caso de los filtros IIR pues lo que hay que especificar son los coeficientes de recursión y no la respuesta al impulso. La respuesta al impulso de un filtro recursivo no es simétrica entre la izquierda y la derecha, y por lo mismo su respuesta de fase es *no lineal*.

Los circuitos analógicos y digitales tienen el mismo problema con la respuesta de fase. Imaginemos un circuito compuesto por resistores y capacitores. Si la entrada es siempre cero entonces la salida también será siempre cero. Cuando se aplica un impulso en la entrada, el capacitor rápidamente cambia de valor y entonces comienza un decaimiento exponencial hacia los resistores. La respuesta al impulso (esto es, la señal de salida) es una combinación de todos estos decaimientos exponenciales. La respuesta al impulso no puede ser simétrica porque la salida será cero después del impulso y el decaimiento exponencial nunca consigue llegar al valor cero. El diseño de filtros analógicos ataca este problema con los llamados **Filtros Bessel** [SMI1997]. Los filtros Bessel son diseñados para tener en lo posible una fase lineal. Por lo mismo, estos están muy por debajo del rendimiento de los filtros digitales. La capacidad de proveer una fase lineal exacta es una clara ventaja que poseen los filtros digitales FIR.

Afortunadamente, existe un método sencillo para modificar los filtros recursivos y obtener una fase cero. La figura 50 muestra como este trabaja. La señal de entrada que será filtrada se muestra en 50(a). La figura 50(b) muestra la señal después de haber sido filtrada por el filtro Single Pole pasa-bajo. Puesto que este es un filtro de fase no lineal, los bordes izquierdo y derecho no se muestran iguales, y estos son cada uno la versión invertida del otro.

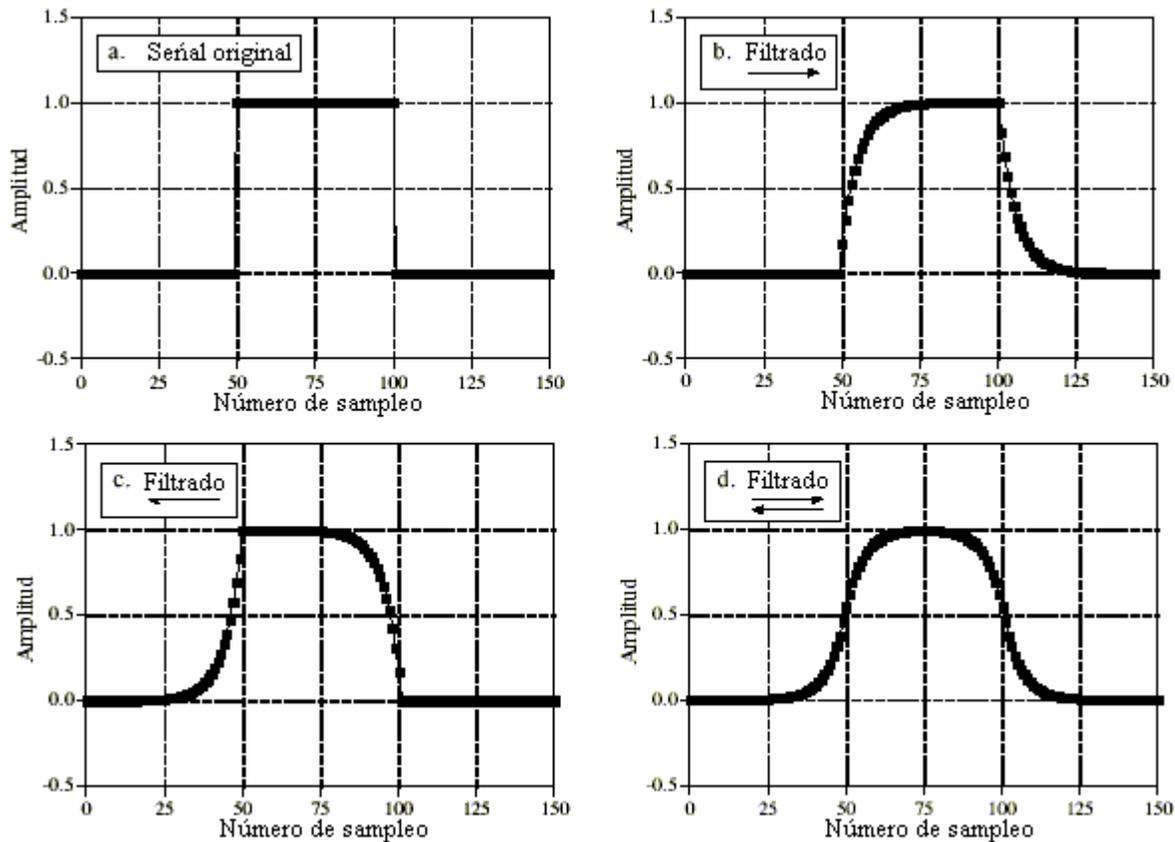


Figura 50: La técnica del filtrado bi-direccional provee una manera de lograr que un filtro recursivo tenga fase lineal.

Analicemos la situación inversa. Si contamos ahora hacia atrás, cada punto de la señal de salida es calculado a partir de la entrada y del término de salida de la derecha. Esto significa que la ecuación de recursión queda de la siguiente manera:

$$y[n] = a_0x[n] + a_1x[n+1] + a_2x[n+2] + \dots + b_1y[n+1] + b_2y[n+2] + \dots$$

La figura 50(c) muestra el resultado de este **Filtro revertido**.

Filtrar en la dirección invertida no produce ningún resultado en sí mismo, pues los bordes izquierdo y derecho por sí solos no tienen algo que los prefiera más a uno de ellos. Lo útil ocurre cuando el filtro normal y el revertido son *combinados*. La figura 50 (d) muestra tal combinación. Esto producirá un filtro recursivo de *fase cero*. En efecto, cualquier filtro recursivo puede ser transformado en un filtro de fase cero con la técnica de **filtrado bidireccional**. La única desventaja de esta mejora de rendimiento del filtro

es un incremento en el tiempo de ejecución, el cual será en un factor de dos. La magnitud de la respuesta en frecuencia es la misma para cada dirección. Mientras que las fases son opuestas en signo. Cuando las dos direcciones se combinan, la magnitud llega a ser cuadrada, mientras que las fases se cancelan y será cero.

CAPÍTULO 5: WAVELETS Y FILTROS DIGITALES

5.1 BANCO DE FILTROS

En lo presente, veremos que los coeficientes $h(n)$ y $h_1(n)$ que se encuentran en las ecuaciones:

$$\varphi(t) = \sum_n h(n) \sqrt{2} \varphi(2t-n) \quad (5.1)$$

$$\psi(t) = \sum_n h_1(n) \sqrt{2} \varphi(2t-n), \quad (5.2)$$

se comportan como filtros digitales.

Podemos notar que la siguiente ecuación corresponde a una expansión de $g(t)$ en términos de una función escala y su función wavelet:

$$g(t) = \sum_{k=-\infty}^{\infty} c(k) \varphi_k(t) + \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} d(j,k) \psi_{j,k}(t), \quad (5.3)$$

$$c(k) = c_0(k) = \langle g(t), \varphi_k(t) \rangle = \int g(t) \varphi_k(t) dt, \quad (5.4)$$

$$d_j(k) = d(j,k) = \langle g(t), \psi_{j,k}(t) \rangle = \int g(t) \psi_{j,k}(t) dt. \quad (5.5)$$

El desarrollo de los Bancos de filtros no se relaciona en forma directa con las funciones escala y wavelet, sino que con los coeficientes relacionados con éstas, esto es $h(n)$ y $h_1(n)$.

5.1.1 ANÁLISIS

En este tema, trataremos la relación entre los coeficientes de expansión a un bajo nivel de escala en términos de aquellos coeficientes de una escala más alta.

Consideremos la ecuación de recursión básica (5.1):

$$\varphi(t) = \sum_n h(n) \sqrt{2} \varphi(2t-n).$$

Asumiendo que existe una única solución, trasladamos y escalamos la variable tiempo y nos queda:

$$\varphi(2^j t - k) = \sum_n h(n) \sqrt{2} \varphi(2(2^j t - k) - n). \quad (5.6)$$

Es decir,

$$\varphi(2^j t - k) = \sum_n h(n) \sqrt{2} \varphi(2^{j+1} t - 2k - n). \quad (5.7)$$

Ahora usamos la sustitución $m=2k+n$, o $n=m-2k$:

$$\varphi(2^j t - k) = \sum_n h(m - 2k) \sqrt{2} \varphi(2^{j+1} t - m). \quad (5.8)$$

Si denotamos a V_j como:

$$V_j = \text{Span}_k \{2^{j/2} \varphi(2^j t - k)\}. \quad (5.9)$$

Entonces, $f(t) \in V_{j+1}$ y puede ser expresado como:

$$f(t) = \sum_k c_{j+1}(k) 2^{(j+1)/2} \varphi(2^{j+1} t - k). \quad (5.10)$$

La cual es claramente expresable en una escala $j+1$ con funciones escala solamente, y no wavelets.

En una escala de baja resolución, las wavelets son necesarias para el “detalle” no abordable en una escala j .

Luego:

$$f(t) = \sum_k c_j(k) 2^{j/2} \varphi(2^j t - k) + \sum_k d_j(k) 2^{j/2} \psi(2^j t - k), \quad (5.11)$$

donde los términos $2^{j/2}$ mantienen la norma unitaria de las funciones base en escalas diferentes.

Si $\varphi_{j,k}(t)$ y $\psi_{j,k}(t)$ son ortonormales, entonces, los coeficientes de escalamiento j son hallados tomando el producto interno:

$$c_j(k) = \langle f(t), \varphi_{j,k}(t) \rangle = \int f(t) 2^{j/2} \varphi(2^j t - k) dt. \quad (5.12)$$

Para lo cual, si usamos (5.8) e intercambiamos la sumatoria por integral, nos queda:

$$c_j(k) = \sum_m h(m - 2k) \int f(t) 2^{(j+1)/2} \varphi(2^{j+1} t - m) dt. \quad (5.13)$$

Pero la integral es un producto interno con un escalamiento de la función en el nivel $j+1$, entonces:

$$c_j(k) = \sum_m h(m - 2k) c_{j+1}(m). \quad (5.14)$$

Un trabajo análogo muestra que la relación de los coeficientes wavelet es [BUR1998]:

$$d_j(k) = \sum_m h_1(m - 2k) c_{j+1}(m). \quad (5.15)$$

Estas operaciones dadas en las ecuaciones (5.14) y (5.15) corresponden a una convolución discreta. La secuencia de entrada dada por $c_{n,k}$ es convolucionada con h y h_1 . Esta estructura ya nos da un indicio de que h y h_1 corresponden a un filtro figital implementado por convolución.

De hecho, si comparamos estas estructuras con la definición de convolución discreta, vemos claramente que corresponden a filtros digitales de tipo FIR [BUR1998]:

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

5.1.1.1 Filtrado y Down-Sampling o Decimación

Dos operaciones básicas en el tema de filtros que son el Down-Sampler y el Up-sampler.

El Down –Sampler (*DS: llamado también Sampler o decimador*), toma una señal $x(n)$ como una entrada, y produce una salida $y(n)=x(2n)$.

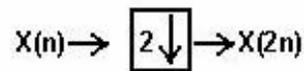


Figura 51: Esquema simbólico de la decimación por dos de una entrada digital x .

En algunos casos, el DS es un factor distinto de dos, y en otros casos, la salida puede ser $y(n)=x(2n+1)$, pero estos casos serán explícitamente establecidos sólo si ello es necesario.

En el DS existe claramente la posibilidad de perder información, pues la mitad de los datos se descartan. El efecto en el dominio de la frecuencia (a través de la transformada de Fourier) es llamado “Aliasing” [RIC1975], el cual establece que la pérdida de esta información conlleva a una superposición de las componentes de frecuencia. Si la señal es de banda limitada en el dominio de la frecuencia (la mitad de los coeficientes de Fourier son cero), no habrá pérdida de información causada por el DS.

Las ecuaciones (5.12) y (5.13) producen precisamente un filtrado digital y DS en forma simultánea. Estas ecuaciones muestran que los coeficientes escala y wavelet en diferentes niveles pueden ser obtenidos convolucionando los coeficientes de expansión en la escala j con los coeficientes de recursión $h(-n)$ y $h_1(-n)$ que son los DS o decimadores (tomando los términos pares), entregando los coeficientes de expansión en el nivel $j-1$. En otras palabras, los coeficientes c_j son “filtrados” por dos filtros digitales FIR, con coeficientes $h(-n)$ y $h_1(-n)$, respectivamente, después de lo cual el DS entrega los coeficientes escala y wavelet en una escala superior.

La implementación de las ecuaciones (5.12) y (5.13) se muestra a continuación:

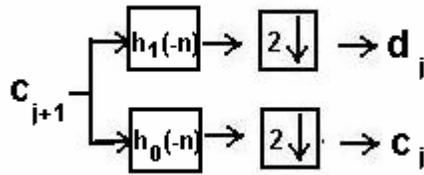


Figura 52: Esquema de un banco de filtros de una etapa.

El “ $2 \downarrow$ ” denota DS o decimación por dos. “ $h_1(-n)$ ” y “ $h_0(-n)$ ” denotan filtrado FIR o una convolución con $h_1(-n)$ o $h_0(-n)$, respectivamente.

El número de términos en la salida se duplica al tener dos filtros, y esto es reducido a la mitad por la decimación, regresando al número original. Esto significa que existe la posibilidad que la información no halla sido perdida y sería posible recuperar la señal original.

El Aliasing ocurre en el banco superior y no el banco inferior, que es donde se descompone la señal. Esta es la idea de la “reconstrucción perfecta” en la teoría del Banco de Filtros. Este proceso puede repetirse varias veces y se conoce como *Iteración de Banco de Filtros* [BUR1998].

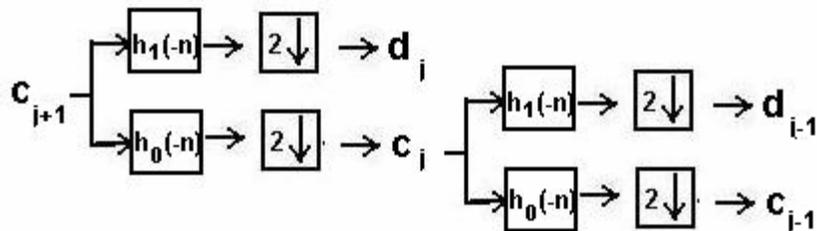


Figura 53: Esquema de Banco de filtros de dos etapas

La respuesta en Frecuencia de un filtro digital es la Transformada Discreta de Fourier en el tiempo (DFT), de la respuesta al impulso $h(n)$:

$$H(\omega) = \sum_{n=-\infty}^{\infty} h(n)e^{i\omega n} . \quad (5.16)$$

La magnitud de esta función compleja entrega la razón entre la salida y la entrada del filtro para un término sinusoidal con una frecuencia ω rad/s. El ángulo de $H(\omega)$ corresponde a la fase entre la salida y la entrada.

La primera etapa de dos bancos, divide el espectro de $c_{j+1}(k)$ por medio de una banda pasa-bajo y una banda pasa-alto, resultando un coeficiente escala $c_j(k)$ y un coeficiente wavelet $d_j(k)$, respectivamente. El siguiente paso, divide esta banda pasa-bajo entre un pasa-bajo y un pasa-banda.

La primera etapa divide el espectro en dos partes iguales. La segunda etapa divide la mitad más baja en dos cuartos iguales. De este proceso resulta un set de anchos de banda que se ilustra en la figura 54.

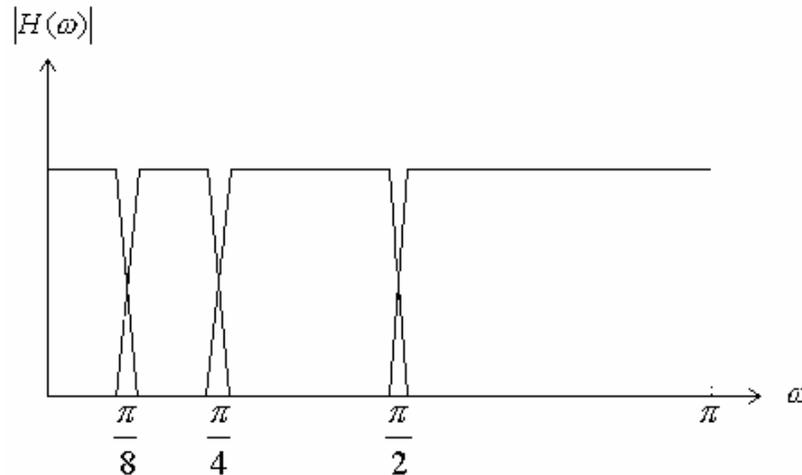


Figura 54: División del espectro de frecuencia al pasar por un banco de filtros.

Estos son llamados filtros “constante Q” en el lenguaje de Banco de Filtros [BUR1998], porque el valor del ancho de banda para la frecuencia central de la banda es constante. Es algo interesante notar que una escala musical define octavas en una forma similar y que el oído responde a la frecuencia de manera logarítmica.

Para cualquier señal que se encuentra limitada en banda, esta tendrá una escala alta $j=J$, sobre la cual los coeficientes Wavelet d_j son pequeños.

Para comenzar con una descripción en alta resolución de una señal en términos de los coeficientes escala c_j , el análisis de árbol de la DWT requerido para bajar a una resolución $j=j_0$ estará compuesta de $J-j_0$ etapas.

Para $f(t) \in V_j$ y recordando que:

$$f(t) = \sum a_k \varphi(2^j t - k);$$

tendremos:
$$f(t) = \sum c_j(k) \varphi_{j,k}(t) \quad (5.17)$$

$$= \sum_k c_{J-1} \varphi_{J-1,k}(t) + \sum_k d_{J-1}(k) \psi_{J-1,k}(t) \quad (5.18)$$

$$f(t) = \sum_k c_{J-2}(k) \varphi_{J-2,k}(t) + \sum_k \sum_{j=J-2}^{J-1} d_j(k) \psi_{j,k}(t) \quad (5.19)$$

$$f(t) = \sum_k c_{j_0}(k) \varphi_{j_0,k}(t) + \sum_k \sum_{j=j_0}^{J-1} d_j(k) \psi_{j,k}(t), \quad (5.20)$$

lo cual es una versión finita de (5.1). Esto es:

$$g(t) = \sum_k c_{j_0}(k) \varphi_{j_0,k}(t) + \sum_k \sum_{j=j_0}^{\infty} d_j(k) \psi_{j,k}(t).$$

5.1.2 SÍNTESIS

La síntesis o reconstrucción de la señal original con coeficientes de escala fina, puede ser hecha desde la combinación de las funciones escala y coeficientes wavelet en una resolución más densa.

Consideremos una señal en la escala $j+1$ del espacio de funciones con $f(t)$ en V_{j+1} . La función puede ser descrita en términos de la función escala por la ecuación (5.10):

$$f(t) = \sum_k c_{j+1}(k) 2^{(j+1)/2} \varphi(2^{j+1}t - k),$$

o en términos de la siguiente escala (la cual requiere wavelets), como:

$$f(t) = \sum_k c_j(k) 2^{j/2} \varphi(2^j t - k) + \sum_k d_j(k) 2^{j/2} \psi(2^j t - k). \quad (5.21)$$

Sustituyendo (5.1) y (5.2) en (5.21), nos queda:

$$f(t) = \sum_k c_j(k) \sum_n h(n) 2^{(j+1)/2} \varphi(2^{j+1}t - 2k - n) + \sum_k d_j(k) \sum_n h_1(n) 2^{(j+1)/2} \varphi(2^{(j+1)/2}t - 2k - n) \quad (5.22)$$

Debido a que estas funciones son ortonormales, multiplicando (5.10) y (5.22) por $\varphi(2^{j+1}t - k)$, e integrando los coeficientes, nos queda:

$$c_{j+1}(k) = \sum_m c_j(m)h(k - 2m) + \sum_m d_j(m)h_1(k - 2m) . \quad (5.23)$$

Esta ecuación es la encargada de pasar una señal desde una escala más densa hasta otra más fina.

5.1.2.1 FILTRADO Y UP-SAMPLING O STRECHING

Para síntesis en el banco de filtros, tendremos una primera secuencia del primer up-sampling. Esto significa que la entrada del filtro tiene ceros insertados entre cada uno de los términos originales. En otras palabras:

$$Y(2n)=x(n) \text{ y } y(2n+1)=0, \quad (5.24)$$

donde la señal de entrada es incrementada al doble del largo de la señal original al añadir ceros. Claramente este up-sampling o estiramiento puede ser hecho con otros factores distintos de dos y estas dos ecuaciones pueden tener el 0 o $x(n)$ intercambiados. Tenemos entonces que el up-sampler no produce ninguna pérdida de información. Si hacemos un up-sampler y luego el down-sampler, regresaremos al mismo punto de partida. Pero si se hace viceversa, no se garantiza volver al mismo lugar de partida.

5.1.3 INTERPRETACIÓN DE LA TRANSFORMADA WAVELET EN TÉRMINOS DE BANCOS DE FILTROS

La aplicación de la DWT a la secuencia de entrada producirá una descomposición del espectro en bandas. Tal división es hecha por un filtro pasa-bajos y un filtro pasa-altos. De hecho, el número de bandas que aparece en la descomposición es directamente proporcional al número de veces que se aplique la transformada a una secuencia en forma iterativa.

Consideremos el caso en que la DWT es aplicada sólo una vez. Entonces, si c_{j+1} representa el espectro de la secuencia de entrada, este es descompuesto en un pasa-bajos y un pasa-altos.

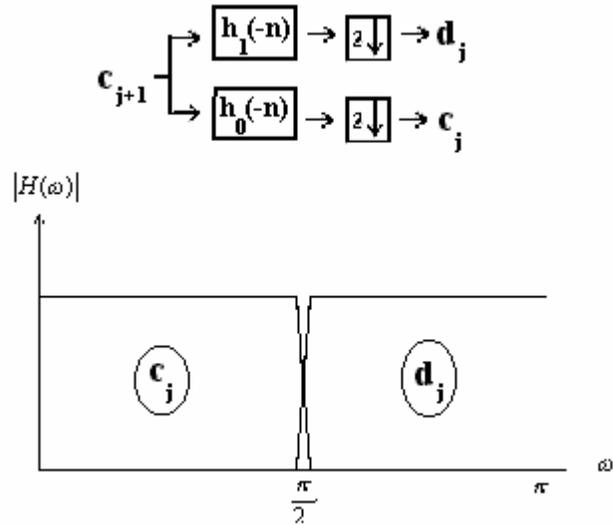


Figura 55: Esquema de banco de filtros y representación en la frecuencia para una etapa.

Al aplicar la DWT por segunda vez, el lado izquierdo del espectro es dividido nuevamente por los filtros $h(n)$ y $h_1(n)$, con lo que quedan un pasa-bajos y un pasa-banda.

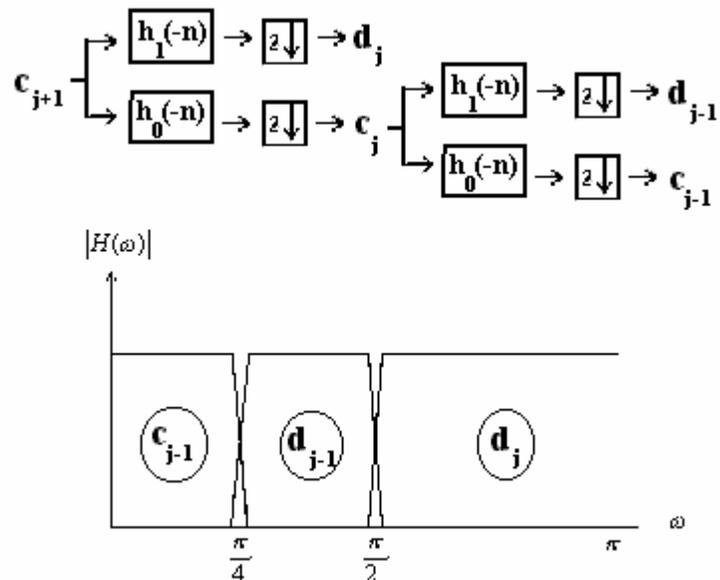


Figura 56: Esquema de banco de filtros y representación en la frecuencia para dos etapas.

Un proceso similar ocurre al aplicar por tercera vez la DWT, como se indica en la figura 57.

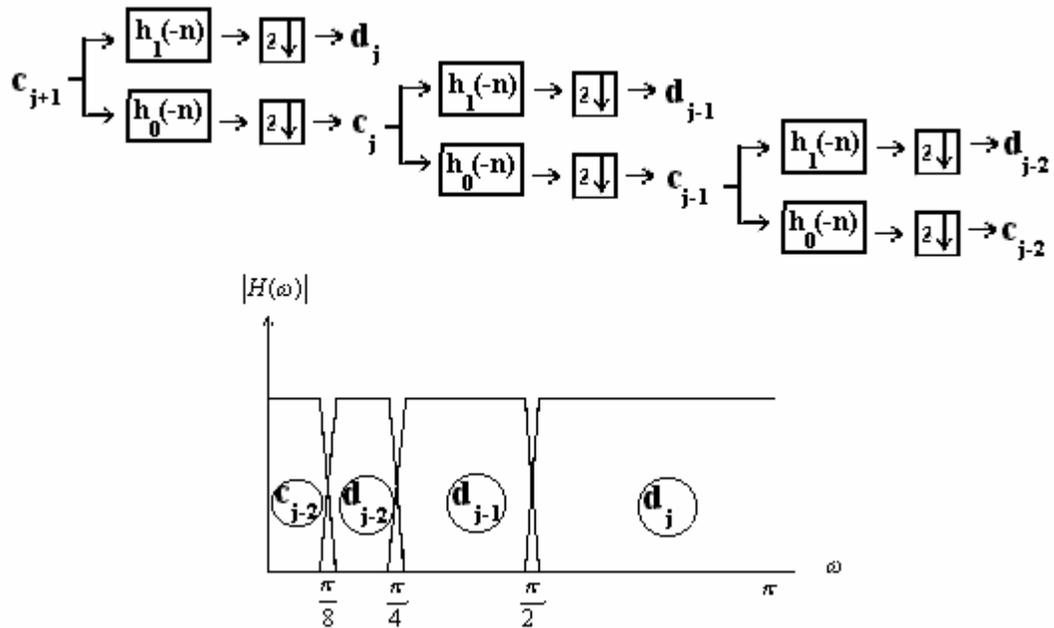


Figura 57: Esquema de banco de filtros y representación en la frecuencia para tres etapas.

Esta interpretación de la DWT en términos de filtros es importante pues permite comprender el uso de la transformada para filtrar parte de una señal no deseada. Por ejemplo, si la secuencia de entrada corresponde a una señal de audio que contiene ruido en las frecuencias altas podríamos asumir lo siguiente, en el caso de usar la DWT en tres etapas:

$$\begin{aligned} c_{j-2} &= c_{j-2} \\ d_{j-2} &= d_{j-2} \\ d_{j-1} &= d_{j-1} \\ d_j &= 0 \end{aligned}$$

Lo cual gráficamente se indica en la figura 58:

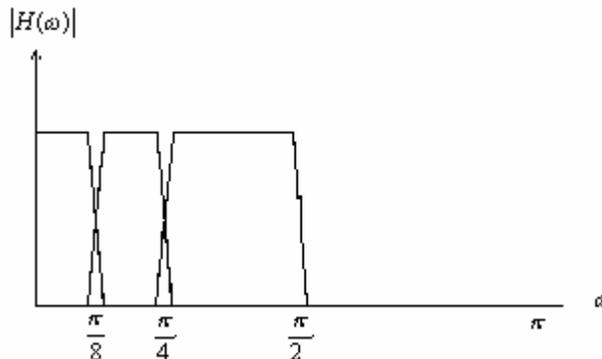


Figura 58: Esquema de la aplicación de un banco de filtros para eliminar parte de una señal.

Es de esperar que al aplicar la transformada inversa wavelet (IDWT) se recuperará la señal original, pero ahora sin ruido. Este procedimiento es igualmente aplicable a todos los filtros por convolución [SMI1997].

5.2 CONDICIONES DE SUFICIENCIA

Las siguientes condiciones corresponden a teoremas sobre admisibilidad de un sistema Wavelet dada una secuencia $h(n)$. Aquí se mostrarán las matemáticas que aseguran que $h(n)$ y $h_1(n)$ corresponden a un filtro pasa-bajo y pasa-alto vistos desde el dominio del tiempo y la frecuencia.

5.2.1 TRANSFORMACIONES DE FOURIER

Consideremos la Transformada de Fourier de $\varphi(t)$, la cual, si existe está definida por:

$$\Phi(\omega) = \int_{-\infty}^{\infty} \varphi(t)e^{-j\omega t} dt, \quad (5.25)$$

y la transformada discreta de Fourier de $h(n)$:

$$H(\omega) = \sum_{-\infty}^{\infty} h(n)e^{-j\omega n}, \quad (5.26)$$

donde j es la unidad imaginaria y n es un entero. Si la convolución con $h(n)$ es vista como un filtro digital, entonces la DTFT de $h(n)$ es la respuesta en frecuencia del filtro, el cual tiene periodo 2π .

Si $\Phi(\omega)$ existe, la ecuación recursiva (5.1) $\varphi(t) = \sum_n h(n)\sqrt{2} \varphi(2t-n)$ nos entrega:

$$\Phi(\omega) = \frac{1}{\sqrt{2}} H(\omega/2)\Phi(\omega/2), \quad (5.27)$$

la cual, después de ser iterada:

$$\Phi(\omega) = \prod_{k=1}^{\infty} \left\{ \frac{1}{\sqrt{2}} H\left(\frac{\omega}{2^k}\right) \right\} \Phi(0). \quad (5.28)$$

La ecuación (5.28) es válida siempre que $\sum_n h(n) = \sqrt{2}$ y que $\Phi(0)$ esté bien definido.

Esta puede ser una distribución, o una función suave, dependiendo de $H()$ y por ende, de $h(n)$.

Aunque (5.1) y (5.28) es equivalentes término a término, el requisito que $\Phi(0)$ esté bien definido y la naturaleza de los límites apropiados en el espacio de funciones, pueden hacer preferible una de la otra. Notemos que los ceros de $H()$ determinan los ceros de $\Phi(\omega)$.

5.2.2 CONDICIONES NECESARIAS. DOMINIO DEL TIEMPO

Teorema 1: Si $\varphi(t) \in L^1$ es una solución de la ecuación de recursión básica (5.1), y sí $\int \varphi(t)dt \neq 0$, entonces:

$$\sum_n h(n) = \sqrt{2}. \quad (5.29)$$

La demostración de este teorema requiere solamente un intercambio en el orden de la sumatoria e integración (permitido en L^1), pero no supone la ortogonalidad de las funciones bases o cualquier otra propiedad de $\varphi(t) \in L^1$, excepto que la integral sea distinta de cero.

Este teorema muestra que para diferentes ecuaciones diferenciales lineales de coeficiente constantes, no sólo un set de coeficientes supondrán una solución. Los coeficientes también deben satisfacer la ecuación (5.29). Esta es la condición más delicada de los $h(n)$.

Teorema 2

Si $\varphi(t) \in L^1$ es solución de la ecuación básica (5.1) con $\int \varphi(t)dt = 1$ y sí además

$$\sum_{\ell} \varphi(t - \ell) = \sum_{\ell} \varphi(\ell) = 1, \quad (5.30)$$

con $\Phi(\pi + 2\pi k) \neq 0$ para algún k , entonces:

$$\sum_n h(2n) = \sum_n h(2n+1). \quad (5.31)$$

Por otro lado, si (5.31) es satisfecho, entonces (5.30) es cierto.

La ecuación (5.31) es llamada la condición fundamental y esta es delicada pues requiere ortogonalidad, y es más fuerte que (5.29). Este es un simple resultado de requerimiento que resulta de evaluar (5.1).

La ecuación (5.30) es llamada Partición de la Unidad (o la condición fuerte o condición de Shoemberg) [BUR1998].

Teorema 3

Si $\varphi(t)$ está en $L^2 \cap L^1$ y es solución de (5.1) y las traslaciones enteras de $\varphi(t)$ definidas por:

$$\int \varphi(t)\varphi(t-k)dt = \begin{cases} E & \text{si } k = 0 \\ 0 & \text{si } k \neq 0 \end{cases}, \quad (5.32)$$

son ortogonales, entonces:

$$\sum_n h(n)h(n-2k) = \begin{cases} 1 & \text{si } k = 0 \\ 0 & \text{si } k \neq 0 \end{cases}. \quad (5.33)$$

Este teorema muestra que para que $\varphi(t)$ sea ortogonal por traslación entera es necesario que los coeficientes de la ecuación recursiva para $h(n)$ sean ortogonales después de una decimación o down-sampling por dos.

Los coeficientes que satisfacen (5.27) son llamados Quadrature mirror filters (QMF) o Conjugate Mirror Filters (CMF) [BUR1998], y la condición (5.33) es llamada **condición cuadrática**.

Corolario 1

Bajo la suposición del teorema 3, la norma de $h(n)$ es automáticamente unitaria:

$$\sum_n |h(n)|^2 = 1. \quad (5.34)$$

Esta normalización unitaria de $h(n)$ es el resultado de la $\sqrt{2}$ de la ecuación (5.1).

Corolario2

Bajo la suposición del teorema 3:

$$\sum_n h(2n) = \sum_n h(2n+1) = \frac{1}{\sqrt{2}}. \quad (5.35)$$

No sólo la suma de todos los términos debe ser igual a $\sqrt{2}$, sino que las sumas de los términos pares e impares, individualmente deben sumar $1/\sqrt{2}$, independientemente de cualquier normalización de $\varphi(t)$.

Teorema 4

Si $\varphi(t)$ tiene soporte compacto en $0 \leq t \leq N-1$ y $\varphi(t-k)$ son linealmente independientes, entonces $h(n)$ también tiene soporte compacto sobre $0 \leq n \leq N-1$, y:

$$h(n)=0 \text{ si } n < 0 \text{ o } n > N-1 \text{ (causalidad)}. \quad (5.36)$$

Si las traslaciones no son linealmente independientes (o tienen alguna restricción equivalente), uno puede tener $h(n)$ con soporte infinito mientras que $\varphi(t)$ tiene soporte finito.

Estos teoremas establecen que si $\varphi(t)$ tiene soporte compacto y es ortogonal sobre traslaciones enteras, la condición cuadrática (5.33) debe satisfacer en adición a la ecuación lineal (5.29).

El soporte o largo de $h(n)$ es N , el cual debe ser un número par. El número de grados de libertad para escoger estos N coeficientes es $N/2 - 1$. Esta libertad se usará en el *diseño de sistemas wavelet*.

5.2.3 CONDICIONES NECESARIAS. DOMINIO DE LA FRECUENCIA

Aquí se mostrarán las condiciones necesarias en el dominio de la frecuencia para la existencia de $\varphi(t)$. Algunas están especificadas sobre el espacio de funciones sobre la que la transformación de Fourier opera, y otras en el espacio donde la transformación reside [BUR1998].

Teorema 5

Si $\varphi(t)$ es solución en L^1 de la ecuación de recursión básica (5.1) entonces las siguientes condiciones equivalentes deben ser ciertas:

$$\sum_n h(n) = H(0) = \sqrt{2}. \quad (5.37)$$

Esta viene directamente de (5.26) (Transformación de Fourier discreta) y establece que el requerimiento básico de existencia (5.29) es equivalente al requerimiento que la respuesta en frecuencia de un filtro FIR para $\omega=0$ (DC) sea $\sqrt{2}$.

Teorema 6

Para $h(n)$ en L^1 , entonces:

$$\sum_n h(2n) = \sum_n h(2n+1) \text{ si y solo si } H(\pi)=0. \quad (5.38)$$

La cual dice que la respuesta en frecuencia del filtro FIR con respuesta al impulso $h(n)$ es cero, para la llamada frecuencia de Nyquist ($\omega=\pi$). Esto resulta de (5.27) y (5.37) y sostiene que el término $h(n)$ es un filtro digital pasa-bajo.

Teorema 7

Si $\varphi(t)$ es una solución de (5.1) en $L^2 \cap L^1$ y $\Phi(\omega)$ es una solución de (5.27), tal que $\Phi(0) \neq 0$, entonces:

$$\int \varphi(t)\varphi(t-k)dt = \delta(k) \text{ si y solo si } \sum_\ell |\Phi(\omega + 2\pi\ell)|^2 = 1. \quad (5.39)$$

Teorema 8

Para cualquier $h(n)$ en L^1 :

$$\sum_n h(n)h(n-2k) = \delta(k) \text{ si y solo si.} \quad (5.40)$$

Este teorema entrega condiciones equivalentes en los dominios del tiempo y frecuencia para los coeficientes escala y establece que el requerimiento de ortogonalidad (5.33) es equivalente a un filtro FIR con los $h(n)$ siendo llamados QMF [HAR1998].

Notar que por (5.39), (5.37) y el teorema 6, resulta que $H(\pi/2)=1$ y que el filtro es de "media banda".

5.2.4 CONDICIONES SUFICIENTES

Estas son las condiciones necesarias para que $\varphi(t)$ exista.

Teorema 9

Si $\sum_n h(n) = \sqrt{2}$ y $h(n)$ tiene soporte finito o decae rápidamente, tal que $\sum |h(n)|(1+|n|)^\varepsilon < \varepsilon$ para algún $\varepsilon > 0$, entonces existe un único $\varphi(t)$ que satisface (5.1) y (5.28).

Esta puede ser obtenida en el dominio de la frecuencia, considerando la convergencia de (5.28). Esto recientemente ha sido obtenido, usando una aproximación mucho más poderosa en el dominio del tiempo, por Lawton[BUR1998].

La función escala obtenida sólo requiere que $\sum_n h(n) = \sqrt{2}$.

Teorema 10

Si $\sum_n h(2n) = \sum_n h(2n+1) = \frac{1}{\sqrt{2}}$ y $h(n)$ tiene soporte finito o decae lo suficientemente rápido, tal que $\sum |h(n)|(1+|n|)^\varepsilon < \infty$ para algún $\varepsilon > 0$, entonces existe un único $\varphi(t)$ que está bien definido en los racionales diádicos. Además, la suma:

$$\sum_k \varphi(t-k) = 1, \text{ se mantiene.}$$

Esta condición es llamada también la condición fundamental y sólo entrega un pequeño cambio del teorema 9.

Teorema 11

Si $h(n)$ tiene un soporte finito o decae lo suficientemente rápido, y sí:

$\sum_n h(n) = \sqrt{2}$ y $\sum_n h(n)h(n-2k) = \delta(k)$, entonces $\varphi(t) \in L^2$ existe, y genera un sistema wavelet que es una estructura ajustada en L^2 .

La demostración usa una iteración de la ecuación de recursión básica (5.1), como una aproximación sucesiva similar al teorema de Picard para las ecuaciones diferenciales[EDW2001]. Este método es usado para calcular $\varphi(t)$ en (5.29). Es interesante notar que la función escala puede ser muy densa como un "fractal". Esto puede ser deseable si la señal que está siendo analizada también es densa.

Aunque este teorema garantiza que $\varphi(t)$ genera una estructura ajustada, en situaciones prácticas, el sistema resultante es una base ortonormal.

Teorema 12

Si $H(\omega)$ es la DTFT de $h(n)$ con soporte compacto, y si $\sum_n h(n) = \sqrt{2}$ con $\sum_n h(n)h(n-2k) = \delta(k)$; y si $H(\omega) \neq 0$ para $-\pi/3 \leq \omega \leq \pi/3$, entonces $\varphi(t-k)$ que satisface (5.1) genera una base ortonormal en L^2 .

Una versión más simple de esta condición de suficiencia en el dominio de la frecuencia es fácil de obtener y de extender al caso de M-bandas, para ceros no permitidos en $-\pi/2 \leq \omega \leq \pi/2$.

Estas son otras condiciones de suficiencia que en conjunto al teorema 11, garantizan una base ortonormal. Por otro lado, los teoremas 5, 6 y 12 muestran que $h(n)$ tienen las características de un filtro digital FIR pasa-bajo.

5.3 DISEÑO DE WAVELETS

5.3.1 MÉTODO DE LOS MOMENTOS CEROS DE DAUBECHIES

Filtros escala k-regulares

Comenzaremos definiendo un *filtro escala unitario* como un filtro FIR cuya secuencia será $h(n)$, a partir de la ecuación de recursión básica (5.1) satisfaciendo las condiciones de admisibilidad desde (5.29) y condiciones de ortogonalidad de (5.33) como:

$$\sum_n h(n) = \sqrt{2} \text{ y } \sum_k h(k)h(k+2m) = \delta(m). \quad (5.41)$$

Un filtro escala unitario o USF (*Unitary Scaling Filter*) se dice que es k-regular, si su transformada z tiene k ceros para $z = e^{i\pi}$. Tal función de transferencia está dada por:

$$H(z) = \left(\frac{1+z^{-1}}{2} \right)^k Q(z) \quad (5.42)$$

Donde $H(z)$ es la transformada Z de la secuencia $h(n)$ y $Q(z)$ no tiene polos y ceros para $z = e^{i\pi}$. Esto corresponde a una definición de regularidad de $h(n)$, y no de la función escala $\varphi(t)$, o de la wavelet $\psi(t)$. Ambas funciones están relacionadas pero no son lo mismo. Del teorema 6 podemos observar que cualquier filtro unitario escala es a lo menos $k=1$ regular.

La longitud del filtro escala es N , lo cual significa que $H(z)$ es un polinomio de $N-1$ grados de libertad. Puesto que el múltiplo cero para $z=-1$ es de orden k , el polinomio $Q(z)$ tiene $N-1-k$ grados de libertad. La existencia de $\varphi(t)$ requiere que la sumatoria de los términos de $h(n)$ sea $\sqrt{2}$, lo cual es el resultado de la condición lineal del teorema 9. Satisfaciendo las condiciones de ortogonalidad se requieren $N/2$ ecuaciones, las cuales están dadas por (5.40). Esto significa que el grado de regularidad está limitado por:

$$1 < k < N/2.$$

Daubechies usa el grado de libertad para obtener la máxima regularidad para un N fijo, o para obtener el mínimo N dada una regularidad. En otras situaciones, además del diseño de sistemas wavelet, se ha permitido una pequeña regularidad para usar el grado extra de libertad resultante con otros propósitos de diseño [BUR1998].

La regularidad está definida en términos de los ceros de la función de transferencia o de la función respuesta en frecuencia de un filtro FIR creado a partir de los coeficientes escala. Esto está relacionado con el hecho de que la diferenciabilidad de la función está ligada a la rapidez con que los coeficientes de la serie de Fourier decaen cuando el índice tiende a infinito, o, análogamente, a la rapidez con que la magnitud de la transformada de Fourier decae cuando la frecuencia tiende a infinito.

La relación de la transformada de Fourier de $\varphi(t)$ y la respuesta en frecuencia del filtro FIR cuya secuencia es $h(n)$ está dada por el producto infinito (5.28). Desde estas conexiones podemos suponer que $H(z)$ es un pasa-bajo y, si este tiene un alto orden cero para $z=-1$ (esto es, $\omega=\pi$), la transformada de Fourier de $\varphi(t)$ podrá decaer rápidamente y, entonces $\varphi(t)$ podrá ser suave.

5.3.1.1 Momentos

Los k -ésimos momentos parciales de $\varphi(t)$ y $\psi(t)$, están definidos por:

$$m(k) = \int t^k \varphi(t) dt \quad (5.43)$$

$$m_1(k) = \int t^k \psi(t) dt. \quad (5.44)$$

Los k -ésimos momentos discretos de $h(n)$ y $h_1(n)$ están definidos por:

$$\mu(k) = \sum_n n^k h(n) \quad (5.45)$$

$$\mu_1(k) = \sum_n n^k h_1(n). \quad (5.46)$$

Los momentos parciales de $h(n)$ (momentos de términos individuales), son definidos como:

$$\nu(k, l) = \sum_n (2n + l)^k h(2n + l). \quad (5.47)$$

Notar que $\mu(k) = \nu(k, 0) + \nu(k, 1)$.

De estas ecuaciones y de la ecuación de recursión básica (5.1) obtenemos:

$$m(k) = \frac{1}{(2^k - 1)\sqrt{2}} \sum_{l=1}^k \binom{k}{l} \mu(l) m(k-l), \quad (5.48)$$

Análogamente, para la wavelet tenemos:

$$m_1(k) = \frac{1}{2^k \sqrt{2}} \sum_{l=0}^k \binom{k}{l} \mu_1(l) m(k-l). \quad (5.49)$$

Estas ecuaciones permiten calcular con exactitud los momentos definidos por las integrales en (5.42) y (5.43) como simples convoluciones finitas de los momentos discretos y los momentos continuos de bajo orden.

5.3.1.2 Momentos Wavelet desvanecidos

El requerimiento de que los momentos $\psi(t)$ sean cero tiene interesantes consecuencias. Los siguientes tres teoremas muestran una variedad de características equivalentes de los filtros escala k -regulares, los cuales relacionan ambos deseos: el de obtener funciones escala suaves y por otro lado, wavelets como una representación polinomial.

Teorema (caracterizaciones equivalentes de filtros k -regulares)

Un filtro unitario es k -regular si y sólo si las siguientes afirmaciones son ciertas [BUR1998]:

1. Todos los momentos de los filtros wavelet son cero, $\mu_1(k) = 0$.
2. Todos los momentos de las wavelets son cero, $m_1(k) = 0$.
3. Los momentos parciales de los filtros escala son iguales para $k=0, 1, \dots, (K-1)$.
4. Las respuestas en frecuencia de los filtros escala tienen un cero de orden K para $\omega = \pi$.
5. La k -ésima derivada del cuadrado de la magnitud de la respuesta en frecuencia del filtro escala es cero para $\omega = 0$ si $k=1, 2, \dots, 2K-1$.
6. Todas las secuencias polinomiales bajo el grado $(K-1)$ pueden ser expresadas como una combinación lineal de filtros escala trasladados.
7. Todos los polinomios de grado menores a $(K-1)$ pueden ser expresados como una combinación lineal de funciones escala trasladadas de una única escala.

Teorema. Si $\Psi(t)$ es k veces diferenciable y decae rápidamente lo suficiente, entonces los primeros $(K-1)$ momentos wavelet se desvanecen, esto es:

$$\left| \frac{d^k}{dt^k} \psi(t) \right| < \infty, \quad 0 \leq k \leq K,$$

implica que:

$$m_1(k) = 0 \quad 0 \leq k \leq K.$$

Estos teoremas muestran la relación existente entre los momentos de $h_1(n)$, $\Psi(t)$, la suavidad de $H(\omega)$ para $\omega=0$ y $\omega=\pi$ y la representación polinomial. Esto establece también una pérdida de la relación entre la suavidad de $\varphi(t)$ y $\psi(t)$.

5.3.1.3 Método de diseño del momento Wavelet cero de Daubechies

Daubechies ha usado las relaciones anteriores para mostrar el siguiente e importante resultado que construye wavelets ortonormales suaves con soporte compacto con el máximo orden de momentos desvanecidos.

Teorema (Daubechies). *La transformada discreta de Fourier de $h(n)$ que tiene k ceros en $\omega=\pi$ de la forma:*

$$H(\omega) = \left(\frac{1 + e^{i\omega}}{2} \right)^k \ell(\omega), \quad (5.50)$$

satisface:
$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = 2, \quad (5.51)$$

si y solo si $L(\omega) = |\ell(\omega)|^2$ es de la forma:

$$L(\omega) = P(\text{sen}^2(\omega/2)), \quad (5.52)$$

con $K \leq N/2$, donde:

$$P(y) = \sum_{k=0}^{K-1} \binom{K-1+k}{k} y^k + y^K R\left(\frac{1}{2} - y\right) \quad (5.53)$$

y $R(y)$ es un polinomio escogido tal que $P(y) \geq 0$ para $0 \leq y \leq 1$.

Si $R=0$, el largo N es mínimo para una regularidad $K=N/2$ dada. Si $N > 2K$, el segundo término que contiene R contiene altas potencias de y cuyos coeficientes pueden ser usados para otros propósitos además de la regularidad.

Desde la ecuación (5.40) notamos que $H(\omega)$ también tiene a lo menos un cero para $\omega=\pi$ como un resultado de que $h(n)$ satisface las condiciones necesarias para que $\varphi(t)$ exista y tenga traslaciones ortogonales enteras. Esto nos da una restricción para $h(n)$ donde el cero de orden más alto posible se encuentra para $\omega=\pi$.

Los requerimientos en el dominio de la frecuencia de (5.51) están dados en términos del cuadrado de la magnitud de la respuesta en frecuencia. Aquí se utiliza la técnica de la *factorización espectral* para encontrar $H(\omega)$ y luego, $h(n)$ [SMI1997].

De la ecuación (5.50) obtenemos:

$$|H(\omega)|^2 = \left| \frac{1 + e^{i\omega}}{2} \right|^{2K} |\ell(\omega)|^2. \quad (5.54)$$

Si usamos la notación funcional:

$$M(\omega) = |H(\omega)|^2 \text{ y } L(\omega) = |\ell(\omega)|^2, \quad (5.55)$$

entonces tendremos que:

$$M(\omega) = \left| \cos^2(\omega/2) \right|^K L(\omega) \quad (5.56)$$

Si $M(\omega)$ y $L(\omega)$ son funciones pares de ω , ellas pueden ser escritas como polinomios de $\cos(\omega)$, y usando la identidad $\cos(\omega) = 1 - 2\text{sen}^2(\omega/2)$ queda:

$$\tilde{M}(\text{sen}^2(\omega/2)) = \left| \cos^2(\omega/2) \right|^K P(\text{sen}^2(\omega/2)), \quad (5.57)$$

la cual, después del cambio de variable $y = \text{sen}^2(\omega/2)$ nos da:

$$\tilde{M}(y) = (1-y)^K P(y), \quad (5.58)$$

donde $P(y)$ es un polinomio de orden $(N-K)$, el cual debe ser positivo puesto que habrá de ser factorizado para poder hallar $H(\omega)$ a partir de (5.55)

Esto nos entrega (5.50) en términos de nuevas variables, las cuales son fáciles de usar. De esta manera, requerimos que (5.58) satisfaga (5.40), esto es:

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = 2,$$

de la cual, usando (5.58) y (5.40) obtenemos la siguiente ecuación:

$$M(\omega) + M(\omega + \pi) = (1-y)^K P(y) + y^K P(1-y) = 2. \quad (5.59)$$

Las ecuaciones de esta forma se pueden solucionar explícitamente a través del teorema de Bézout [SER2000]. Si todos los $(N/2-1)$ grados de libertad son usados como un set de momentos wavelet, todos ceros, hallamos que debe cumplirse que $K=N/2$ y que la solución de (5.59) está dada por:

$$P(y) = \sum_{k=0}^{K-1} \binom{K-1+k}{k} y^k. \quad (5.60)$$

Esta última expresión entrega una parametrización completa del método de diseño maximizado del momento wavelet cero. También nos entrega una forma muy directa para el cálculo de $h(n)$ que satisface éstas condiciones.

Los pasos para calcular los valores actuales de $h(n)$ son estos: primero se escoge el largo N (para una regularidad deseada) de $h(n)$, luego, el factor $|H(\omega)|^2$ donde existe la libertad de escoger que cuadrados usar para hallar $H(\omega)$.

Los cálculos son muy fáciles de hacer usando la transformada z desde la función de transferencia y usando convolución en el dominio del tiempo.

5.3.2 WAVELETS COIFLETS

5.3.2.1 Desvanecimiento de los momentos de la función escala

Mientras que los momentos de las wavelets entregan información acerca de la planidad de $H(\omega)$, y suavidad de $\psi(t)$, los momentos de $\varphi(t)$ y de $h(n)$ son medidas de la “localización” y características de la simetría de la función escala y, entonces, de la transformada wavelet. Sabemos de (5.29) que $\sum_n h(n) = \sqrt{2}$ y que, después de normalizar, que $\int \varphi(t) dt = 1$. Usando (5.48) puede demostrarse que para $K \geq 2$:

$$m(2) = m^2(1). \quad (5.61)$$

5.3.2.2 Aproximación de funciones mediante la proyección de la función escala

La proyección ortogonal de una señal $f(t)$ en el subespacio de la función escala V_j esta dada y denotada por:

$$P^j \{f(t)\} = \sum_k \langle f(t), \varphi_{j,k}(t) \rangle \varphi_{j,k}(t) \quad (5.62)$$

la cual entrega la componente de $f(t)$ que esta en V_j , y que es la mejor aproximación de $f(t)$ en V_j .

Como se definió antes, el ℓ -ésimo momento de $\psi(t)$ está definido por:

$$m_1(\ell) = \int t^\ell \psi(t) dt \quad (5.63)$$

Ahora podemos establecer una importante relación de la proyección (5.62) como una aproximación de $f(t)$ en términos del número de momentos wavelet cero y la escala.

Teorema

Si $m_1(\ell) = 0$ para $\ell = 0, 1, \dots, L$; entonces, el error en L^2 es:

$$\varepsilon_1 = \|f(t) - P^j \{f(t)\}\| \leq C 2^{-j(L+1)}, \quad (5.64)$$

donde C es una constante independiente de j y L , pero que si depende de $f(t)$ y del sistema wavelet escogido.

5.3.2.3 Aproximación de coeficientes escala por términos de la señal

Una segunda aproximación utiliza los términos de $f(t)$ en vez de los coeficientes de los productos internos de la expansión wavelet de $f(t)$ en (5.62). Definiremos y denotaremos esta expansión por:

$$S^j \{f(t)\} = \sum_k 2^{-j/2} f(k/2^j) \varphi_{j,k}(t), \quad (5.65)$$

y el momento de la función escala por:

$$m(\ell) = \int t^\ell \varphi(t) dt$$

Teorema

Si $m(\ell) = 0$ para $\ell = 0, 1, \dots, L$ entonces el error en L^2 está dado por:

$$\varepsilon_2 = \|S^j \{f(t)\} - P^j \{f(t)\}\| \leq C_2 2^{-j(L+1)}, \quad (5.66)$$

donde C_2 es una constante independiente de j y L , pero dependiente de $f(t)$ y del sistema wavelet.

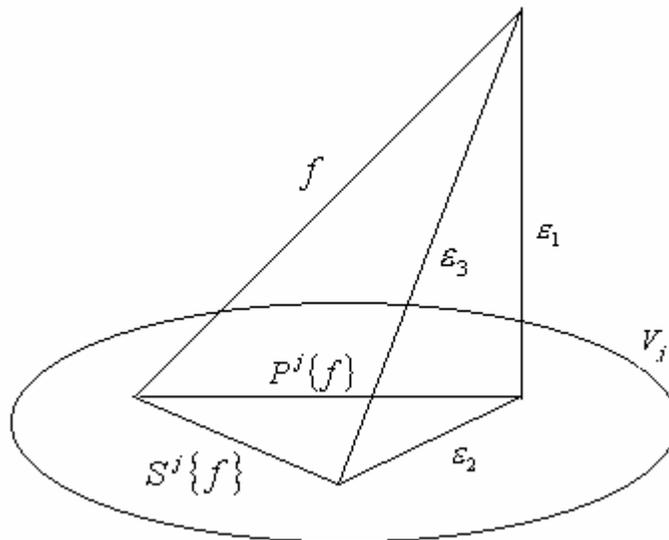


figura 59: Representación geométrica de f y sus aproximaciones en el espacio de funciones.

La figura 59 ilustra el “espacio vectorial de funciones” y muestra la naturaleza y relación entre los dos tipos de aproximaciones. El uso de muestros de la señal como productos internos es una aproximación de $f(t)$ en el subespacio V_j . Se puede utilizar una expansión finita para representar una señal $f(t)$ como una aproximación desde L^2 dentro del subespacio V_j . Los dos teoremas anteriores muestran la naturaleza de estas aproximaciones, las cuales, para wavelets, es muy buena.

Si consideramos un sistema wavelet donde el número L de momentos de las funciones escala y wavelet son cero, y este número es lo más grande posible, entonces lo siguiente es cierto:

Teorema

Si $m(\ell) = 0$ para $\ell = 0, 1, \dots, L$, entonces el error en L^2 está dado por:

$$\varepsilon_3 = \|f(t) - S^j \{f(t)\}\| \leq C_3 2^{-j(L+1)}, \quad (5.67)$$

donde C_3 es una constante independiente de j y L ; pero dependiente de $f(t)$ y del sistema wavelet.

Un sistema definido de esta manera (como muestra el teorema) recibe el nombre de *Sistema wavelet Coifman* [BUR1998], el cual usa sampleos de la señal en vez de los productos internos, obteniendo una excelente aproximación.

5.3.2.4 Coiflets

Como se dijo recién, la combinación de momentos cero de las funciones escala y wavelet usando términos de la señal puede entregar mejores resultados para wavelets que tener únicamente el momento wavelet igual a cero. Forzar los momentos de la función escala a cero no sólo entrega una mejor aproximación de los coeficientes de expansión por términos de la señal, sino que además hace que la función escala sea más simétrica. Ciertamente, en ciertas aplicaciones esta característica puede ser más importante que la aproximación por sampleos [BUR1998].

El sistema wavelet Coifman (Daubechies llamó a estas funciones base “coiflets”), es un sistema wavelet de multiresolución ortonormal [BUR1998], dado por:

$$m(k) = \int t^k \varphi(t) dt = 0 \quad (5.68)$$

$$m_1(k) = \int t^k \psi(t) dt = 0; k=1, 2, \dots, L-1.$$

Esta definición implica el requerimiento de que sean a lo menos $L-1$ momentos de las funciones escala igual a cero y a lo menos $L-1$ momentos wavelet sean ceros en adición al único momento wavelet $m_1(0)$ requerido para ortogonalidad. En este sistema se ve que es de orden o grado L y algunas veces tiene el requerimiento adicional que la longitud N del filtro de la función $h(n)$, sea mínimo.

5.3.3 SIMLETS

Con excepción del sistema Haar, ningún sistema Wavelet puede tener soporte compacto y ser simétrico a la vez. Para algunos propósitos prácticos, como por ejemplo, en procesamiento de imágenes, se puede tratar de obtener un cierto grado de simetría con el fin de lograr que la fase de $h(n)$ sea mínima, sin que cambie el valor de su norma $|h(n)|$ [HAR1998].

Propiedades de las Simlets.

Las funciones wavelet “Padre” y “Madre” cumplen con:

1. $supp \varphi \subseteq [0, 2N - 1]$
2. $supp \psi \subseteq [-N + 1, N]$
3. $m_1(\ell) = \int t^\ell \psi(t) dt = 0, \ell = 0, 1, \dots, N-1.$ (5.69)

Se debe notar que las Simlets no son simétricas.

CAPÍTULO 6: PARTE EXPERIMENTAL

6.1 MÉTODO DE DISEÑO DE WAVELETS MEDIANTE FILTROS DIGITALES FIR

6.1.1 Proposición y desarrollo del problema

En lo que sigue, consideraremos una secuencia de coeficientes a_1, \dots, a_N que corresponden a un filtro FIR. Entonces esta secuencia debe satisfacer la siguiente función de transferencia [PRO1996] :

$$H(z) = a_0 + a_1 z^1 + a_2 z^2 + \dots + a_N z^N . \quad (6.1)$$

Esta ecuación corresponde a la transformada Z de la secuencia del filtro. Si Z es la variable compleja, esta la expresaremos por:

$$z = r e^{j\omega} , \quad (6.2)$$

por lo que nos queda:

$$H(z) = a_0 + a_1 r e^{j\omega} + a_2 r^2 e^{j2\omega} + \dots + a_N r^N e^{jN\omega} . \quad (6.3)$$

Si evaluamos para $r=1$ obtendremos la DTFT del filtro. Esto es:

$$H(\omega) = a_0 + a_1 e^{j\omega} + a_2 e^{j2\omega} + \dots + a_N e^{jN\omega} . \quad (6.4)$$

Ahora bien, en el capítulo de condiciones de suficiencia y admisibilidad, vimos que una secuencia $h(n)$ admite un sistema wavelet, si la función de transferencia cumple con:

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = 2 . \quad (6.5)$$

Para, la función de transferencia será:

$$H(\omega + \pi) = a_0 - a_1 e^{j\omega} + a_2 e^{j2\omega} - a_3 e^{j3\omega} + \dots \pm a_N e^{jN\omega} . \quad (6.6)$$

En forma separada, los cuadrados de los módulos de estas funciones complejas son, respectivamente:

$$\begin{aligned} |H(\omega)|^2 &= (a_0 + a_1 \cos \omega + a_2 \cos 2\omega + \dots + a_N \cos N\omega)^2 + (a_1 \sin \omega + \dots + a_N \sin N\omega)^2 \\ |H(\omega + \pi)|^2 &= (a_0 - a_1 \cos \omega + a_2 \cos 2\omega + \dots \pm a_N \cos N\omega)^2 + (a_1 \sin \omega + \dots \pm a_N \sin N\omega)^2 . \end{aligned} \quad (6.7)$$

Recordando que el cuadrado de un polinomio está dado por:

$$(a_1 + a_2 + \dots + a_N)^2 = \sum a_i^2 + 2 \sum_{i \neq j} a_i a_j. \quad (6.8)$$

Entonces:

$$\begin{aligned} |H(\omega)|^2 + |H(\omega + \pi)|^2 &= 2a_0^2 + 2\{a_1^2 \cos^2 \omega + a_2^2 \cos^2 2\omega + \dots + a_N^2 \cos^2 N\omega\} + \\ &+ 2\{a_1^2 \sen^2 \omega + a_2^2 \sen^2 2\omega + \dots + a_N^2 \sen^2 N\omega\} + \\ &+ 4\{a_0 a_2 \cos 2\omega + a_0 a_4 \cos 4\omega + \dots + a_2 a_4 \cos 2\omega \cos 4\omega + \dots + a_{2i} a_{2j} \cos 2i\omega \cos 2j\omega\} + \\ &+ 4\{a_1 a_3 \cos \omega \cos 3\omega + a_1 a_5 \cos \omega \cos 5\omega + \dots + a_{2i+1} a_{2j+1} \cos(2i+1)\omega \cos(2j+1)\omega\} + \\ &+ 4\{a_2 a_4 \sen 2\omega \sen 4\omega + a_2 a_6 \sen 2\omega \sen 6\omega + \dots + a_{2i} a_{2j} \sen 2i\omega \sen 2j\omega\} + \\ &+ 4\{a_1 a_3 \sen \omega \sen 3\omega + a_1 a_5 \sen \omega \sen 5\omega + \dots + a_{2i+1} a_{2j+1} \sen(2i+1)\omega \sen(2j+1)\omega\}. \end{aligned} \quad (6.9)$$

Observemos que hay agrupaciones de índices pares e impares. Usando la identidad

$\sen^2 \omega + \cos^2 \omega = 1$ y definiendo las agrupaciones pares por $P\{\cos\}, P\{\sen\}$ y las impares por $I\{\cos\}, I\{\sen\}$:

$$P\{\cos\} = \{a_0 a_2 \cos 2\omega + a_0 a_4 \cos 4\omega + \dots + a_{2i} a_{2j} \cos 2i\omega \cos 2j\omega + \dots\} \quad (6.10)$$

$$I\{\cos\} = \{a_1 a_3 \cos \omega \cos 3\omega + a_1 a_5 \cos \omega \cos 5\omega + \dots\} \quad (6.11)$$

$$P\{\sen\} = \{a_2 a_4 \sen 2\omega \sen 4\omega + a_2 a_6 \sen 2\omega \sen 6\omega + \dots\} \quad (6.12)$$

$$I\{\sen\} = \{a_1 a_3 \sen \omega \sen 3\omega + a_1 a_5 \sen \omega \sen 5\omega + \dots\}, \quad (6.13)$$

Sustituyendo en (6.10), (6.11), (6.12) y (6.13) en (6.9) nos queda:

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = 2a_0^2 + 2 \sum_1^N a_k^2 + 4P\{\cos\} + 4I\{\cos\} + 4P\{\sen\} + 4I\{\sen\}. \quad (6.14)$$

Agrupando los pares, notamos que:

$$\begin{aligned} P\{\cos\} + P\{\sen\} &= a_0 (a_2 \cos 2\omega + a_4 \cos 4\omega + \dots + a_N \cos N\omega) + \\ &+ \{a_2 a_4 (\cos 2\omega \cos 4\omega + \sen 2\omega \sen 4\omega) + a_2 a_6 (\cos 2\omega \cos 6\omega + \sen 2\omega \sen 6\omega) + \dots\} \end{aligned}$$

Usando la identidad $\cos(A - B) = \cos B \cos A - \text{sen} B \text{sen} A$ y asociando términos; obtenemos:

$$P\{\cos\} + P\{\text{sen}\} = a_0 \sum_k a_{2k} \cos 2k\omega + (a_2 a_4 + a_4 a_6 + a_6 a_8 + \dots) \cos 2\omega + \\ + (a_2 a_6 + a_4 a_8 + a_6 a_{10} + \dots) \cos 4\omega + \dots + (a_2 a_N \cos(N-2)\omega) \quad (6.15)$$

Análogamente, agrupando los términos impares:

$$I\{\cos\} + I\{\text{sen}\} = (a_1 a_3 + a_3 a_5 + a_5 a_7 + \dots) \cos 2\omega + (a_1 a_5 + a_3 a_7 + a_5 a_9 + \dots) \cos 4\omega + \\ + (a_1 a_7 + a_3 a_9 + a_5 a_{11} + \dots) \cos 6\omega + \dots + (a_1 a_{N-1} \cos(N-2)\omega) \quad (6.16)$$

Remplazando en la condición 1 (6.5):

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = 2a_0^2 + 2 \sum_1^N a_k^2 + 4a_0 \sum_k a_{2k} \cos 2k\omega + \\ + 4\{ (a_2 a_4 + a_4 a_6 + a_6 a_8 + \dots) \cos 2\omega + \dots + (a_2 a_N \cos(N-2)\omega) \} + \\ + 4\{ (a_1 a_3 + a_3 a_5 + \dots) \cos 2\omega + \dots + (a_1 a_{N-1} \cos(N-2)\omega) \}. \quad (6.17)$$

Ahora, definamos las constantes:

$$A_1 = (a_2 a_4 + a_4 a_6 + a_6 a_8 + \dots) + (a_1 a_3 + a_3 a_5 + a_5 a_7 + \dots) \\ A_2 = (a_2 a_6 + a_4 a_8 + a_6 a_{10} + \dots) + (a_1 a_5 + a_3 a_7 + a_5 a_9 + \dots) \\ A_3 = (a_2 a_8 + a_4 a_{10} + a_6 a_{12} + \dots) + (a_1 a_7 + a_3 a_9 + a_5 a_{11} + \dots), \text{ en general:} \\ A_{(N-2)/2} = (a_2 a_N) + (a_1 a_{N-1}). \quad (6.18)$$

Entonces:

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = 2a_0^2 + 2 \sum_1^N a_k^2 + 4a_0 \sum_k a_{2k} \cos 2k\omega + \sum_{k=1}^{(N-2)/2} A_k \cos(2k\omega). \quad (6.19)$$

Puesto que se requiere que $|H(\omega)|^2 + |H(\omega + \pi)|^2 = 2$, debemos preguntarnos ¿Qué condiciones deben cumplir los coeficientes A_k y a_0 ?

Para hacer el trabajo más sencillo, asumamos que $a_0=0$. Entonces (6.5) queda:

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = 2 \sum_1^N a_k^2 + \sum_{k=1}^{(N-2)/2} A_k \cos(2k\omega). \quad (6.20)$$

El término más complicado resulta ser la combinación lineal de cosenos pares.

Usando las técnicas de la serie de Fourier [HSU1973], vemos que la sumatoria será una constante sí y sólo sí:

$$A_k=0, \text{ para todo } k \text{ entero.}$$

Por lo tanto, la única solución posible es:

$$\sum_1^N a_k^2 = 1 \text{ y } A_k=0, \text{ para todo } k \text{ entero.} \quad (6.21)$$

Ejemplos.

Consideremos una secuencia de longitud $N=12$, entonces, el sistema a resolver será:

$$A_1 = (a_2 a_4 + a_4 a_6 + a_6 a_8 + a_8 a_{10} + a_{10} a_{12}) + (a_1 a_3 + a_3 a_5 + a_5 a_7 + a_7 a_9 + a_9 a_{11}) = 0$$

$$A_2 = (a_2 a_6 + a_4 a_8 + a_6 a_{10} + a_8 a_{12}) + (a_1 a_5 + a_3 a_7 + a_5 a_9 + a_7 a_{11}) = 0$$

$$A_3 = (a_2 a_8 + a_4 a_{10} + a_6 a_{12}) + (a_1 a_7 + a_3 a_9 + a_5 a_{11}) = 0$$

$$A_4 = (a_2 a_{10} + a_4 a_{12}) + (a_1 a_9 + a_3 a_{11}) = 0$$

$$A_5 = (a_2 a_{12}) + (a_1 a_{11}) = 0; \text{ y}$$

$$\sum_1^{12} a_k^2 = 1.$$

Revisemos algunos sistemas wavelet clásicos para verificar la validez de las condiciones encontradas:

Ejemplo 1. Para el filtro wavelet Daubechies 12 (Apéndice C), y con una aproximación de 7 decimales, tenemos que:

$$A_1 = 0.1115000 - 0.1113136 = 0.0001864$$

$$A_2 = -0.0556000 + 0.0484600 = -0.0071400$$

$$A_3 = 0.01393 - 0.01393 = 0.0000000$$

$$A_4 = -0.0005109 + 0.0005109 = 0.0000000$$

$$A_5 = 0.0000299 - 0.0000299 = 0.0000000$$

$$\sum_1^{12} a_k^2 = 1.00000025$$

Ejemplo 2. Los filtros de Coifman, o Coiflets de orden dos, $C2$ y $N=12$:

$$A_1 = 0.0378566 - 0.037895 = -0.0000384$$

$$A_2 = 0.0043306 - 0.0043306 = 0.0000000$$

$$A_3 = 0.0005109 - 0.0005109 = 0.0000000$$

$$A_4 = 0.0000299 - 0.0000299 = 0.0000000$$

$$\sum_1^{12} a_k^2 = 1.00000039$$

Ejemplo 3. Para un filtro de longitud $N=8$, el sistema será:

$$A_1 = (a_2 a_4 + a_4 a_6 + a_6 a_8) + (a_1 a_3 + a_3 a_5 + a_5 a_7) = 0$$

$$A_2 = (a_2 a_6 + a_4 a_8) + (a_1 a_5 + a_3 a_7) = 0$$

$$A_3 = (a_2 a_8) + (a_1 a_7) = 0; y$$

$$\sum_1^8 a_k^2 = 1$$

Para un filtro wavelet Simlet $S4$ de longitud $N=8$, los resultados obtenidos son:

$$A_1 = -0.1035658 + 0.1067631 = 0.0031973$$

$$A_2 = 0.0288394 - 0.0263212 = 0.0025182$$

$$A_3 = -0.0009550 + 0.0009550 = 0.0000000$$

$$\sum_1^8 a_k^2 = 1.00000084.$$

Podemos ver que el error para los A_k es del orden de 10^{-2} como el máximo error, lo cual se puede considerar aceptable tal vez para varias aplicaciones.

6.1.2 Comparación del método con antecedentes teóricos

Si observamos la estructura de los A_k podemos ver que una forma compacta de expresar estos coeficientes es:

$$\sum_n h(n)h(n-2k) = \begin{cases} 1 & \text{si } k = 0 \\ 0 & \text{si } k \neq 0 \end{cases}$$

que corresponde al resultado o consecuencia de la existencia y ortogonalidad de $\varphi(t)$ dada en el Teorema 3 (5.33) sobre condiciones de suficiencia. Claramente de esta ecuación vemos que los $h(n)$ son ortogonales después de una decimación por dos [BUR1998].

Por otro lado, en este modelo ya planteamos la condición de que la norma de $h(n)$ sea unitaria. Observemos que esto se desprende también de la ecuación (5.33) cuando $k=0$.

Además, como corolario del teorema 3, tenemos que:

$$\sum_n h(2n) = \sum_n h(2n+1) = \frac{1}{\sqrt{2}},$$

lo cual también es satisfecho por todos los filtros vistos en los ejemplos recientes.

6.1.3 Síntesis del proceso de diseño

Resumiendo, las ecuaciones con que contamos para hallar los coeficientes de un filtro $h(n)$ que admite un sistema wavelet son las que aparecen en el siguiente sistema:

$$\sum_n h(n)h(n-2k) = \begin{cases} 1 & \text{si } k = 0 \\ 0 & \text{si } k \neq 0 \end{cases}, \quad (6.22)$$

$$\sum_n h(2n) = \sum_n h(2n+1) = \frac{1}{\sqrt{2}}. \quad (6.23)$$

Una manipulación algebraica de (6.22) y (6.23) nos muestra que la norma unitaria del filtro es dependiente de (6.22) para el caso k distinto de cero, y de (6.23). Esto es particularmente útil pues reduce el número de ecuaciones no lineales en el sistema. Este queda de la siguiente forma:

$$\begin{aligned} \sum_n h(n)h(n-2k) &= \delta(k) = 0, k \neq 0, \\ \sum_n h(2n) &= \frac{1}{\sqrt{2}}, \\ \sum_n h(2n+1) &= \frac{1}{\sqrt{2}}. \end{aligned} \quad (6.24)$$

Si el largo del filtro es N , entonces tendremos un sistema con $N/2+1$ ecuaciones linealmente independientes.

Puesto que el sistema de ecuaciones (6.24) no es cuadrado, existe una libertad de grado $N/2-1$ para escoger términos. Esta libertad es útil para adecuar el filtro a las especificaciones de diseño que se requieran. Por ejemplo, si se desea que el filtro cumpla con cierta función de transferencia $H(z)$, debemos usar estos $N/2-1$ términos de

manera que nuestro futuro filtro pueda cumplir estas condiciones de la función de transferencia hasta donde le sea posible. Una vez escogidos estos términos usamos las ecuaciones de (6.24) para calcular los términos restantes.

6.1.4 Algoritmo y solución del sistema no lineal

El sistema (6.24) nos entregará como resultado un filtro wavelet, lo cual es muy bueno. Sin embargo, debemos notar que el sistema *no es lineal*. De hecho, (6.22) es conocida también como *condición cuadrática*. Un camino alternativo consiste en usar el Método de Iteración de Newton [DEM1980] el cual utiliza una aproximación de primer orden de una función f con lo cual el sistema (6.24) se reduce a uno equivalente de un sistema lineal de ecuaciones.

Usando este criterio, definiremos las siguientes funciones multivariantes:

$$f_k(h_1, \dots, h_N) = \sum_n h(n)h(n-2 \cdot k), \quad k = 1, 2, \dots, N/2-1, \quad (6.25)$$

$$g_1(h_1, \dots, h_N) = \sum_n h(2n) - \frac{1}{\sqrt{2}}, \quad (6.26)$$

$$g_2(h_1, \dots, h_N) = \sum_n h(2n+1) - \frac{1}{\sqrt{2}}. \quad (6.27)$$

El Método de Iteración de Newton requiere que las funciones en (6.25), (6.26) y (6.27) sean idénticamente cero. De esta manera (6.24) se presenta como un sistema homogéneo. Es importante destacar que estas funciones son todas diferenciables en \mathfrak{R}^N y que sus derivadas son lineales. Esto facilitará un rápido cálculo de los coeficientes del sistema.

6.1.5 Implementación en MATLAB

Se utilizó MATLAB 6.0 para resolver el sistema (6.24) usando el Método de Iteración de Newton y para un filtro de largo $N=4$. A continuación se presentan algunos filtros y sus respuestas en frecuencia que fueron obtenidos a través del programa implementado.

Daubechies 4		Experimental1		Experimental4		Experimental5	
h(n)	FFT	h(n)	FFT	h(n)	FFT	h(n)	FFT
0.4830	1.4142	0.3	1.4142	-0.0815	1.4142	0.8294	1.4142
0.8365	1	0.8507	1	0.1071	1	0.2	1
0.2241	0	0.4071	0	0.7886	0	-0.12	0
-0.1294	1	-0.1436	1	0.6	1	0.5071	1

Tabla 1: Filtros experimentales de orden 4 implementados en MATLAB.

Se escogieron tres ejemplos de filtros de un total de cinco, todos obtenidos a través del programa Experiment_4.m en MATLAB, al cual se le ingresó en cada caso un vector aleatorio $h_a(n)$, para posteriormente obtener en la salida el $h_w(n)$, el cual satisface (6.24). La siguiente tabla lo muestra a continuación:

	Experimento1	Experimento4	Experimento5
Norma	1.00004186	1.00000262	0.99945477
Suma términos pares	0.7071	0.7071	0.7071
Suma términos impares	0.7071	0.7071	0.7094

Tabla 2: Evaluación de los coeficientes $h(n)$ de filtros experimentales.

Como es de notar en la tabla 2, la norma del filtro en todos los casos es unitaria. Además, la suma de todos los términos pares e impares son iguales entre sí e igual a la mitad de la raíz de 2.

Es interesante y sorprendente notar que las respuestas en frecuencia de los tres casos es exactamente la misma, más aún si tomamos en cuenta que los coeficientes de $h(n)$ se ingresaron a la aplicación en forma aleatoria. Esto último es propio del Método de Iteración de Newton, pues aunque se ingrese un valor inicial aleatorio, el algoritmo tiende a buscar la solución más cercana a este valor inicial, la cual es, en nuestro caso, solución al sistema de diseño de filtros (6.24).

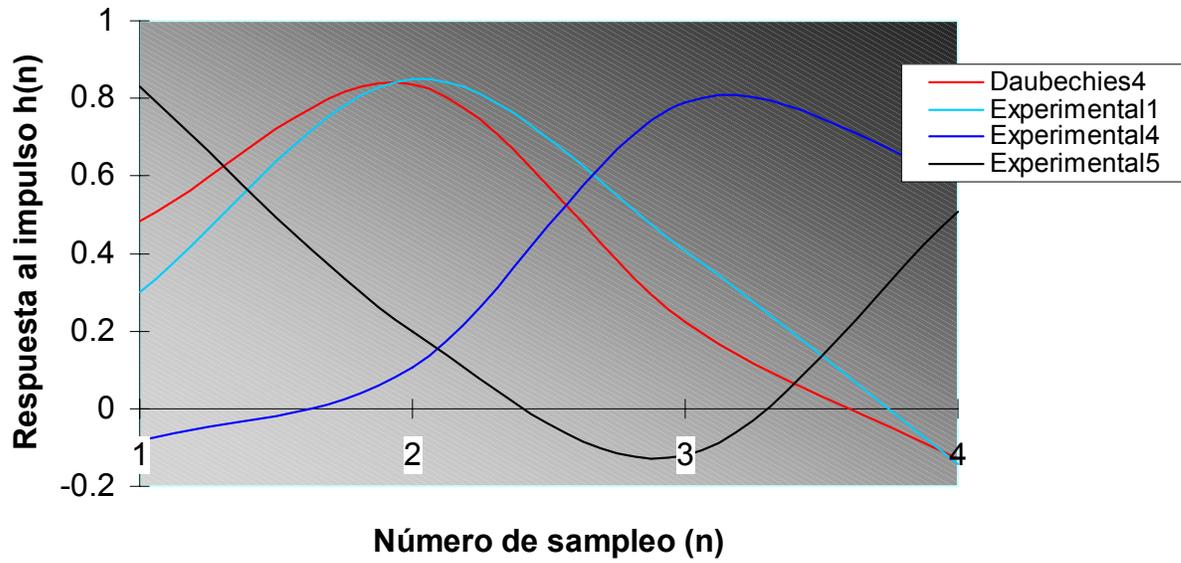


Figura 60: Respuestas al impulso de filtros experimentales de orden 4.

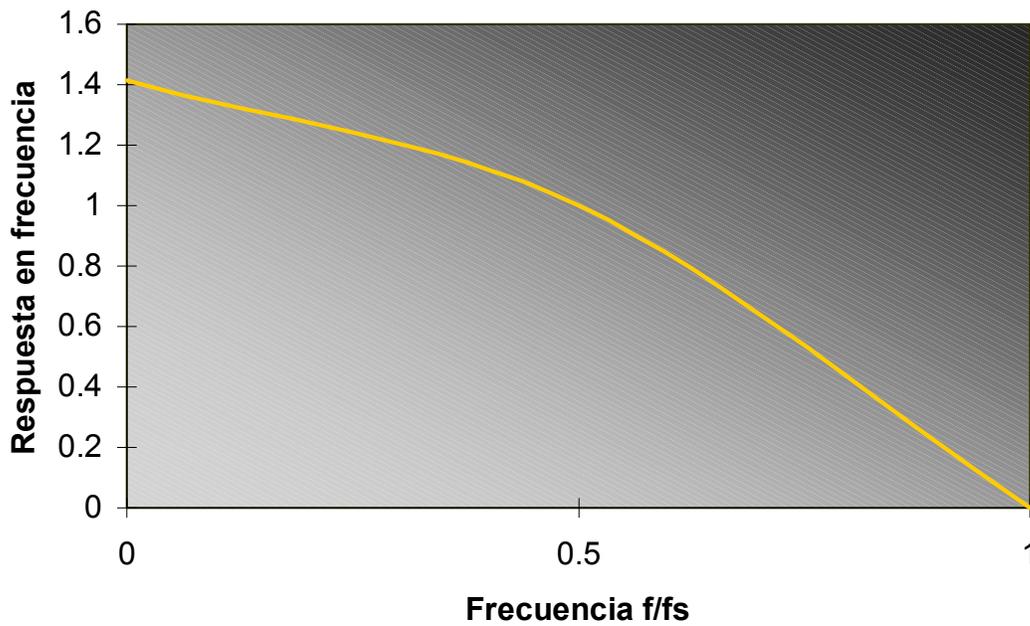


Figura 61: Respuesta en frecuencia de filtros experimentales de orden 4. Observar que en todos los casos la respuesta fue idéntica.

Posteriormente se implementó el programa *Experimental_8.m*, para resolver el mismo sistema, pero ahora con $N=8$. A continuación se presentan los resultados de algunos casos:

Daubechies8		Experimental4		Experimental6		Experimental7	
$h(n)$	FFT	$h(n)$	FFT	$H(n)$	FFT	$h(n)$	FFT
0.230378	1.4142	0.2291	1.4142	-0.0008	1.4142	0.2	1.4142
0.714847	1.4063	0.4176	1.2799	-0.0607	1.3514	0.7643	1.378
0.630881	1	0.7195	1	0.1	1	0.6	1
-0.0279838	0.149	0.3178	0.6016	0.782	0.4169	-0.0815	0.3178
-0.187035	0	-0.3648	0	0.612	0	-0.0929	0
0.0308414	0.149	0.0393	0.6016	-0.0142	0.4169	0.0243	0.3178
0.032883	1	0.1232	1	-0.0041	1	0	1
-0.0105974	1.4063	-0.0676	102799	0.0001	1.3514	0	1.378

Tabla 3: Filtros experimentales de orden 8 implementados en MATLAB.

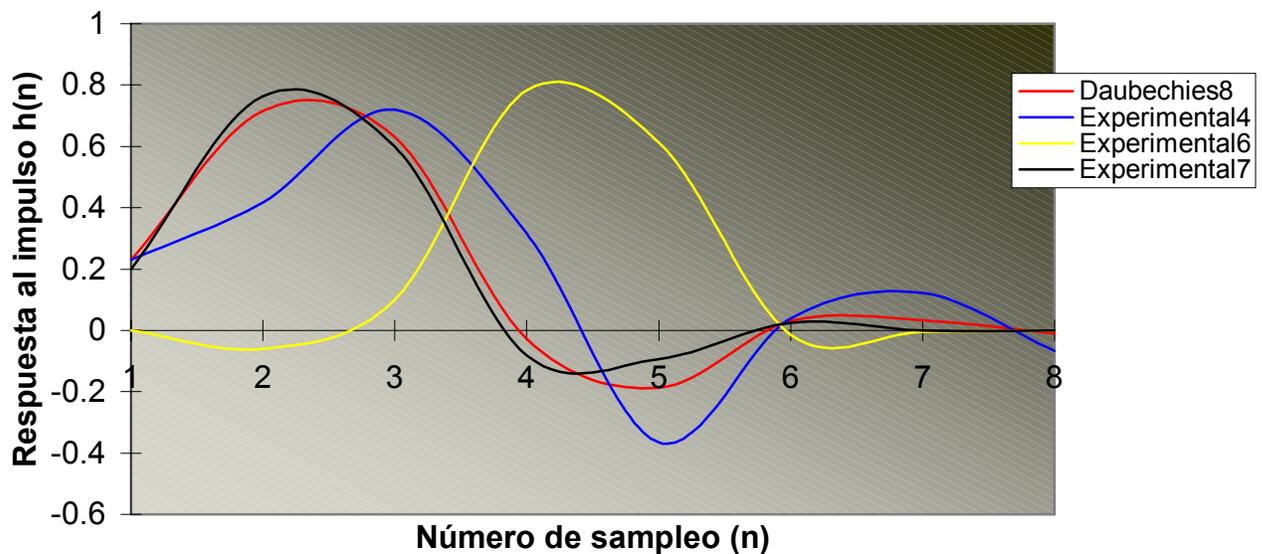


Figura 62: Respuestas al impulso de filtros wavelet de orden 8 implementados en MATLAB.

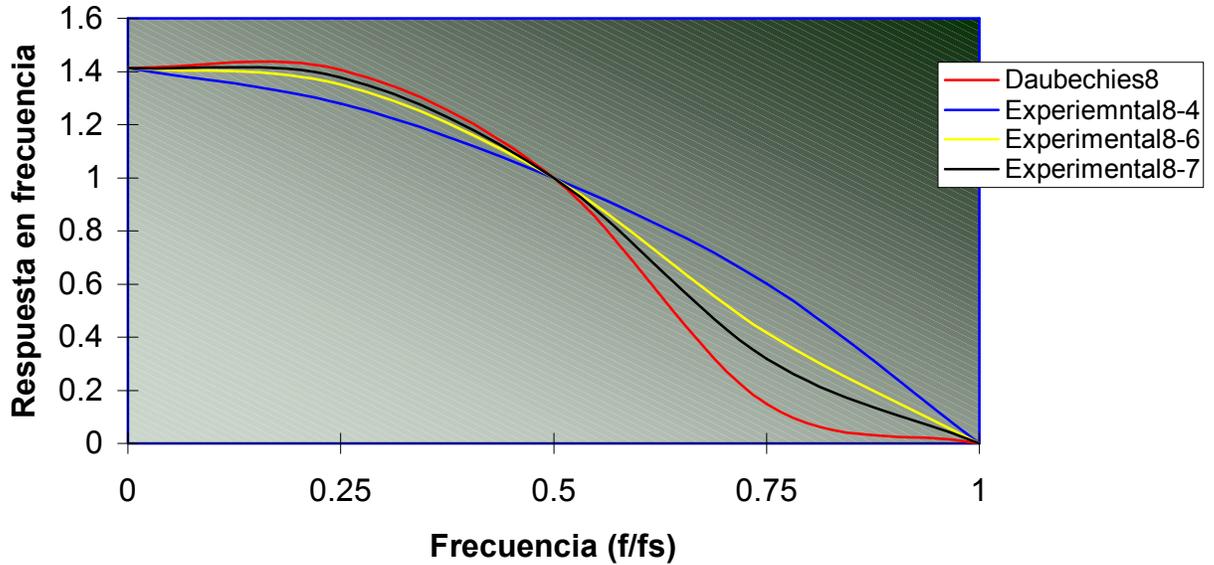


Figura 63: Respostas em frequência de filtros wavelet de ordem 8 implementados em MATLAB.

Finalmente se consideró la misma situación para filtros de orden 12. Se realizaron 4 experimentos y los resultados fueron los siguientes:

Experimento 1		Experimento 2		Experimento 3		Experimento 4	
$h(n)$	FFT	$h(n)$	FFT	$h(n)$	FFT	$h(n)$	FFT
0.1792	1.6638	0.1364	1.463	0.0098	1.4165	0.01	1.4097
0.0902	0.7063	0.4862	1.3476	0.0014	1.4142	-0.04	1.4155
0.2969	0.4528	0.7345	1.4113	-0.1111	1.3853	-0.0598	1.3623
0.0493	1.2293	0.3	0.9513	-0.048	1.0006	0.38	0.9955
0.5009	1.4818	-0.2015	0.3677	0.49	0.2753	0.8183	0.389
0.3027	0.9574	-0.185	0.218	0.7877	0.0065	0.4128	0.0544
-0.399	0.2496	0.1605	0.0488	0.34	0.0023	-0.0845	0.0045
0.5345	0.9574	0.015	0.218	-0.0782	0.0065	-0.05	0.0544
0.3728	1.4818	-0.0318	0.3677	-0.02	0.2753	0.0202	0.389
-0.2578	1.2293	0.079	0.9513	0.0488	1.0006	0.0047	0.9955
0.0059	0.4528	-0.0421	1.4113	0.0007	1.3853	-0.0016	1.3623
-0.0118	0.7063	0.0118	1.3476	-0.0047	1.4142	-0.0004	1.4155

Tabla 4: Filtros experimentales de orden 12 implementados en MATLAB.

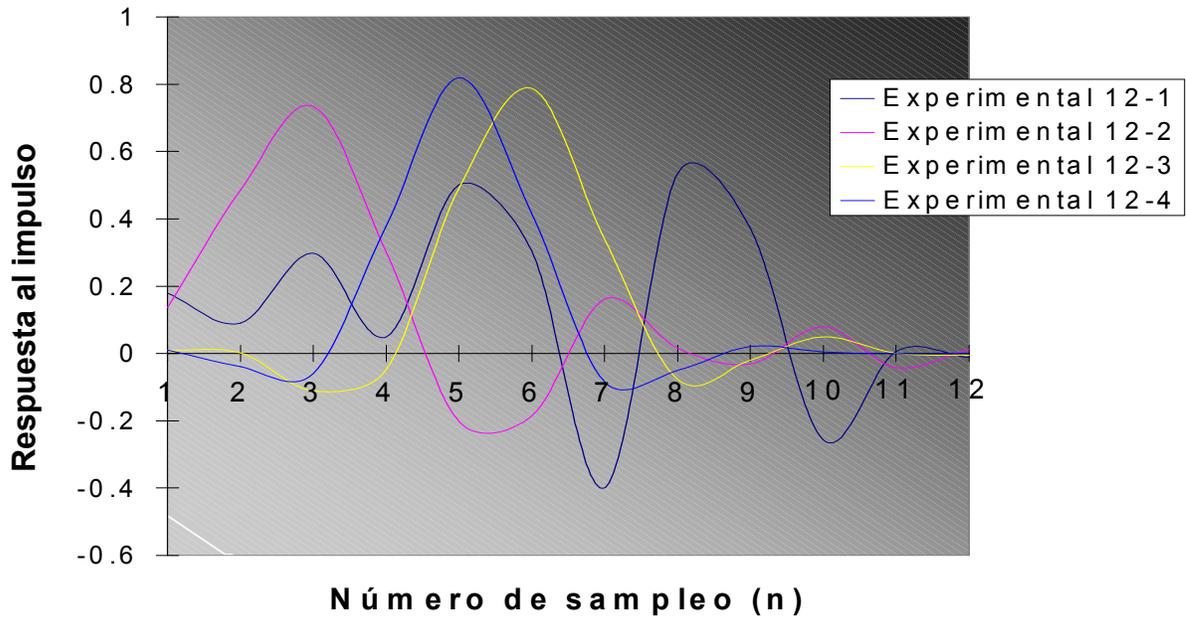


Figura 64: Respuestas al impulso de filtros wavelet de orden 12 implementados en MATLAB.

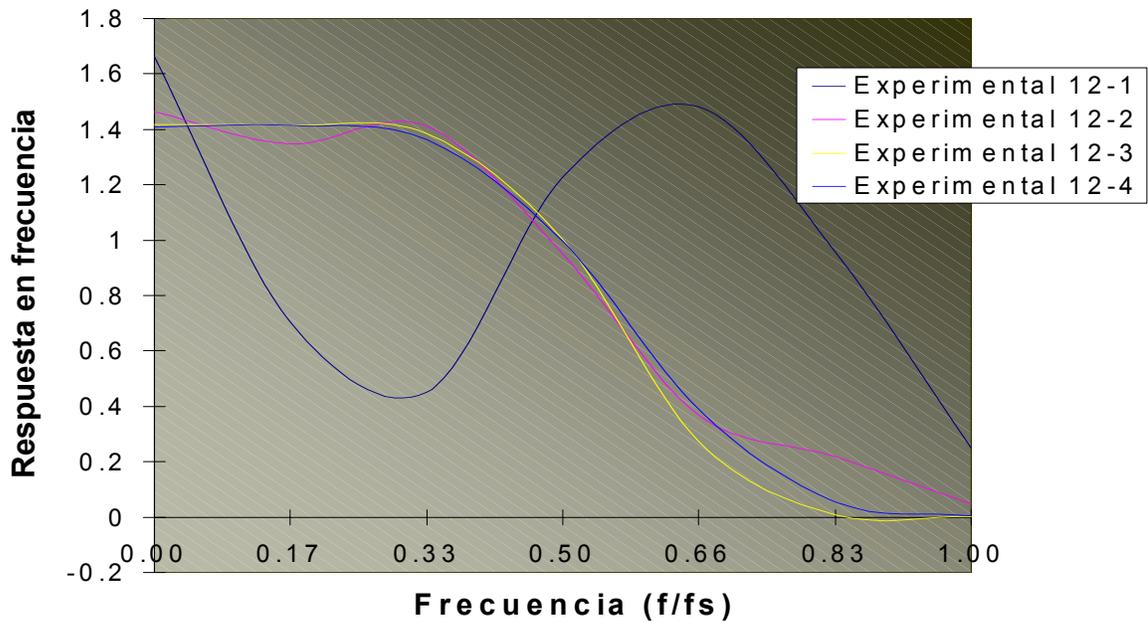


Figura 65: Respuestas en frecuencia de filtros wavelet de orden 12 implementados en MATLAB.

6.2 Aplicación de la DWT a una señal con ruido

Se construyó una señal con ruido de alta frecuencia, dada por la siguiente función:

$$y=f(t)=\text{sen}(t)+\text{sen}(10t)+\text{sen}(50t)+\text{sen}(200t)+\text{sen}(500t),$$

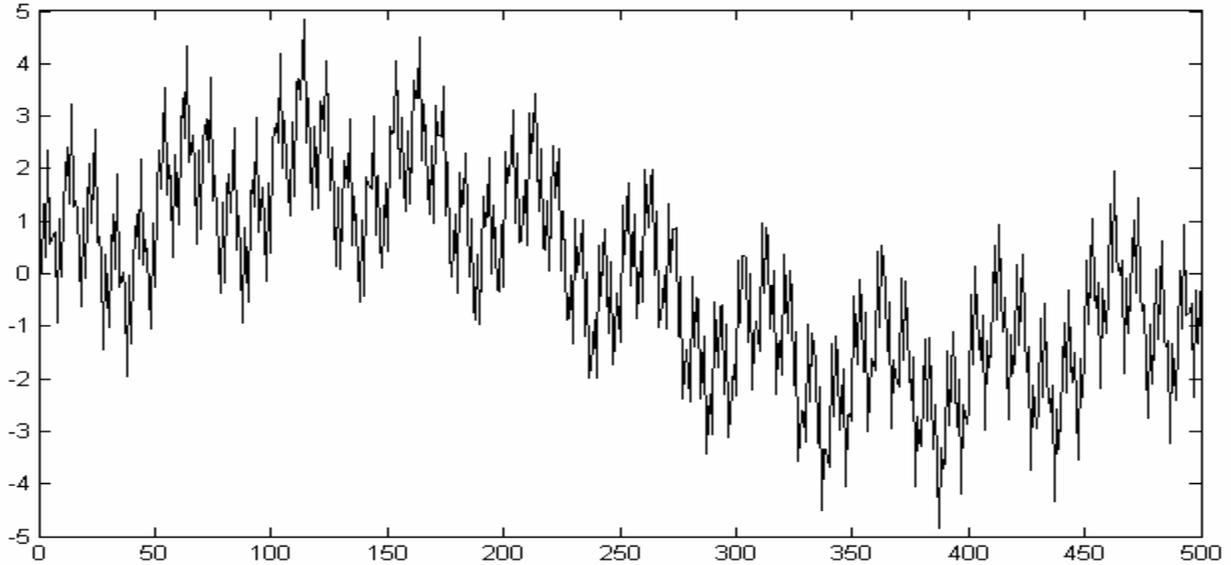


Figura 66: Simulación de una señal con ruido.

Esta señal, si bien ha sido creada artificialmente, podría ser perfectamente la representación de una señal real. De hecho, existen señales en la vida cotidiana que presentan problemas de resolución muy similares al presente [FAF2000]. Si este es el caso de una señal de audio, podemos notar claramente que contiene ruido en las frecuencias altas y medias, y existe una onda moduladora de baja frecuencia. Es deseable poder eliminar parte del contenido espectral de esta señal si queremos ver “como es en realidad” la información que esta contenida.

Existen muchos métodos de filtrado, algunos de los cuales ya han sido mencionados. Veremos ahora como trabaja la transformada discreta wavelet para lograr descomponer esta señal en ondas más simples.

Utilizando los algoritmos de la DWT implementados en MATLAB y utilizando el sistema Wavelet Haar, se ha obtenido la descomposición para una etapa, como se muestra en la figura 67.

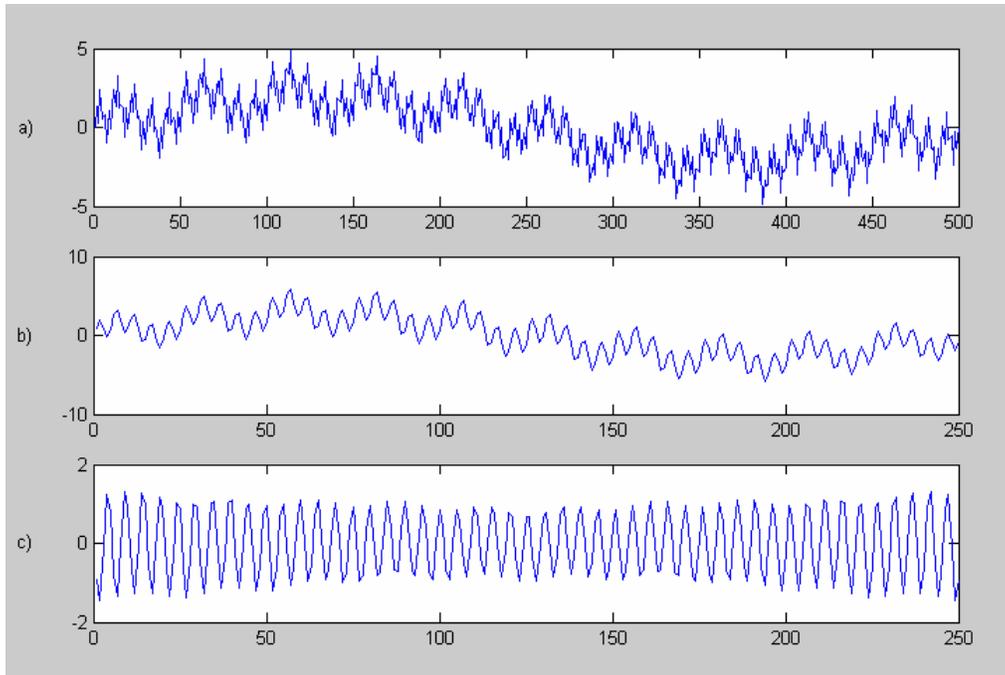


Figura 67: Descomposición de la señal con ruido aplicando la DWT una vez.

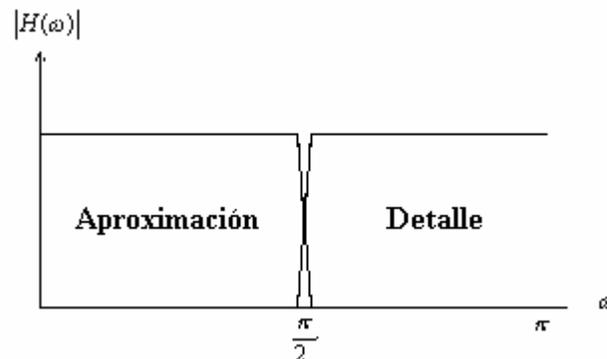


Figura 68: Esquema de la DWT en el dominio de la frecuencia de la descomposición de la señal en una etapa.

La curva a) de la figura 67 corresponde a la señal original. Las siguientes corresponden a las señales obtenidas luego de aplicar la DWT. Claramente la señal en b), que corresponde a la aproximación de la señal original, ha sido obtenida a través de un filtro pasa-bajos, pues la onda moduladora de baja frecuencia se mantiene; mientras que la señal en c), correspondiente al detalle de la señal original, contiene alta frecuencia debido a la aplicación de un filtro pasa-altos. Es importante notar que la descomposición no ha sido del todo completa, pues ambas señales contienen aún componentes en frecuencias altas y bajas, respectivamente. Sin embargo, la DWT se puede aplicar tantas veces como se quiera hasta obtener el grado de descomposición deseado.

Los resultados de la DWT en tres etapas, para la misma señal se muestran en la figura 69.

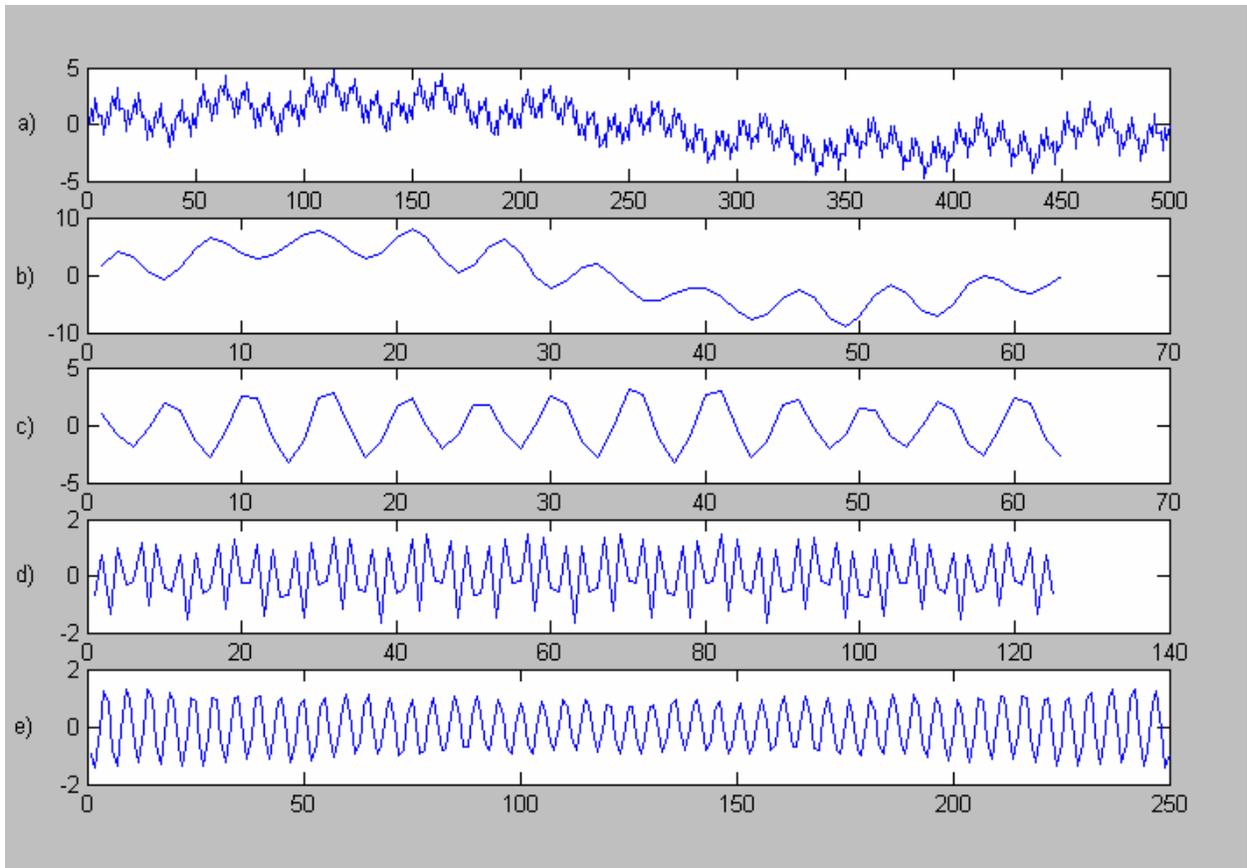


Figura 69: Descomposición de la señal con ruido al aplicar la DWT tres veces.

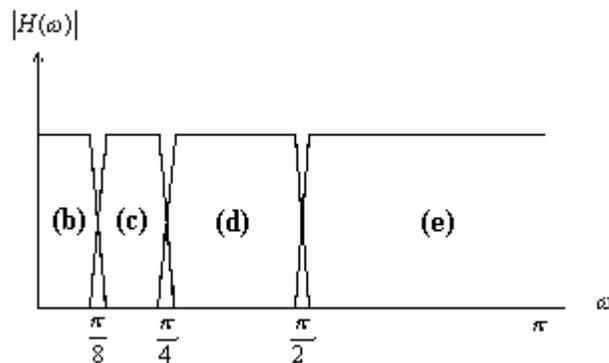


Figura 70: Esquema de la DWT en el dominio de la frecuencia. Aquí se descompuso la señal mediante tres etapas.

Podemos observar que hemos logrado descomponer la señal en una nueva aproximación, la cual se muestra en b. Las señales en c, d y e corresponden al detalle de la señal original.

Posteriormente se aplicó la misma técnica de descomposición, pero ahora usando el sistema wavelet Daubechies 4, como se muestra en la figura 71.

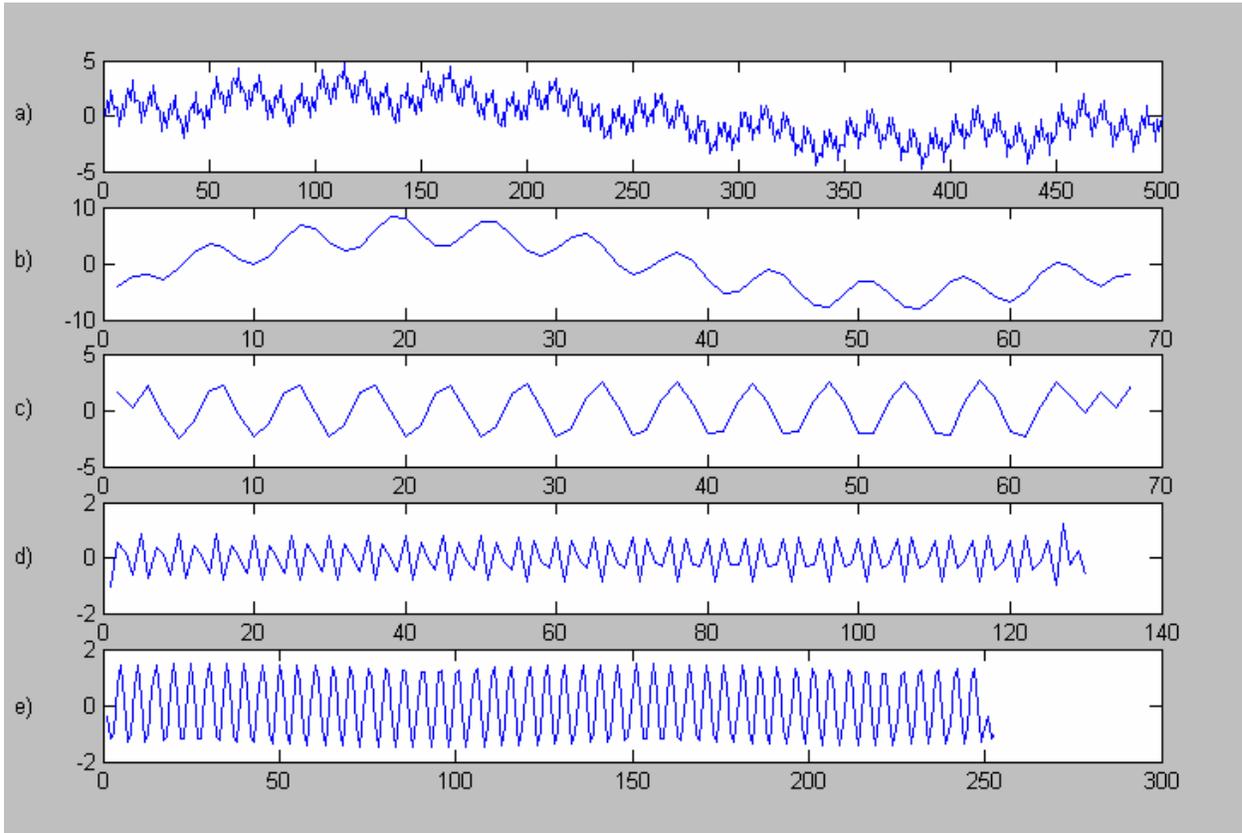


Figura 71: Esquema de la DWT en el dominio de la frecuencia. Descomposición en tres etapas usando el sistema Daubechies 4.

6.2.1. Reconstrucción de la señal filtrada

Una vez que se ha realizado la descomposición de la señal a través de la DWT, es posible seleccionar la parte de mayor interés de la señal, para reconstruir una señal “filtrada”. En este caso debemos volver hacia atrás; por lo tanto la herramienta que necesitaremos es la Transformada Inversa Wavelet (IDWT).

En este caso, fue de interés filtrar todo el ruido de las frecuencias altas y medias, dejando solamente la señal representada en b). Es importante notar que la señal original tiene 500 puntos, mientras que la señal b) solo posee 70. Esto se debe a la decimación por dos luego de aplicar la transformada sucesivas veces. Uno de los objetivos de la reconstrucción es recuperar la longitud de la señal original.

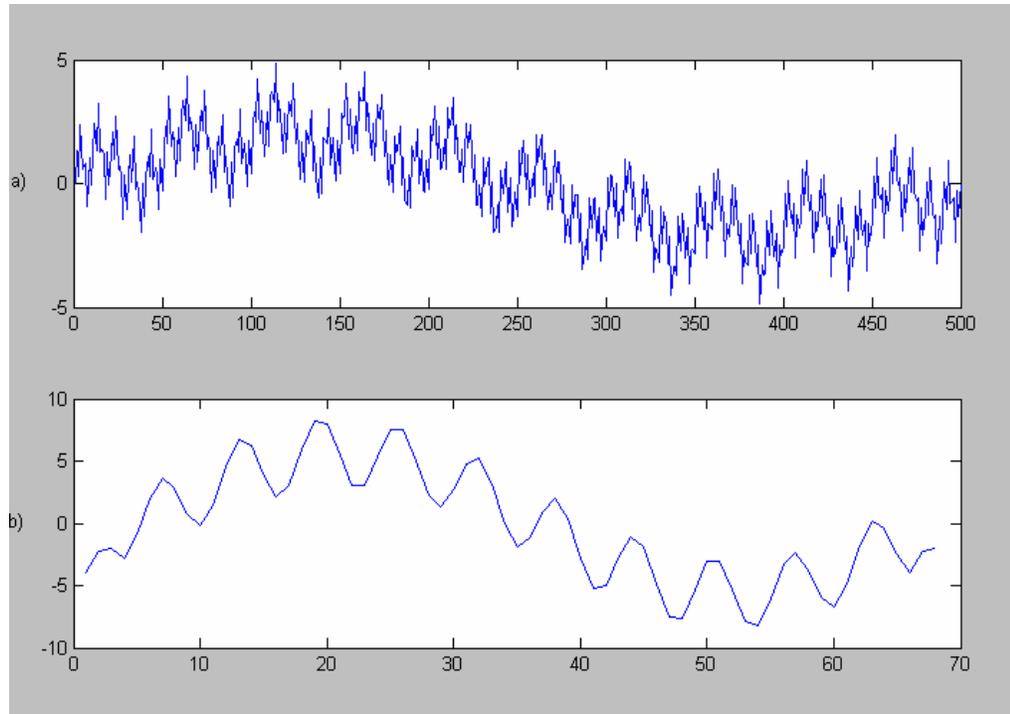


Figura 72: Filtrado de la señal de interés sin reconstruir.

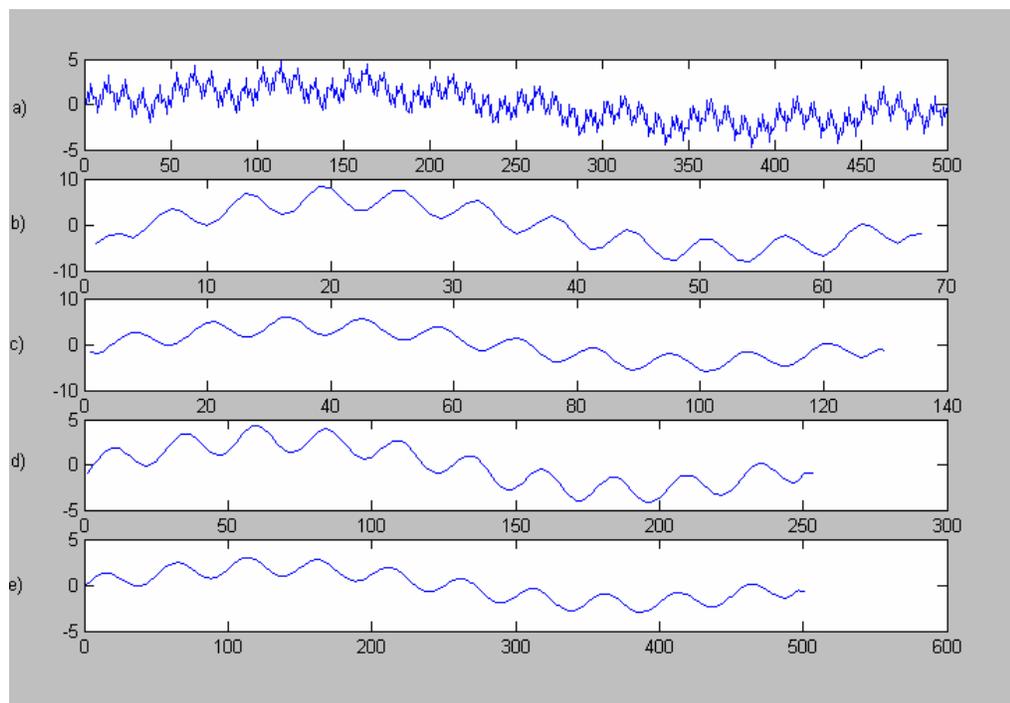


Figura 73: Señal filtrada y reconstruida aplicando la IDWT en tres etapas.

La reconstrucción puede considerarse buena por dos aspectos: se ha recuperado la longitud de la señal original, y además el rango de amplitud de la señal filtrada también coincide con el de la señal original.

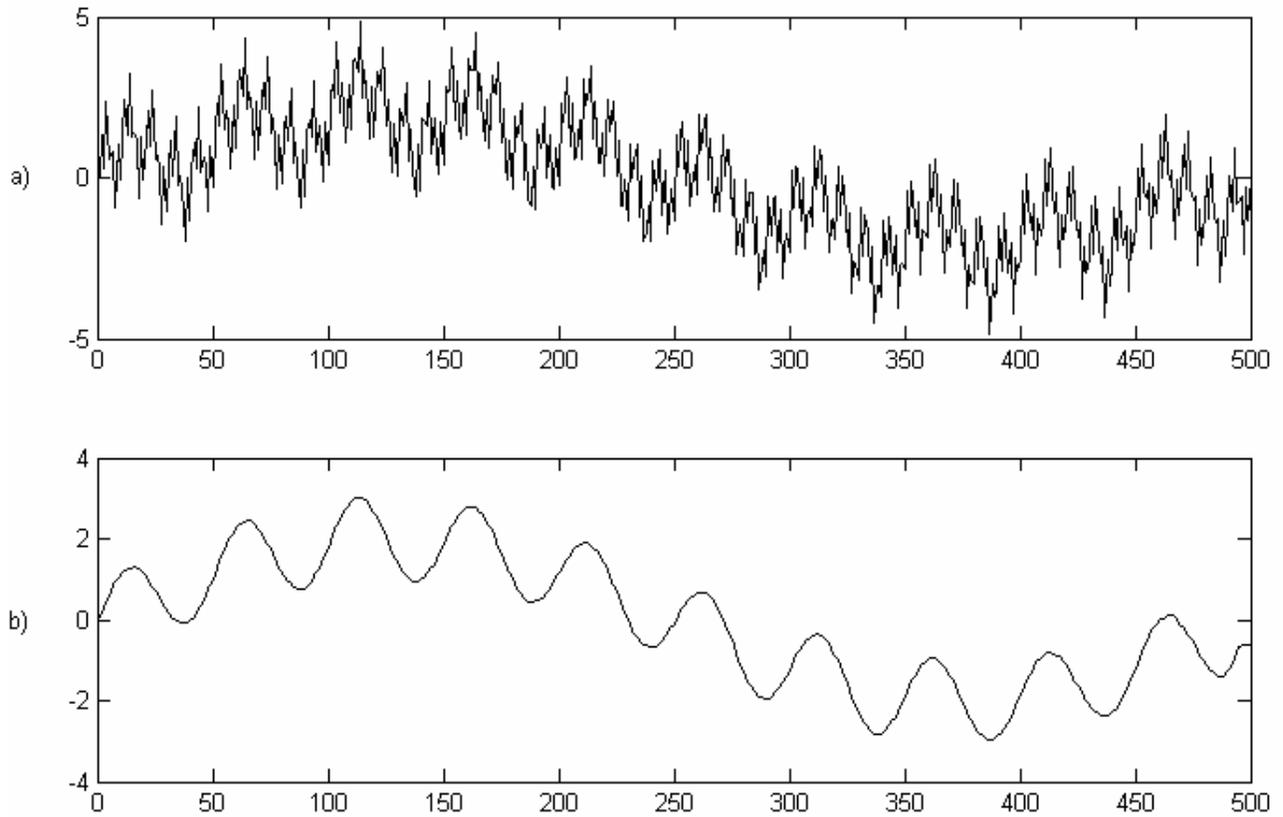


Figura 74: Señal con ruido y señal filtrada-reconstruida final.

DISCUSIÓN Y PROYECCIONES DE TRABAJO

La motivación del desarrollo experimental se dio con la idea de diseñar filtros digitales de tipo FIR, con la particularidad que estos correspondan a un sistema wavelet. De hecho, cualquier sistema wavelet conocido, tal como Daubechies, Coifman, etc., lleva asociado consigo un filtro digital FIR, que es intrínsecamente propio del sistema wavelet escogido, por lo que no es posible rediseñar el filtro sin intervenir el sistema wavelet. Cabe mencionar que los sistemas wavelet son infinitos, y bien podría diseñarse un sistema para una aplicación particular; de hecho en muchos problemas reales se procede de esta manera, tales como en las señales de electrocardiogramas, imágenes, etc. Por otro lado, existen varios caminos para diseñar un filtro digital; lo mismo ocurre para el diseño de sistemas wavelet.

En el presente trabajo se ha intentado buscar un camino que permita diseñar ambas cosas a la vez. Resolver el sistema (6.24) puede responder a esta inquietud. Sin embargo, es importante dejar en claro qué es exactamente lo que se está buscando de la secuencia $h(n)$; qué características se desean que tenga el filtro, pendiente en la banda de transición, ancho de banda, etc. Es decir, los parámetros para diseñar el filtro. Lo ideal sería poder determinar la función de transferencia del filtro con tales especificaciones. Debido a que (6.24) contiene un número de incógnitas que supera al número de ecuaciones, este puede ser resuelto de infinitas maneras, y por lo tanto existe la oportunidad de acotar el conjunto solución para aquellos filtros que cumplan con las características fijadas previamente y que son deseadas. En términos matemáticos, fijar estas características implica agregar un número extra de ecuaciones en (6.24), lo cual trae algunas consecuencias:

En primer lugar, si se busca la solución exacta del sistema (lo cual involucra expresar la solución en términos elementales) la dificultad de ello dependerá de la complejidad de las ecuaciones que se agreguen al sistema (6.24). Debemos recordar que el sistema no es lineal, y por lo tanto, su solución ya es complicada. Esto es un tema de importancia si lo que se busca es un modelo matemático de filtros wavelet.

Por otra parte, colocar restricciones es muy bueno si se necesita una solución aproximada del problema ya que el conjunto solución se hace más pequeño, lo cual puede facilitar la convergencia hacia una solución si es que se está utilizando algún método de aproximación numérico. En esta línea se ha investigado cómo resolver el problema.

En el presente trabajo se ha utilizado uno de los métodos de aproximación más conocidos y sencillos: el método de iteración de Newton. En la solución del sistema sólo se ha dado un vector inicial $h_0(n)$ como única restricción e iterado una cierta cantidad de veces hasta lograr un margen de error aceptable, que en este caso fue del orden de 10^{-9} . El sistema (6.24) fue implementado usando el método de iteración de Newton en el software MATLAB 6.0, y se trabajó solamente con filtros de orden 4, 8 y 12 respectivamente.

En cuanto a la convergencia hacia una solución, en los resultados obtenidos se ha observado una rápida convergencia para los filtros de orden 4, de tal manera que al colocar una secuencia inicial, incluso aleatoria, la convergencia era casi inmediata,

requiriendo incluso hasta 3 iteraciones solamente. Posteriormente, se implementó el mismo algoritmo para filtros de orden 8. Aquí, la convergencia también fue rápida, aunque en algunos casos fue necesario escoger con más cuidado el vector inicial $h_0(n)$.

La implementación de filtros de orden 12 dejó en claro que la rapidez de convergencia del algoritmo disminuye notablemente en la medida que aumenta la dimensión del filtro. De hecho fue necesario modificar el programa en MATLAB para que la computadora pudiese efectuar los cálculos pues de lo contrario no se habría podido ejecutar el algoritmo. A pesar de estas dificultades, se logró obtener algunos filtros con una pendiente aceptable en la banda de transición y sin ripple.

La convergencia es un factor importante del problema, pues es claro que para filtros de orden mucho mayor, por ejemplo, de 20 términos, la rapidez de cómputo se volvería extremadamente lenta, por lo que este método de diseño de filtros sería obsoleto. Por lo mismo, es recomendable buscar soluciones para el sistema (6.24) utilizando un sistema de aproximación numérico más adecuado. Esto puede ayudar en gran manera a mejorar el algoritmo de diseño.

Cabe señalar que la convergencia por si sola no sirve de mucho si no se deja en claro qué es exactamente lo que se está buscando, pues se puede designar un valor inicial al sistema y encontrar un filtro que a lo mejor no cumplirá ninguna de las expectativas esperadas. Por esta razón deben fijarse las condiciones del problema; como se mencionó antes, utilizando una cierta cantidad de ecuaciones que indiquen las características deseadas del filtro. Esto de por sí facilitará una rápida convergencia hacia una solución, pues dichas condiciones nos darán un "itinerario" en donde se encuentra el filtro que estamos buscando, reduciendo, de esta manera, el número de soluciones a sólo aquellas que son de interés.

Surge la pregunta natural, ¿Qué es lo que sigue una vez que se ha logrado encontrar el filtro digital con las características deseadas?, pues existen varios y eficientes algoritmos para el diseño de filtros digitales que no involucran las dificultades que plantea la presente metodología. A diferencia de los métodos de diseño expuestos en esta tesis, que son los más conocidos y usados, la aplicación de la transformada wavelet a una señal implica por un lado, representar dicha señal como una expansión finita de funciones, tal como la expansión de Fourier; y en términos de la frecuencia, significa que se está aplicando simultáneamente un filtro pasa-altos y otro filtro pasa-bajos. Esto es muy bueno por dos razones: Por un lado, sólo necesitamos aplicar una vez la DWT para realizar dos filtrados en forma simultánea; y por otro, realizar esta operación en términos de cálculos es muy rápido y además es fácil de implementar como algoritmo computacional, más aún si el sistema wavelet se conoce de antemano. Esto hace de los filtros wavelet una potente herramienta en términos de la rapidez con que puede filtrarse la información contenida en una señal.

A pesar de estas grandes ventajas, los filtros wavelet tienen un inconveniente en términos del diseño de filtros digitales, y es que su frecuencia de corte no puede ser modificada. Esto puede hacer pensar que estaremos limitados para seleccionar la banda de frecuencias que es de nuestro interés. En parte esto es cierto si aplicamos la DWT una sola vez. Pero al igual que en el caso de otros filtros digitales, podemos usar la DWT en varias etapas con lo que el espectro que- dará dividido en un set de filtros con anchos de banda en progresión geométrica, con lo que es posible seleccionar la(s) banda(s) que sean de interés. Claramente esto no ocurrirá en las frecuencias altas, y para este caso deberá aplicarse otro criterio.

Respecto a la fase de los filtros digitales wavelet, el modelo FIR de estos no garantiza su total linealidad. Esto se debe a que ningún sistema wavelet puede tener soporte compacto y ser simétrico a la vez, excepto el sistema Haar [HAR1998].

Proyecciones futuras

Uno de los objetivos de este trabajo ha sido mostrar la conexión existente entre la teoría de wavelets y la de filtros digitales. Se ha mostrado que es posible seccionar el espectro de una señal en un set de filtros con ancho de banda variable, lo cual permite una cierta libertad de la elección de las frecuencias que se desean seleccionar pero con preferencia en las frecuencias bajas. Una manera de lograr esta misma selectividad en las frecuencias altas es a través de los *sistemas wavelet biortogonales*, los cuales permiten seccionar el espectro en M bandas después de una aplicación de la DWT. Un desarrollo de estos sistemas y sus correspondientes filtros ofrece una selectividad mucho más fina de las bandas de frecuencia en el espectro.

El algoritmo de diseño de filtros wavelet ha sido resuelto por métodos numéricos. No se descarta el hecho de que sea posible resolverlo en términos elementales y obtener así la solución exacta en función de las variables libres. También se deja abierta la elección de otro sistema numérico para obtener soluciones aproximadas que favorezca una rápida convergencia del sistema hacia la(s) solución(es); así como también las ecuaciones más adecuadas que permitan restringir el problema.

Finalmente, se espera que con estas herramientas sea posible diseñar un software que permita diseñar un filtro, para luego transformarlo en un filtro wavelet, o en su mejor aproximación, en que el sistema wavelet asociado podrá ser utilizado para descomponer y reconstruir señales de la manera que más convenga.

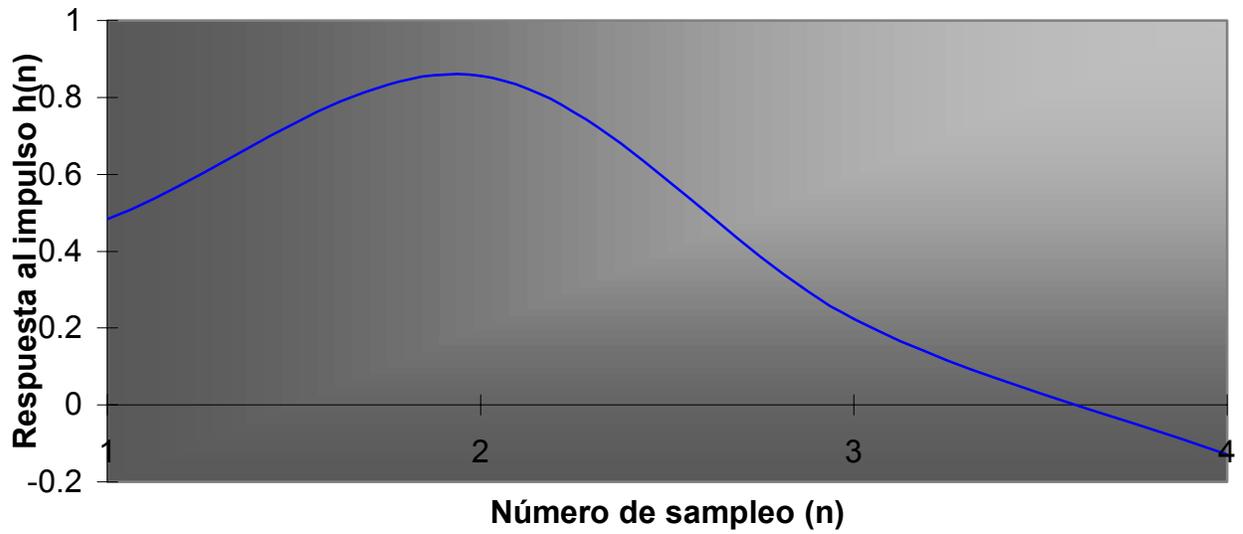
CONCLUSIONES

- La teoría matemática de las Wavelets deja en claro que las secuencias de recursión h y h_l , provenientes de un sistema wavelet, se comportan como filtros digitales FIR, en el que h es un filtro pasa-bajo, y h_l un filtro pasa-alto. Esto se debe, fundamentalmente, a la aplicación de la transformada discreta wavelet (DWT) que responde a una convolución discreta de la señal de entrada con las secuencias h y h_l .
- Los filtros wavelet poseen una frecuencia de corte invariable. De hecho son filtros de “media banda”. Sin embargo, es posible dividir el espectro de la señal en un set de filtros, lo cual permitirá hacer una selección de la(s) banda(s) de frecuencia de interés.
- Los filtros wavelet, con excepción del sistema Haar, son de fase no lineal.
- Cualquier secuencia h proveniente de un sistema wavelet corresponde a un filtro digital FIR. Sin embargo, no todo filtro digital FIR pertenece a un sistema wavelet.
- A partir de las condiciones matemáticas de suficiencia en la admisibilidad de un sistema wavelet, se ha encontrado un algoritmo matemático de fácil implementación computacional para encontrar secuencias h (filtros) que a su vez pertenecen a un sistema wavelet. Tal algoritmo puede ser utilizado para diseñar nuevas familias de wavelets con el fin de resolver un problema más específico.
- Al ser no lineal el algoritmo de diseño, se han debido usar métodos de aproximación lineal como el Método de Iteración de Newton, con el fin de encontrar la(s) solución(es). Aunque este método funciona bien para longitudes de filtros pequeñas, la convergencia hacia una solución se hace más lenta para una mayor longitud del filtro. Esto puede ser mejorado utilizando algún método de aproximación lineal más adecuado, y añadiendo las ecuaciones necesarias asociadas con las especificaciones del filtro que se está buscando.
- Existen muchas maneras de diseñar filtros digitales FIR usando wavelets, pues para diferentes sistemas wavelet se puede obtener una misma respuesta en frecuencia. Una manera de reducir la búsqueda consiste en diseñar filtros usando una familia o un número limitado de familias wavelet en diferentes longitudes.
- En el ámbito de la acústica, la transformada discreta Wavelet es una potente herramienta para separar señales en diferentes bandas de frecuencia, pues una sola aplicación de la DWT involucra la aplicación de un filtro pasa-altos y otro filtro pasa-bajos en forma simultánea, usando algoritmos de implementación computacional de fácil programación.

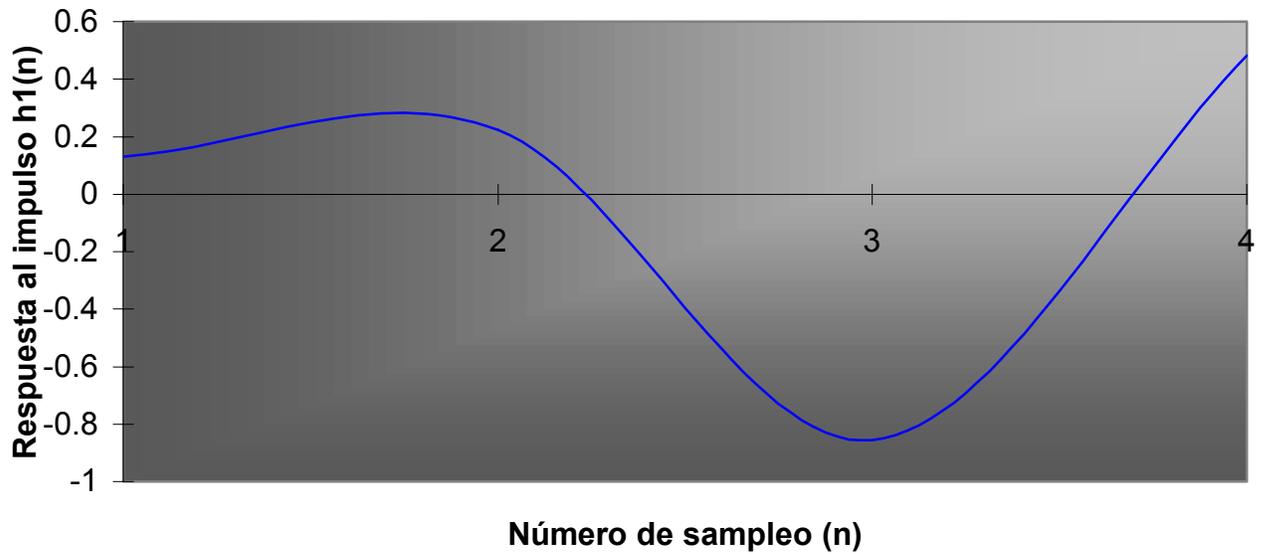
BIBLIOGRAFÍA

- [FAF2000] Faundez, P., Fuentes, A. *Teoría y aplicación de Wavelet en el procesamiento digital de señales acústicas. Tesis de pregrado. Instituto de acústica. Universidad Austral de Chile, 2000.*
- [VER1997] Vergara, M.I. *Implementación de una herramienta computacional para el diseño de filtros digitales de octava y tercio de octava. Tesis de pregrado. Instituto de Acústica. Universidad Austral de Chile, 1997.*
- [APO1972] Apóstol, T.M. *Análisis matemático: - introducción moderna al cálculo superior.* Barcelona : Reverté, 1972.
- [BUR1998] Burrus, S., Ramesh,G. *Introduction to wavelets and wavelet transforms : - a primer.* Upper Saddle River : Prentice-Hall, 1998.
- [DEM1980] Demidovich, B.P. *Métodos numéricos de análisis.* Madrid, Parainfo. 1980.
- [EDW2001] Edwards,C., Penney, D. *Ecuaciones diferenciales.* Universidad de Georgia. Pearson Education, 2001.
- [HAR1998] Hardle,W., Kerkyacharian, G., Picard, D., Tsybakov, A. *Wavelets, Approximation, and Statistical Applications.* New York : Springer, 1998.
- [HAS1973] Haser, N.B., Lasalle, J.P., Sullivan, J.A., *Análisis Matemático, Tomo 2.* México : Trillas, 1973.
- [HSU1973] Hsu, H.P. *Análisis de Fourier.* Bogotá. Fondo de educación iberoamericano.
- [MAL1999].Mallat, S.G. *A wavelet tutorial from S. Mallat's book.* San Diego: Academic Press, 1999.
- [NAK1997] Nakamura, S. *Análisis numérico y visualización gráfica con MATLAB.* México : Prentice Hall, 1997.
- [PRO1996] Proakis, J. *Digital Signal Processing.* New Jersey : Prentice Hall, 1996.
- [RIC1975] Richard, R., Robert, G. *Sistemas y señales.* México : Limusa, 1975.
- [SAV2000] Savant, C.J. *Diseño electrónico, circuitos y sistemas.* México: Pearson Education, 2000.
- [SER2000] Sérroul, R. *The Bézout theorem.* Berlin: Springer-Verlag, 2000.
- [SMI1997] Smith, S. *The Scientist and Engineer's Guide to Digital Signal Processing.* California Technical Publishing, 1997.

APENDICE A. RESPUESTAS AI IMPULSO DE FILTROS WAVELET. DOMINIO DEL TIEMPO. FAMILIA DAUBECHIES

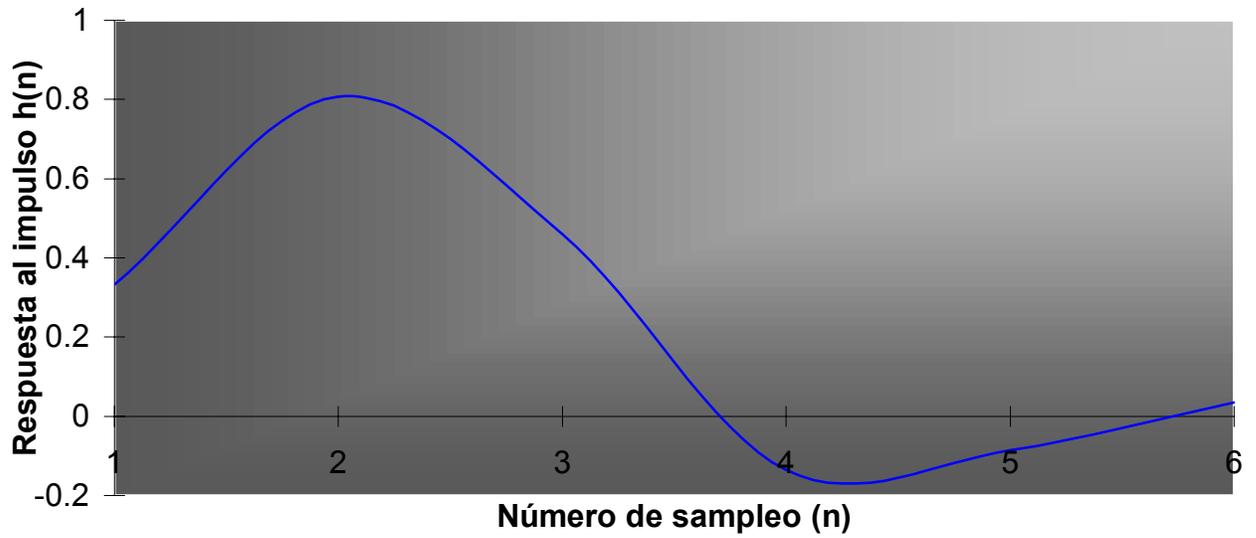
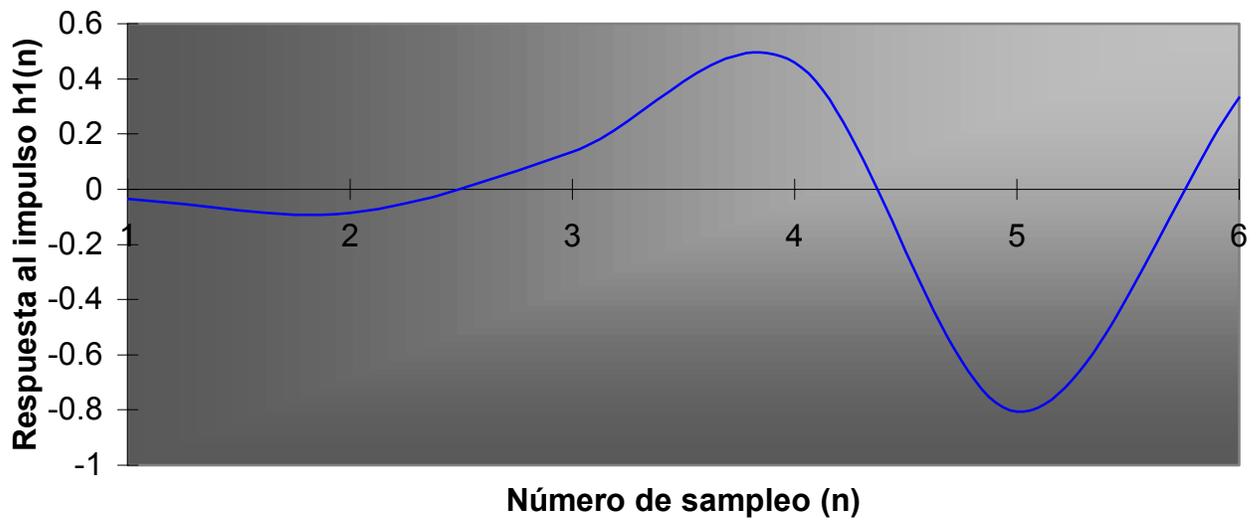


Filtro Escala Daubechies 4.

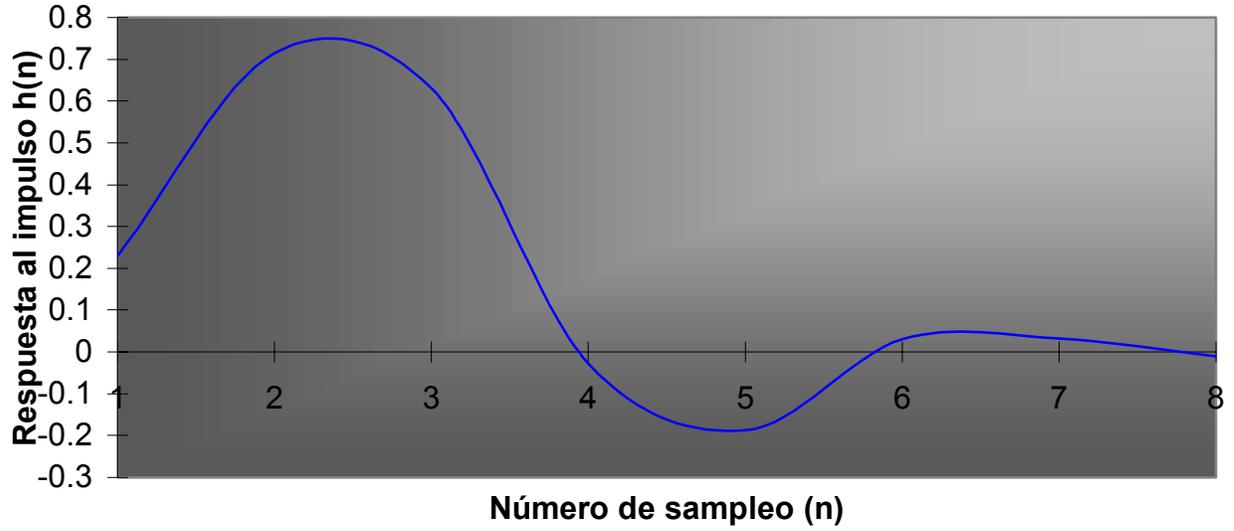
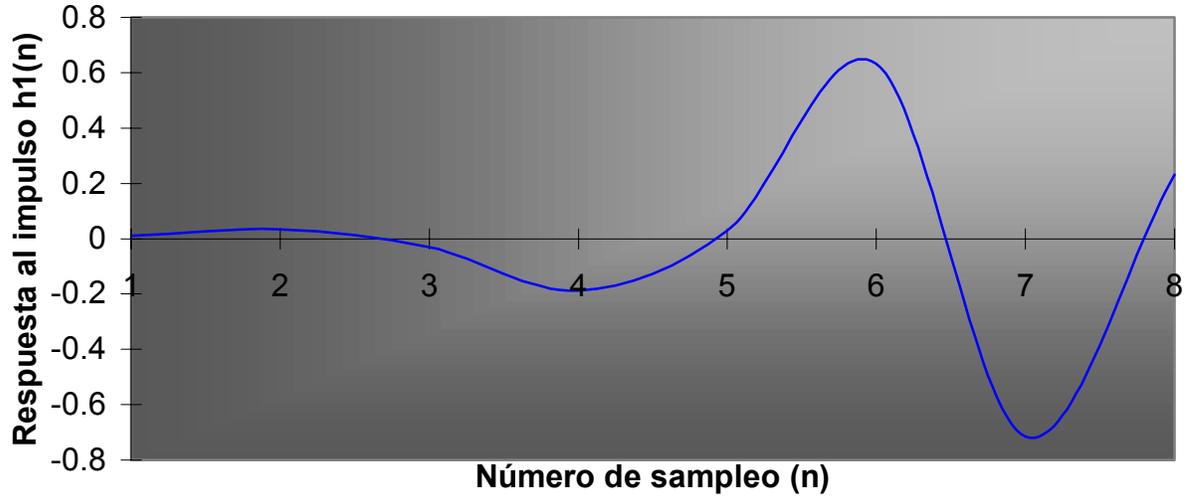


Filtro Wavelet Daubechies 4.

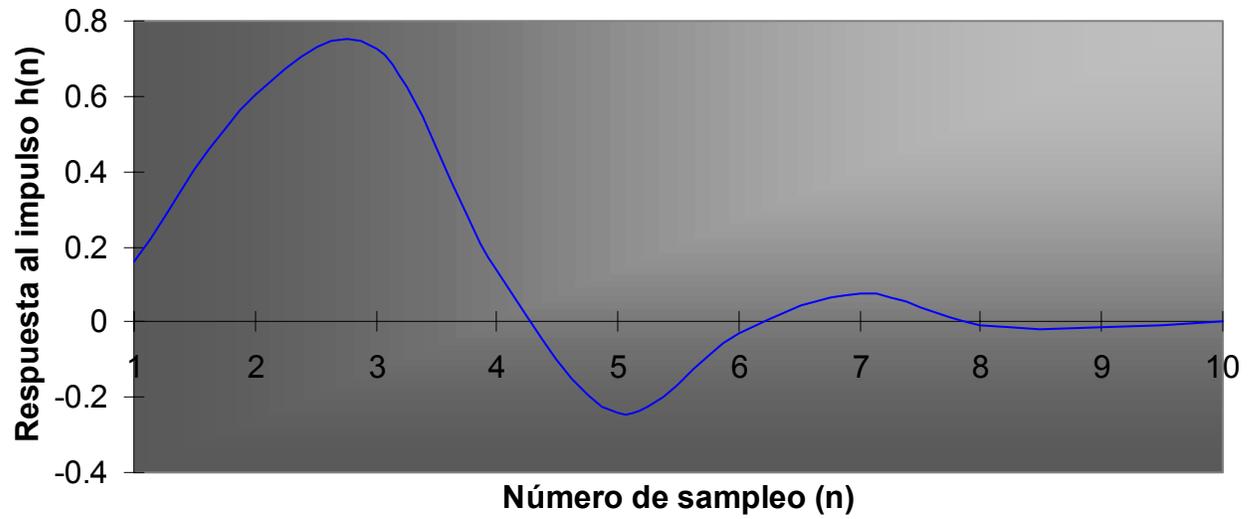
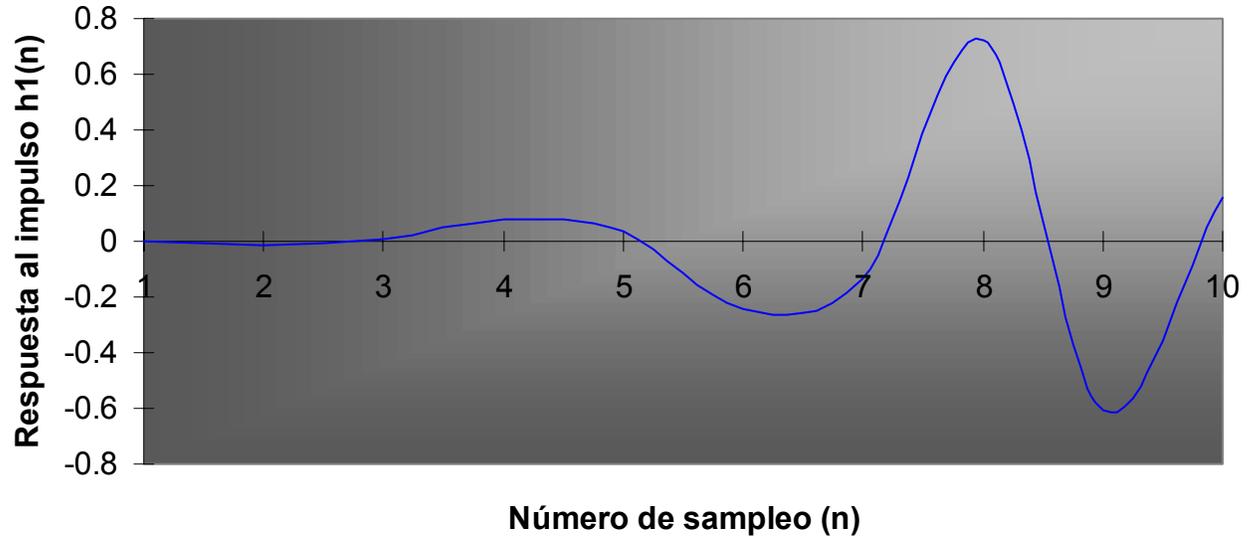
FAMILIA DAUBECHIES

*Filtro Escala Daubechies 6.**Filtro Wavelet Daubechies 6.*

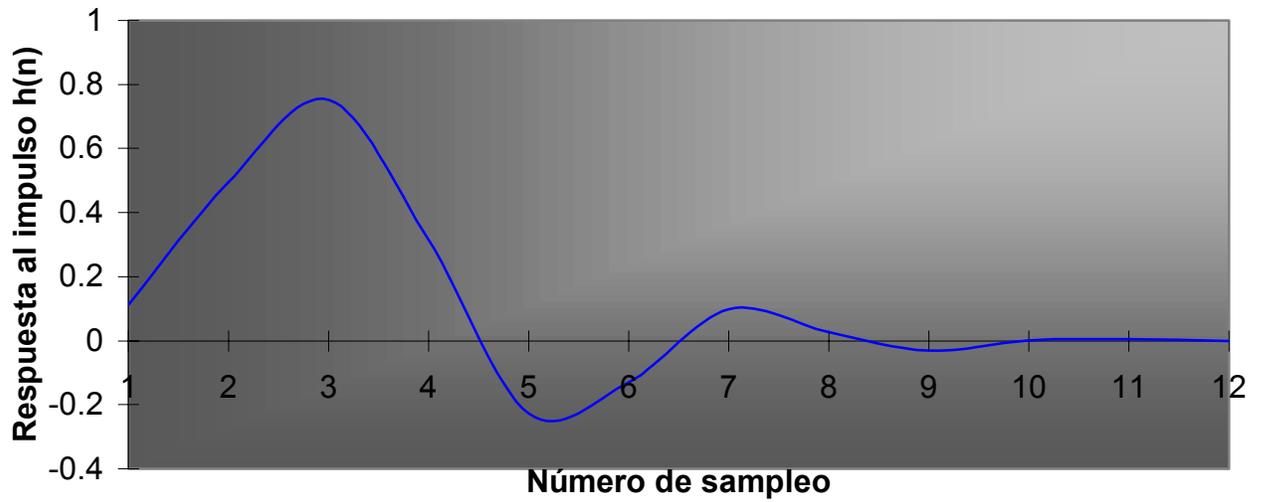
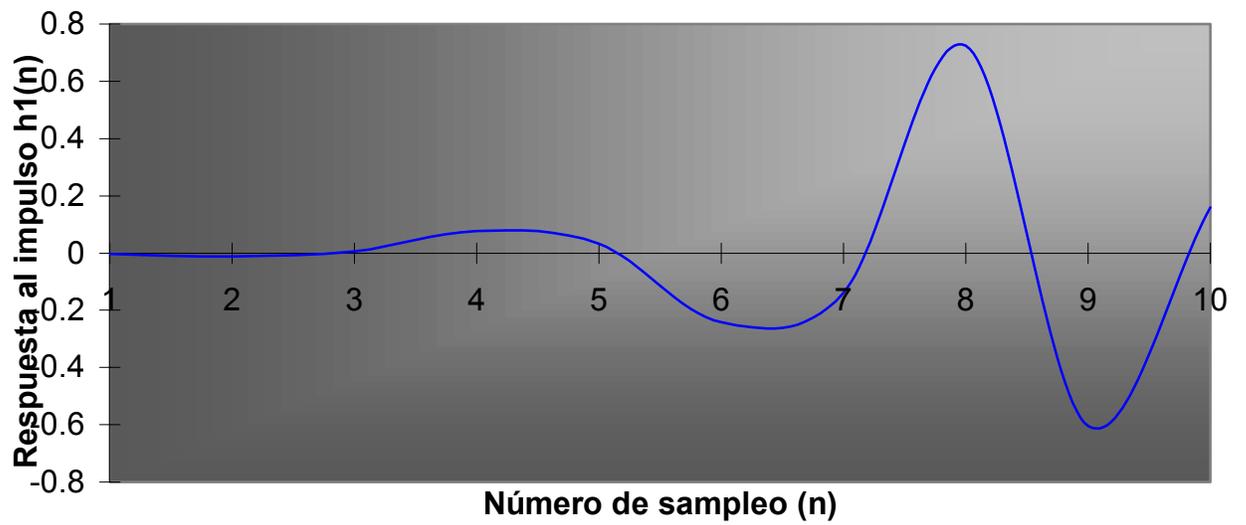
FAMILIA DAUBECHIES

*Filtro Escala Daubechies 8.**Filtro Wavelet Daubechies 8.*

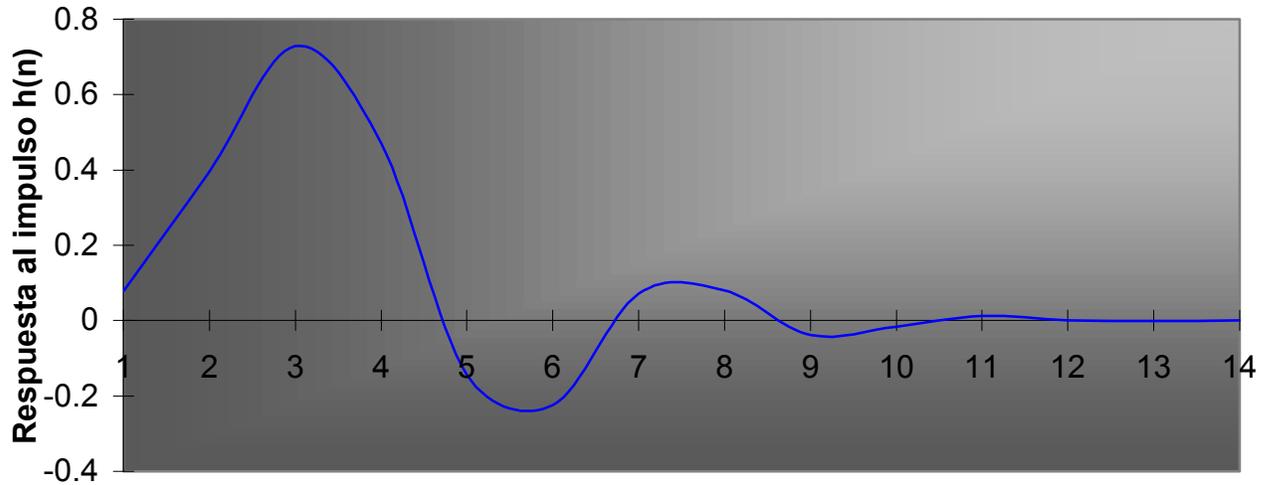
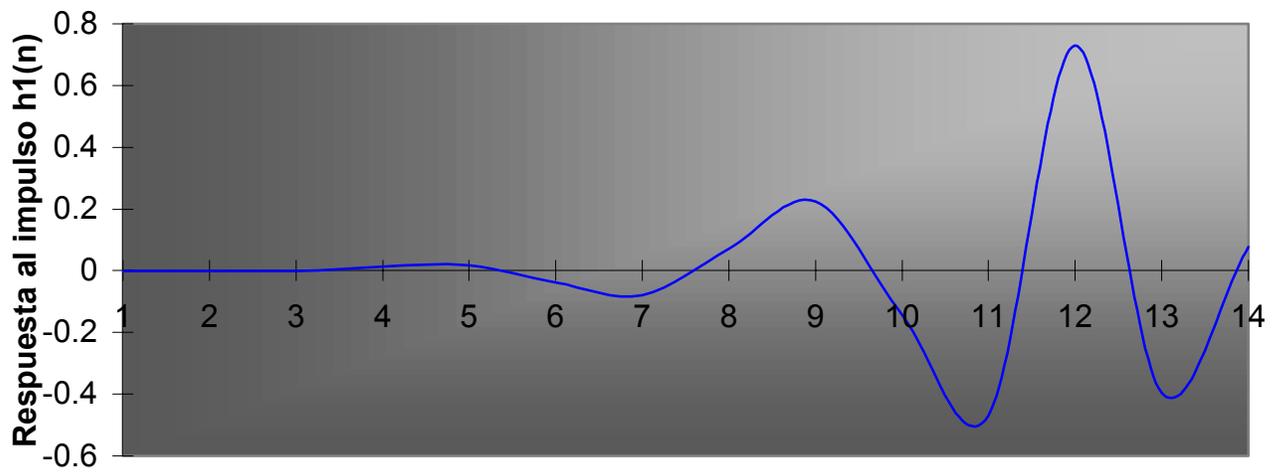
FAMILIA DAUBECHIES

*Filtro Escala Daubechies 10.**Filtro Wavelet Daubechies 10.*

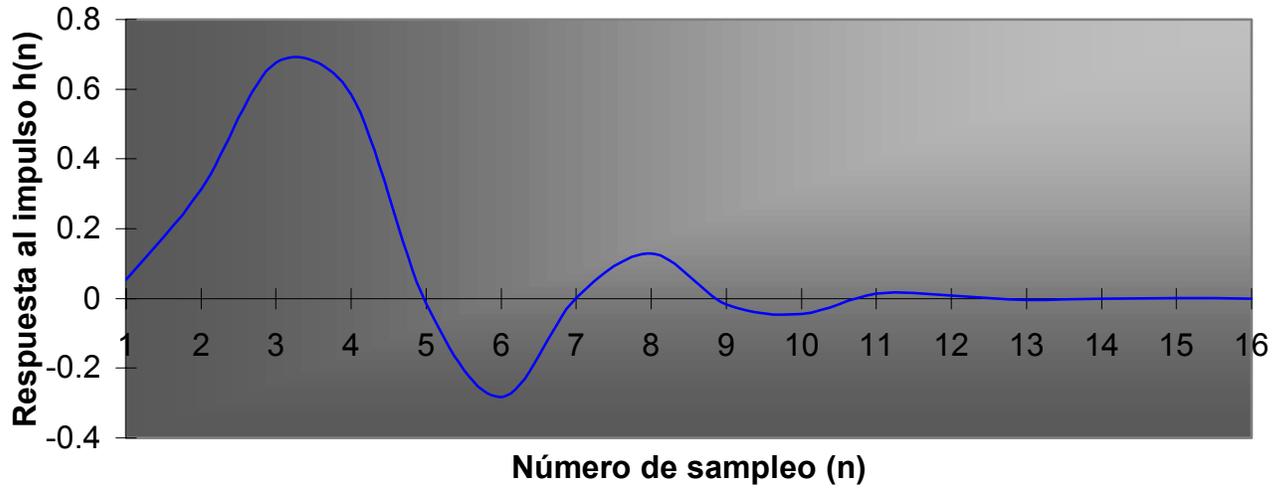
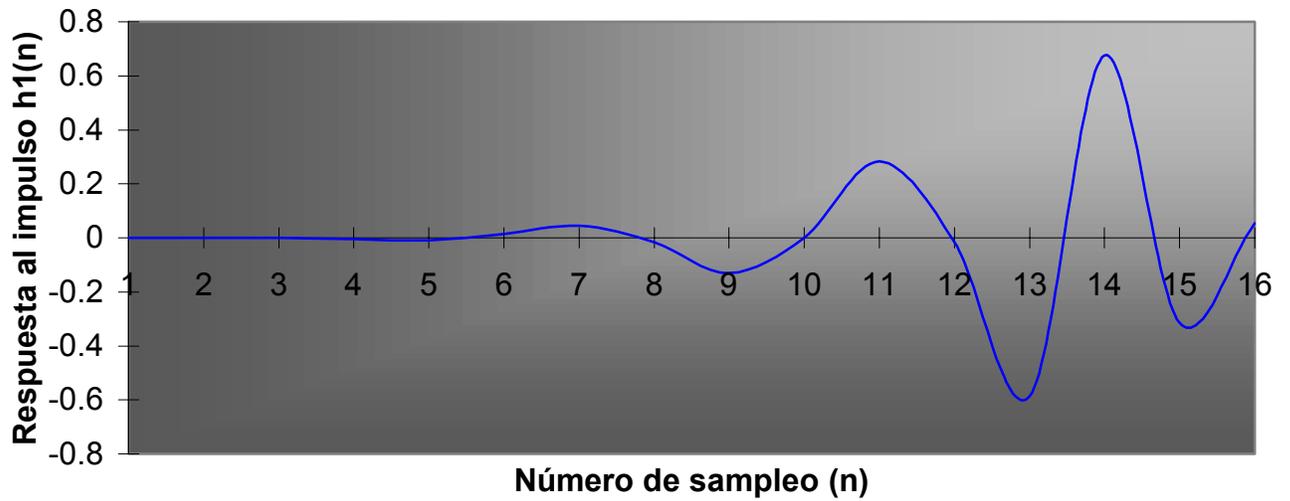
FAMILIA DAUBECHIES

*Filtro Escala Daubechies 12 .**Filtro Wavelet Daubechies 12.*

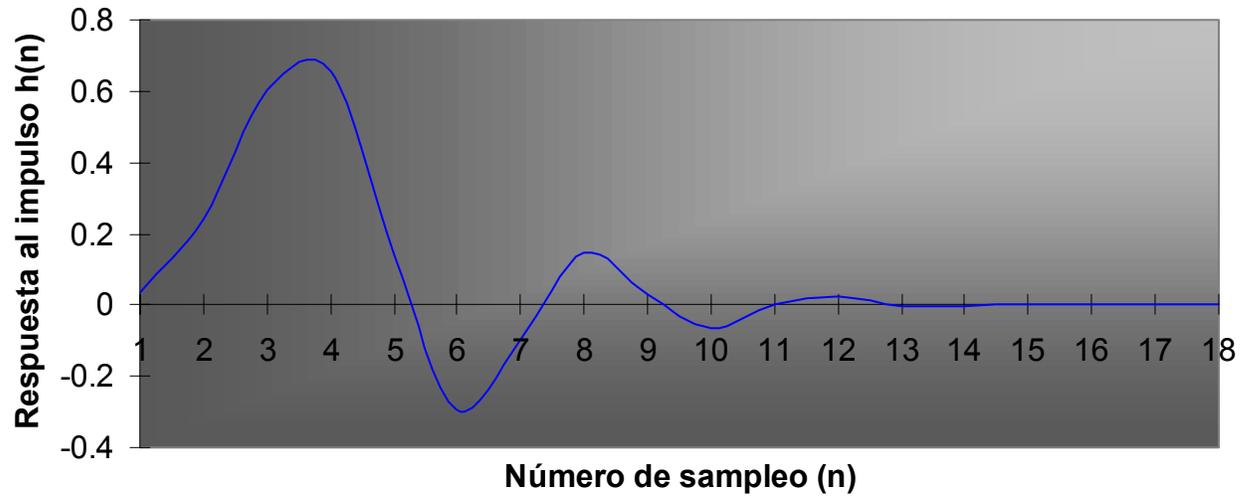
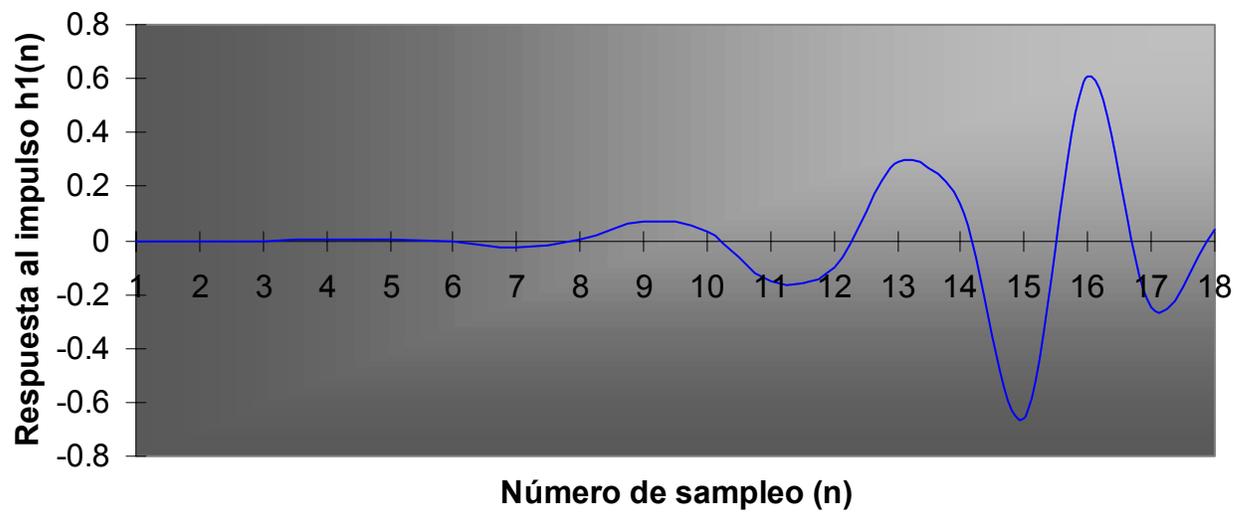
FAMILIA DAUBECHIES

**Número de sampleo (n)***Filtro Escala Daubechies 14.***Número de sampleo (n)***Filtro Wavelet Daubechies 14.*

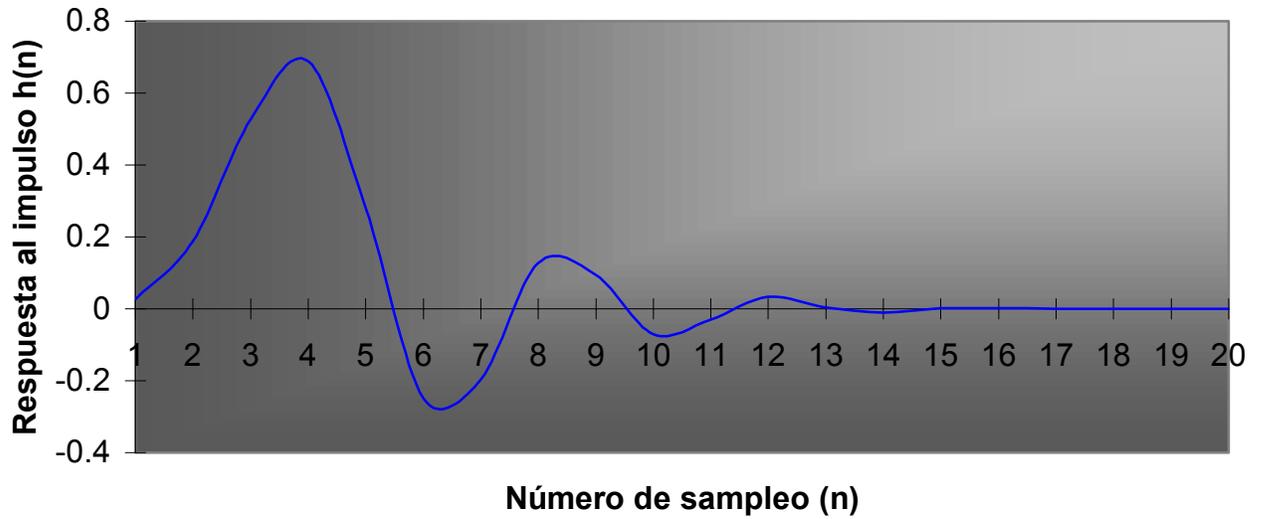
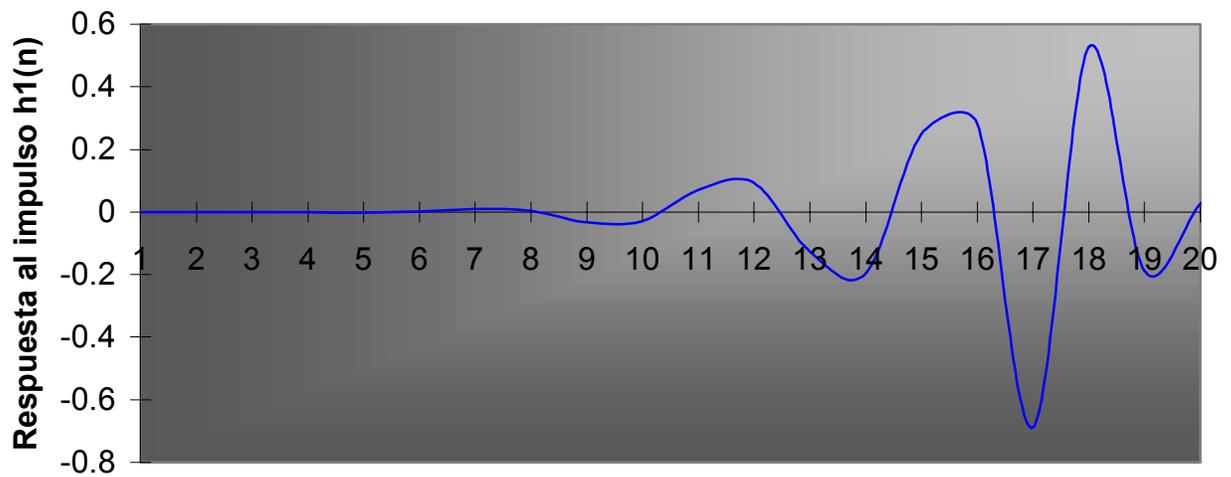
FAMILIA DAUBECHIES

*Filtro Escala Daubechies 16.**Filtro Wavelet Daubechies 16.*

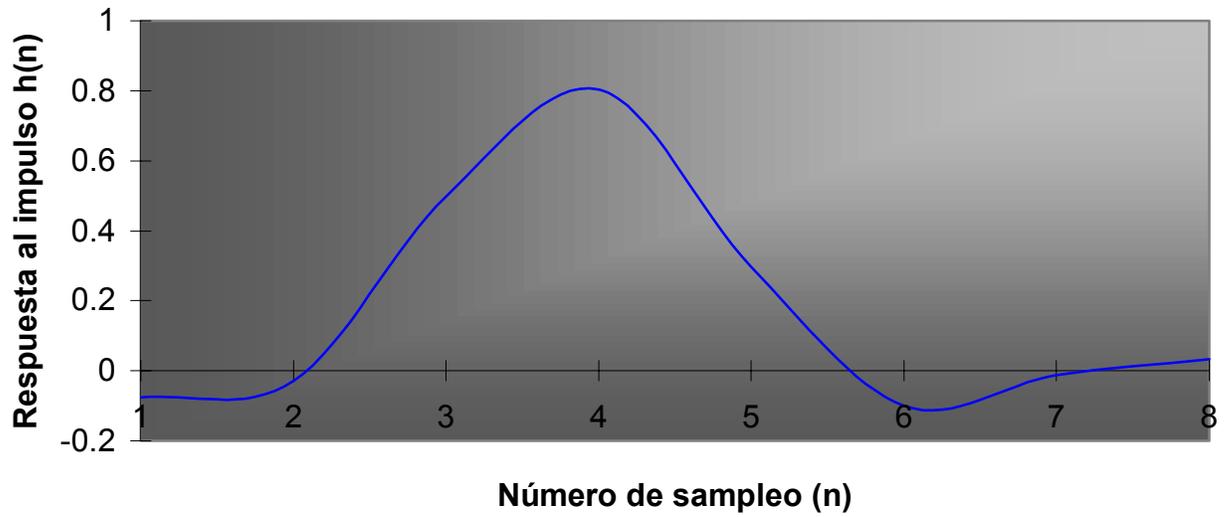
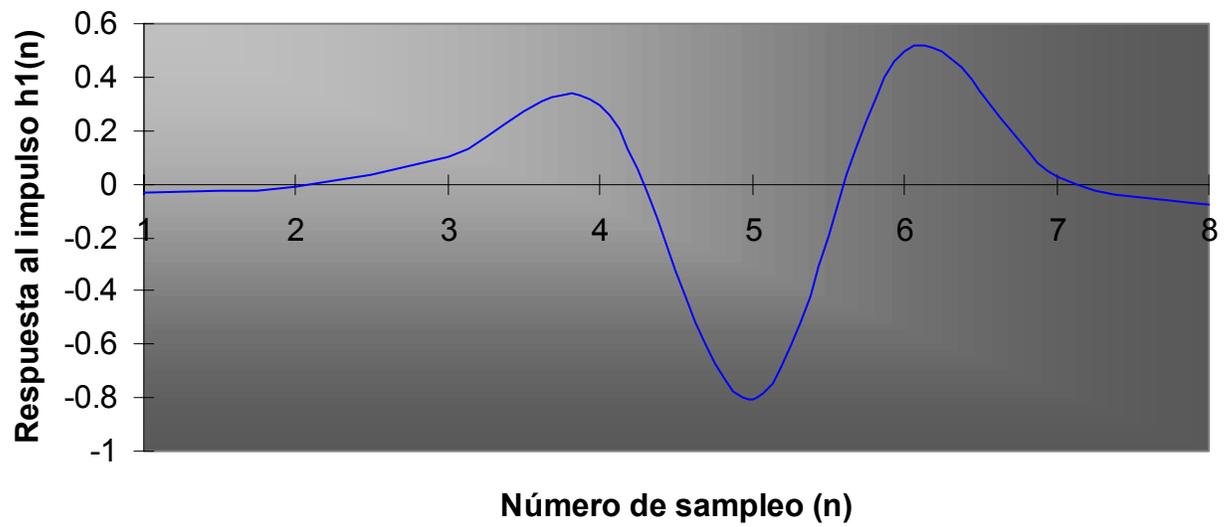
FAMILIA DAUBECHIES

*Filtro Escala Daubechies 18.**Filtro Wavelet Daubechies 18.*

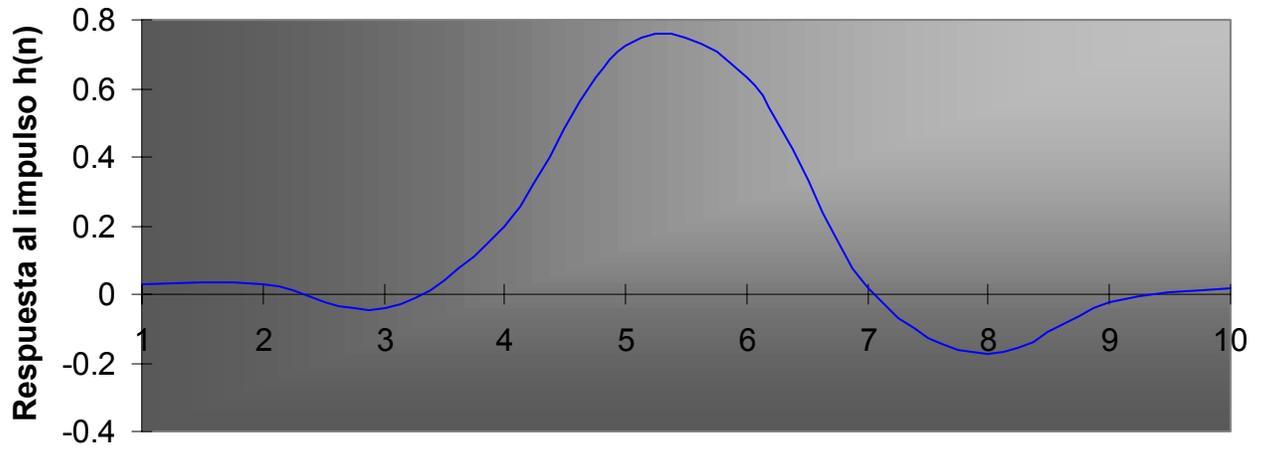
FAMILIA DAUBECHIES

**Número de sampleo (n)***Filtro Escala Daubechies 20.***Número de sampleo (n)***Filtro Wavelet Daubechies 20.*

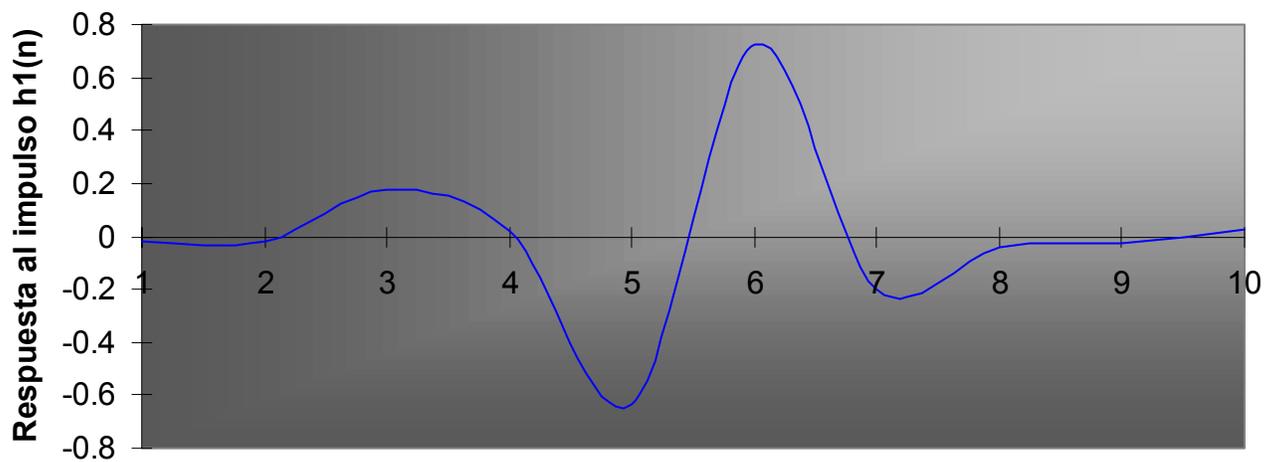
FAMILIA SIMLETS

*Filtro Escala Simlet 4.**Filtro Wavelet Simlet 4.*

FAMILIA SIMLETS



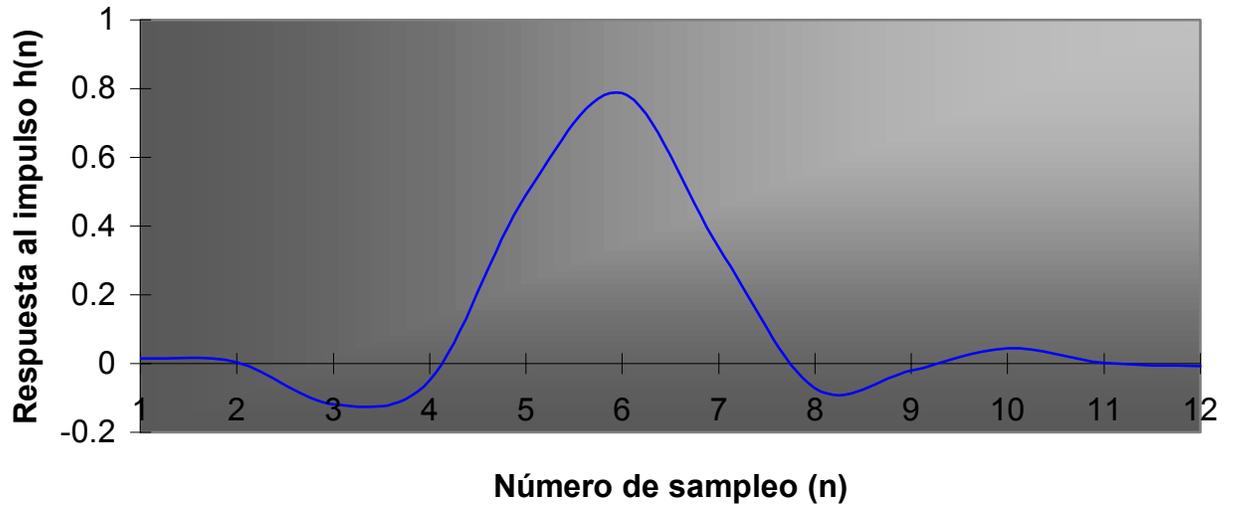
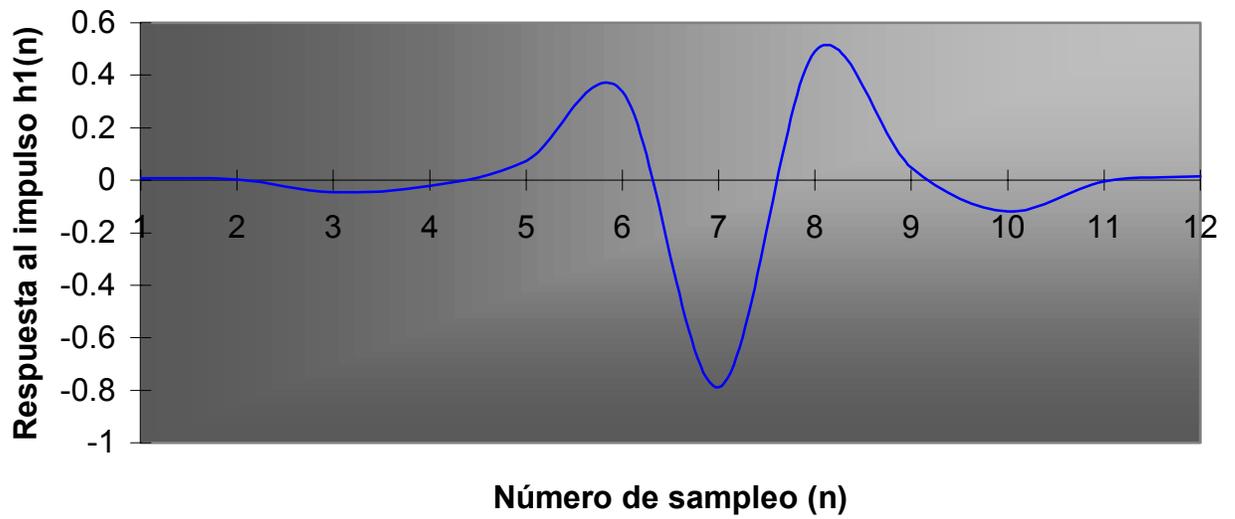
Número de sampleo (n)

Filtro Escala Simlet 5.

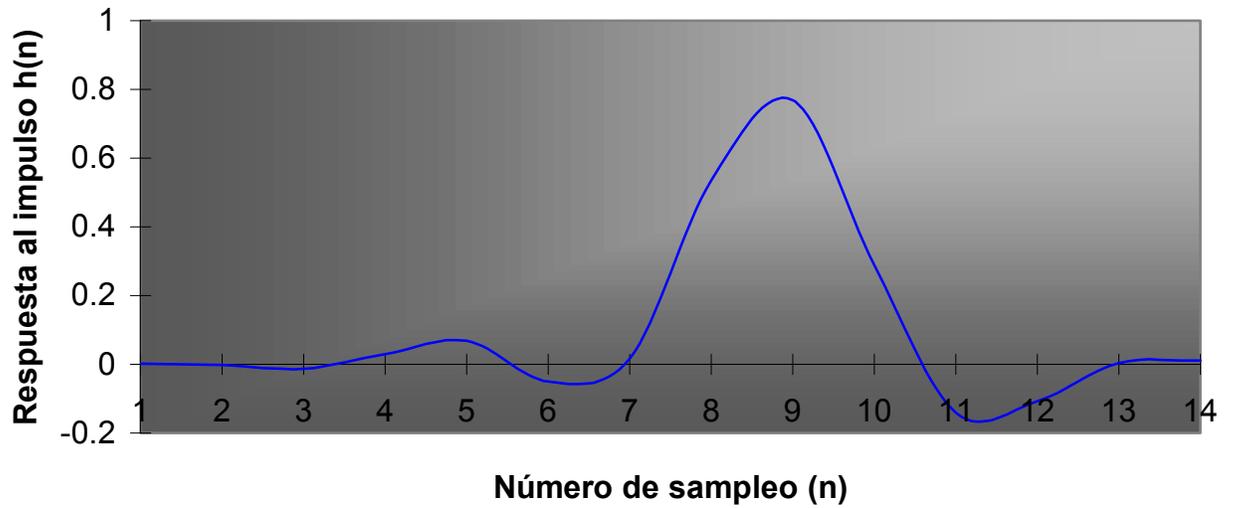
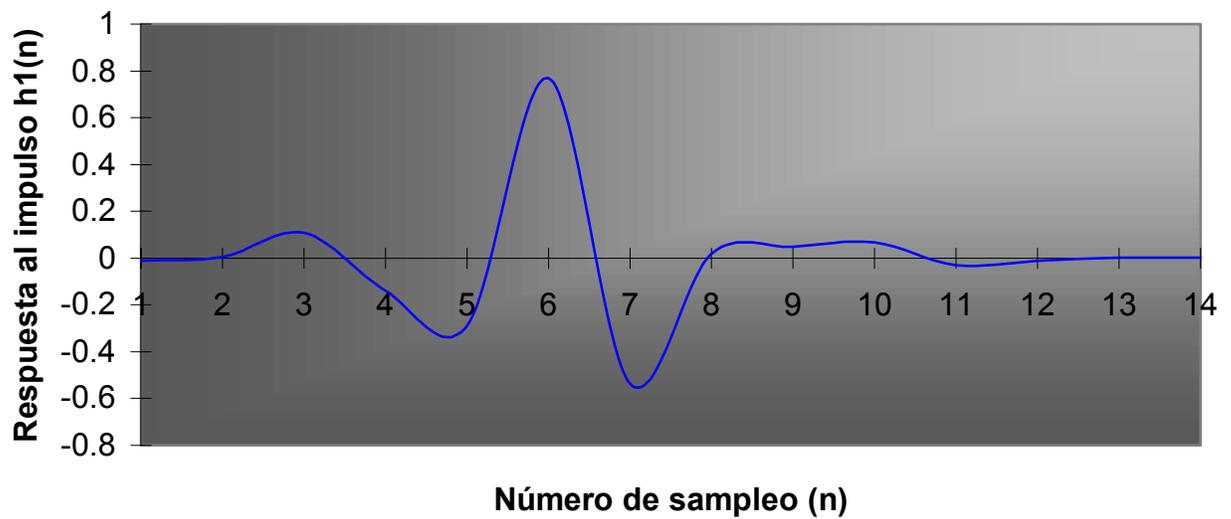
Número de sampleo (n)

Filtro Wavelet Simlet 5.

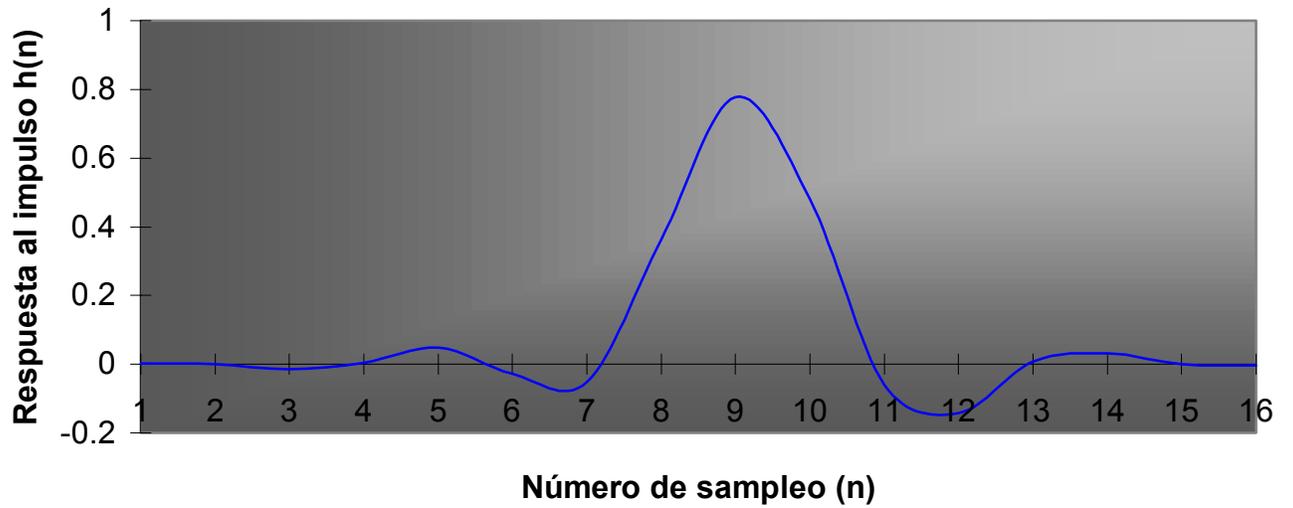
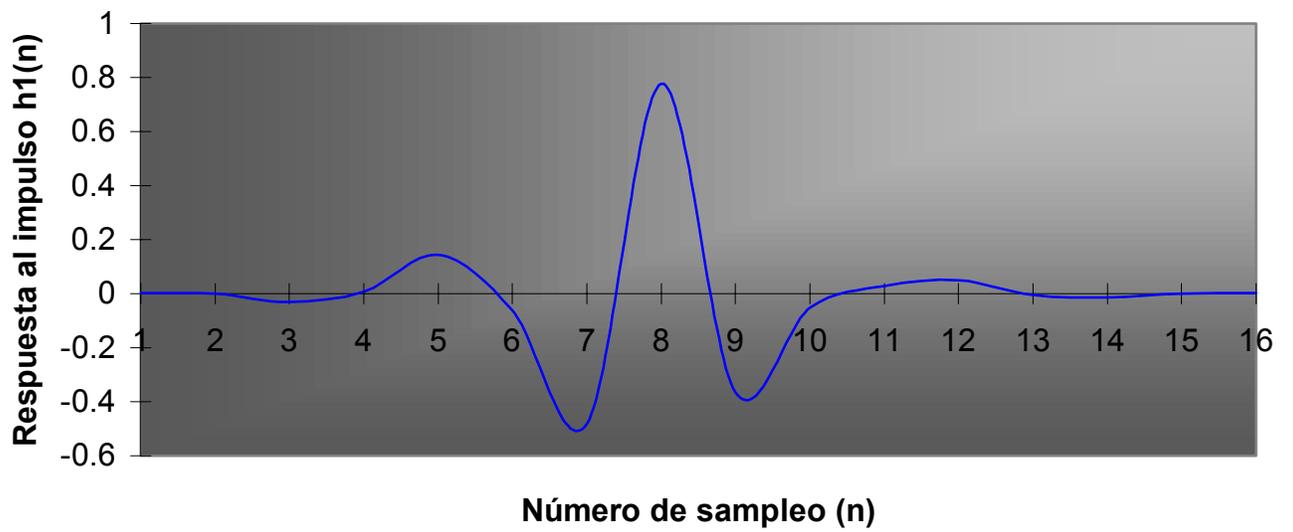
FAMILIA SIMLETS

*Filtro Escala Simlet 6.**Filtro Wavelet Simlet 6.*

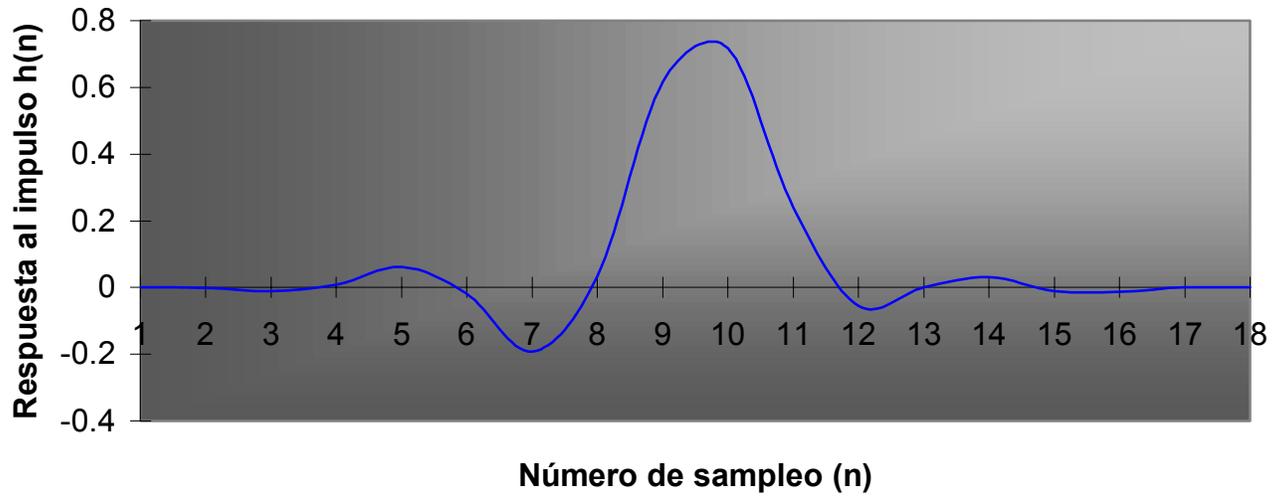
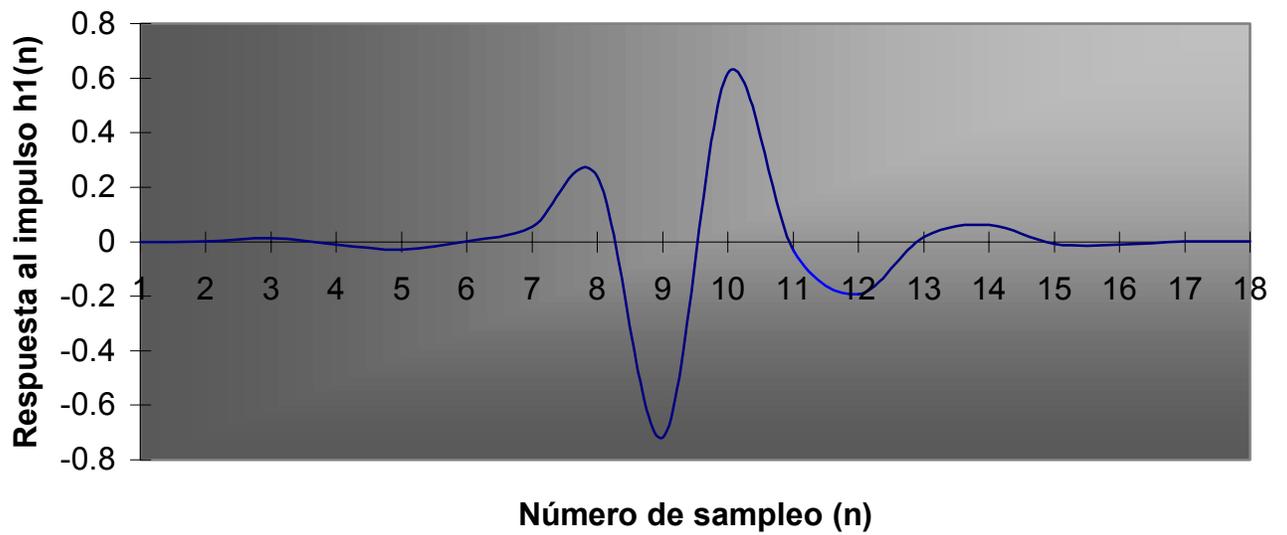
FAMILIA SIMLETS

*Filtro Escala Simlet 7.**Filtro Wavelet Simlet 7.*

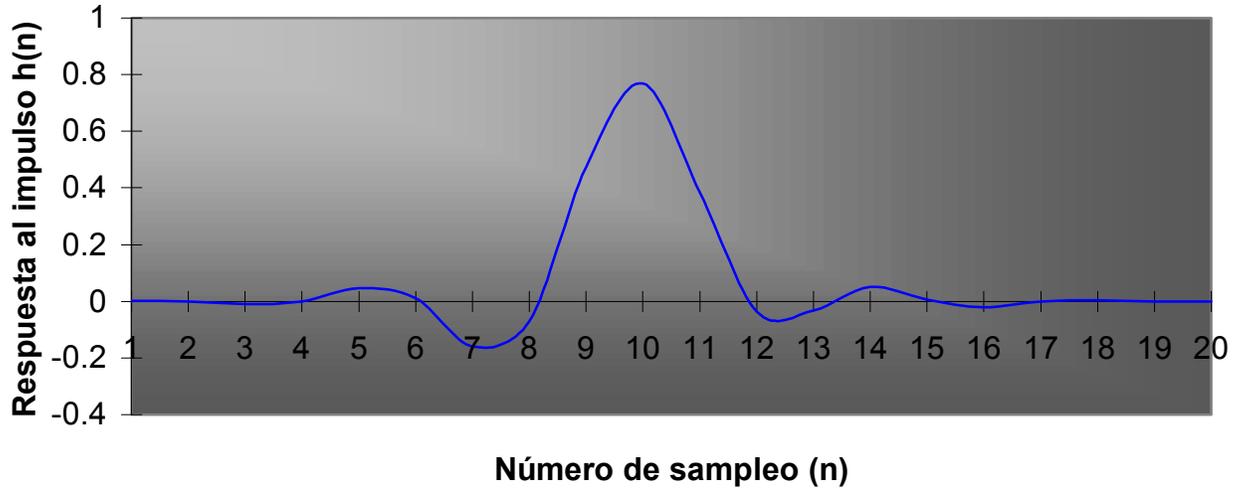
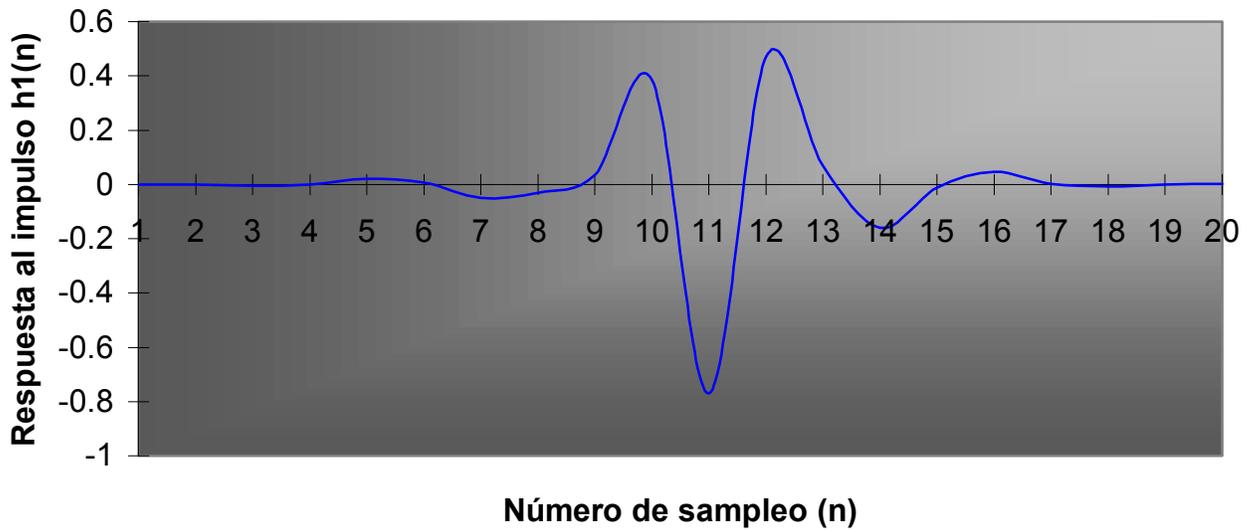
FAMILIA SIMLETS

*Filtro Escala Simlet 8.**Filtro Wavelet Simlet 8.*

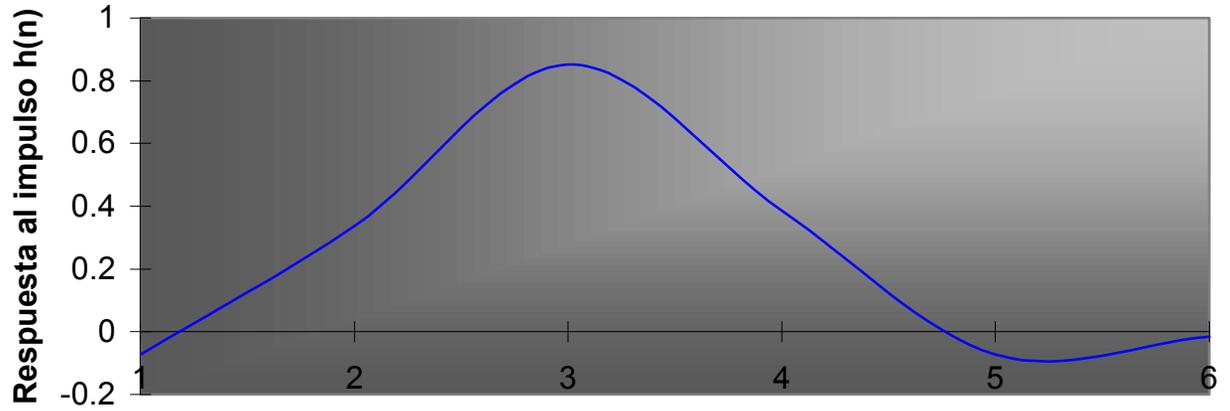
FAMILIA SIMLETS

*Filtro Escala Simlet 9.**Filtro Wavelet Simlet 9.*

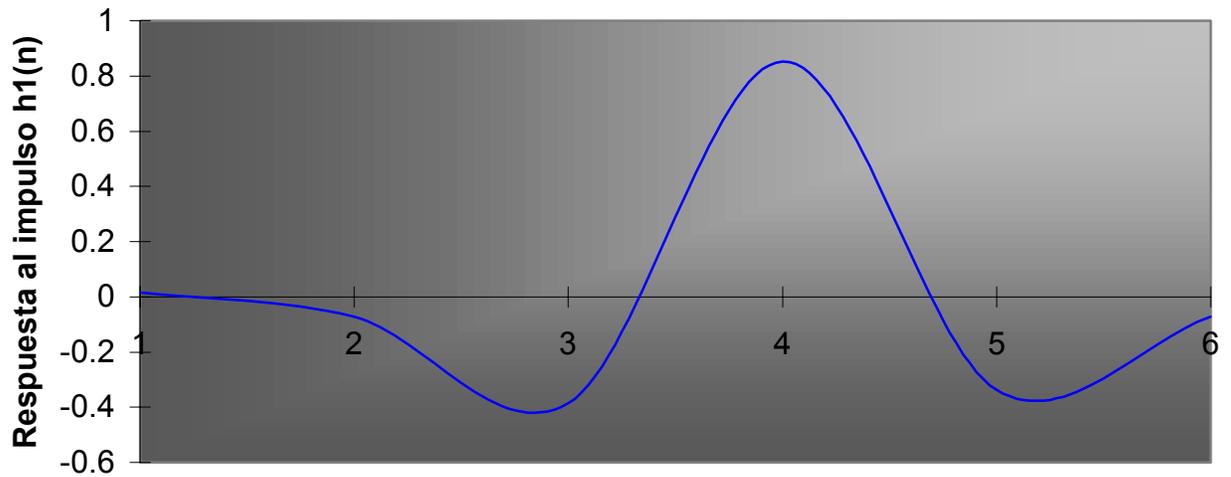
FAMILIA SIMLETS

*Filtro Escala Simlet 10.**Filtro Wavelet Simlet 10.*

WAVELETS COIFLETS



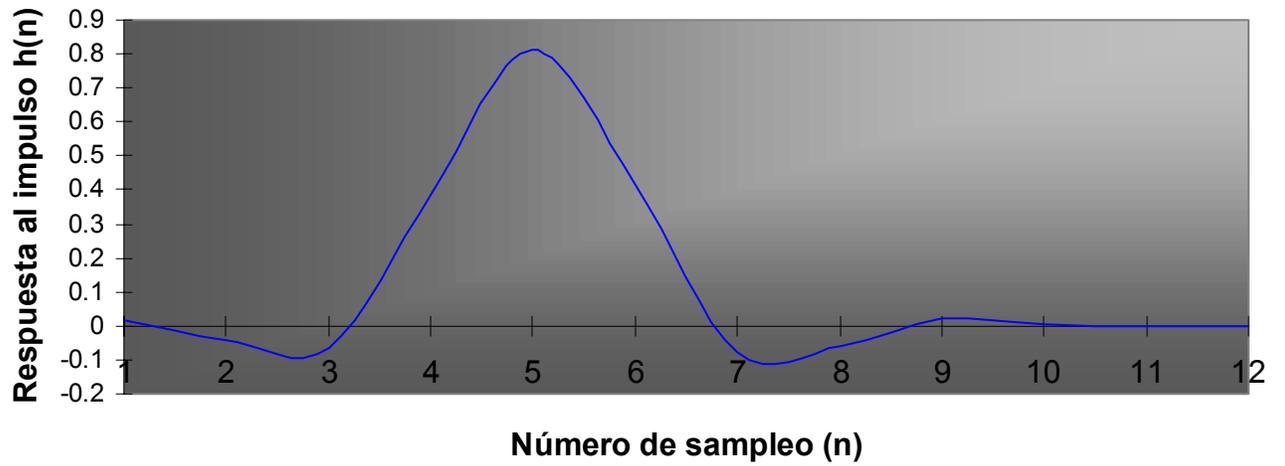
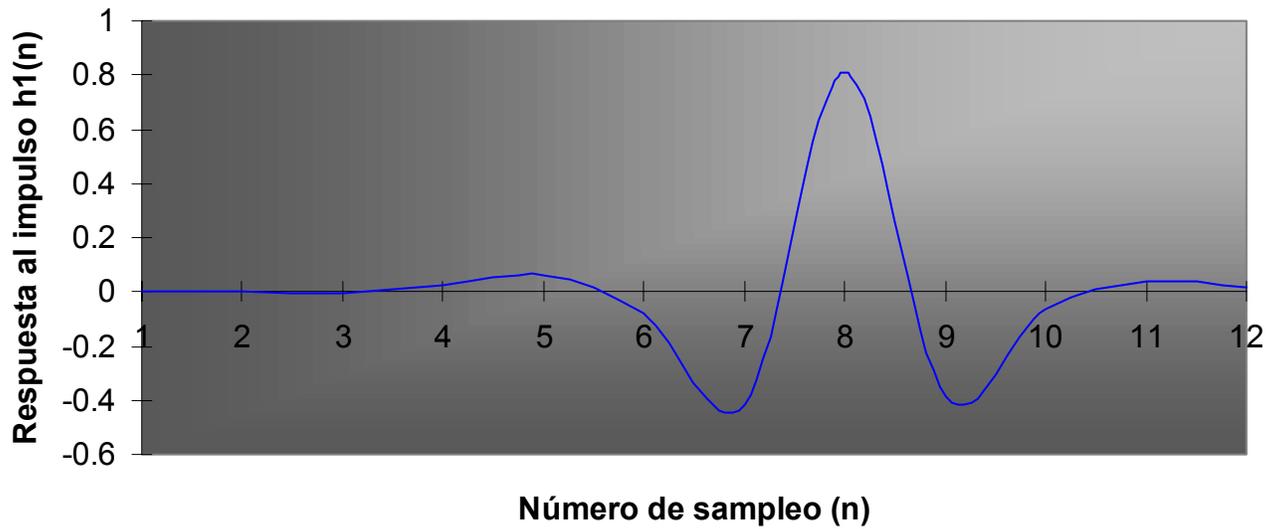
Número de sampleo (n)

Filtro Escala Coiflet 1.

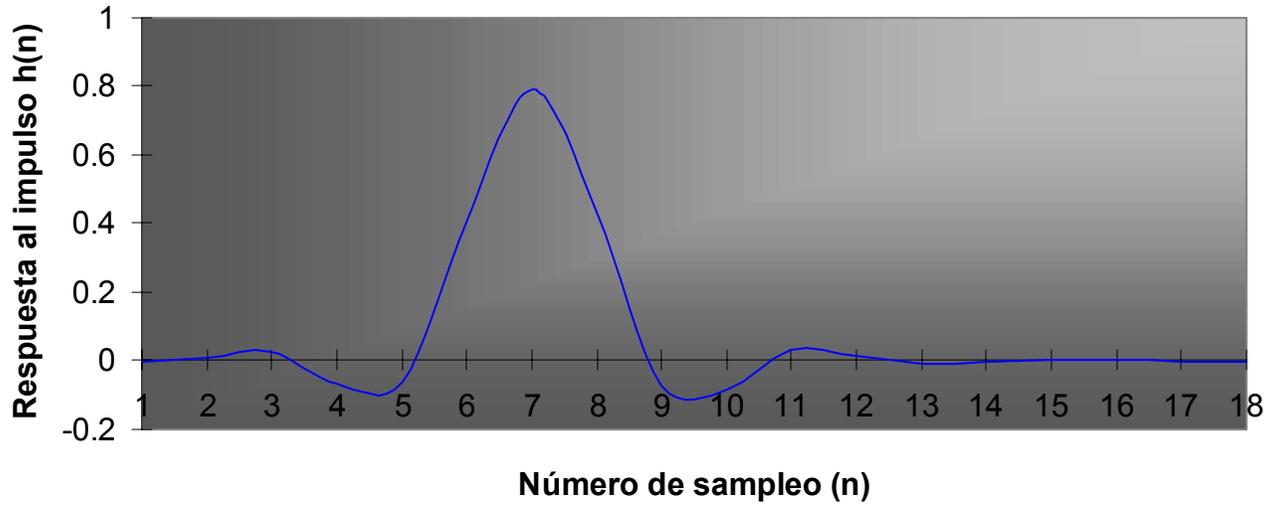
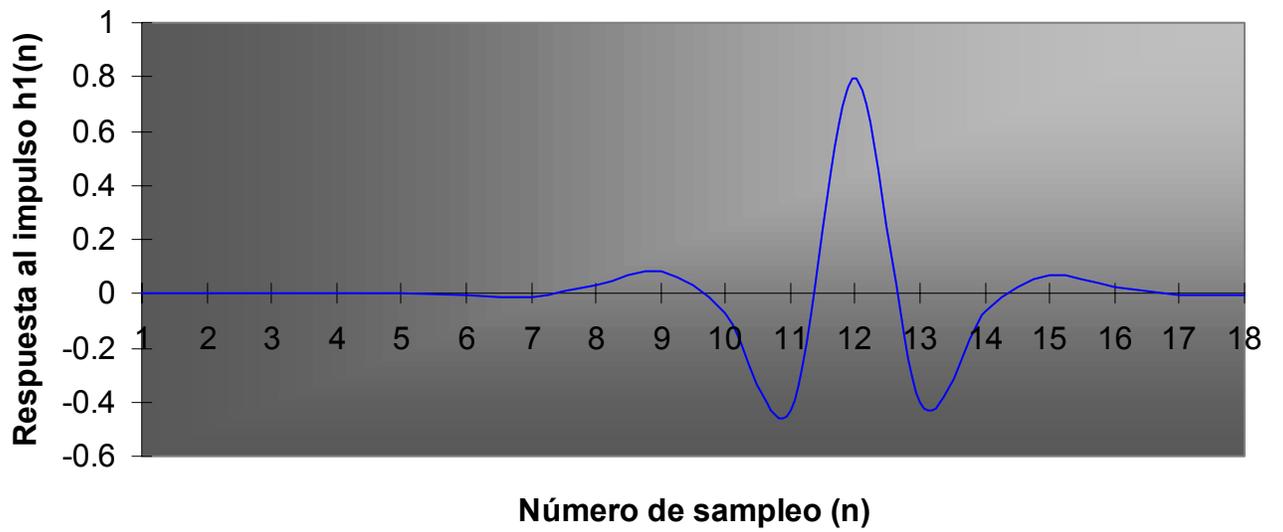
Número de sampleo (n)

Filtro Wavelet Coiflet 1.

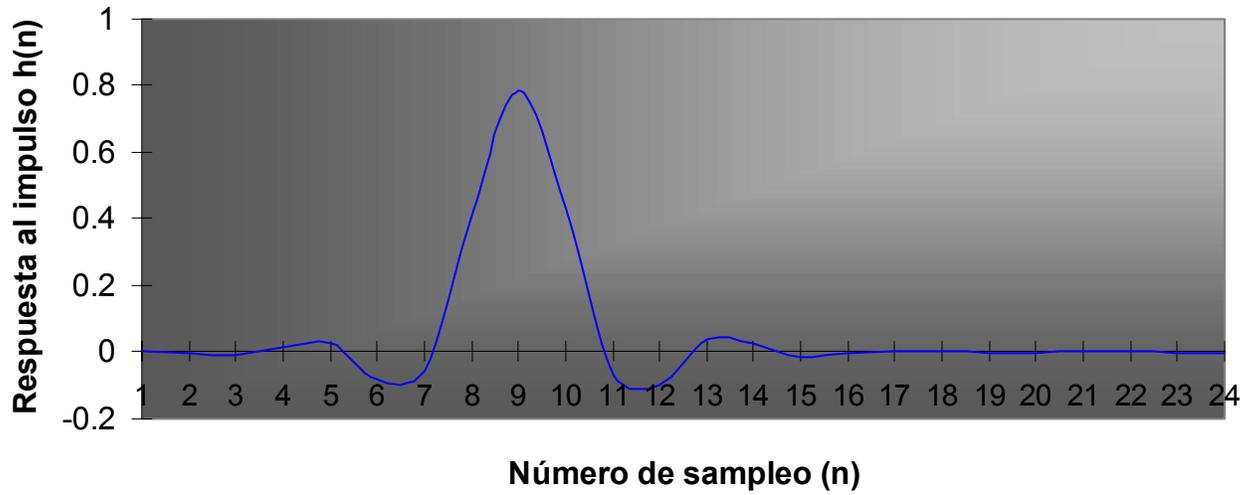
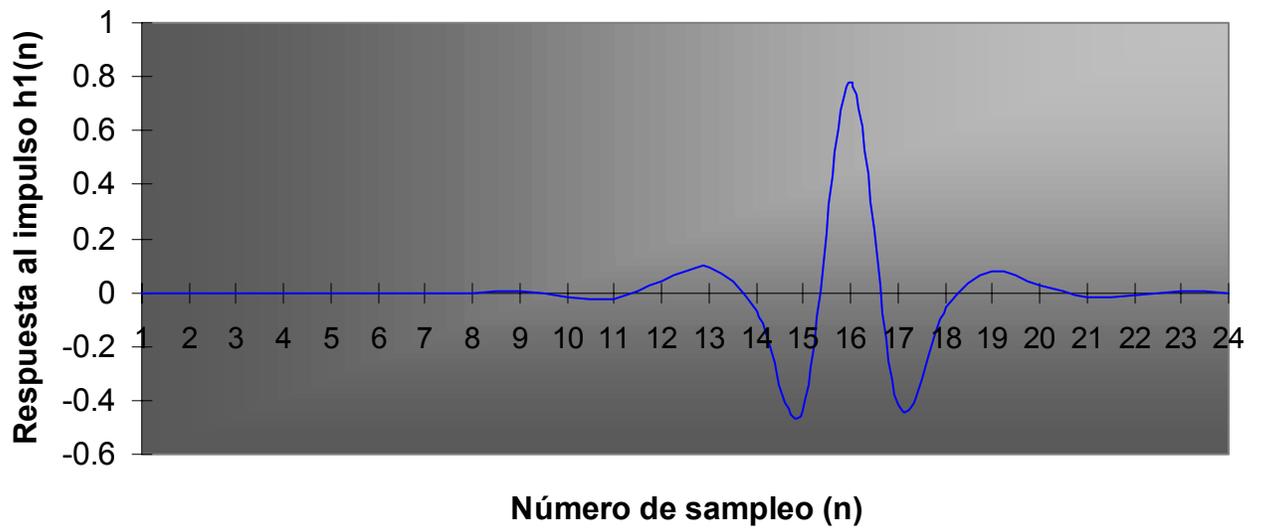
WAVELETS COIFLETS

*Filtro Escala Coiflet 2.**Filtro Wavelet Coiflet 2.*

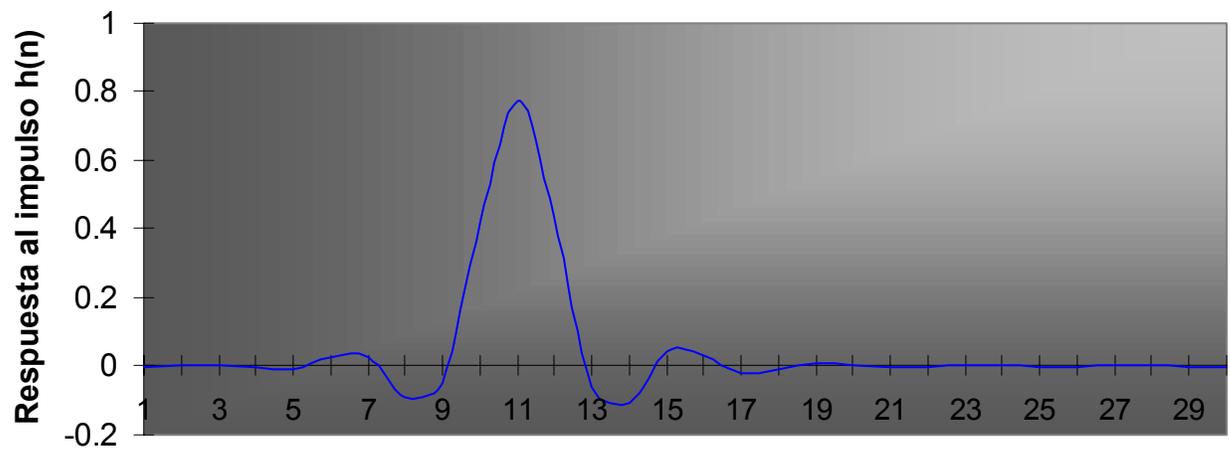
WAVELETS COIFLETS

*Filtro Escala Coiflet 3.**Filtro Wavelet Coiflet 3.*

WAVELETS COIFLETS

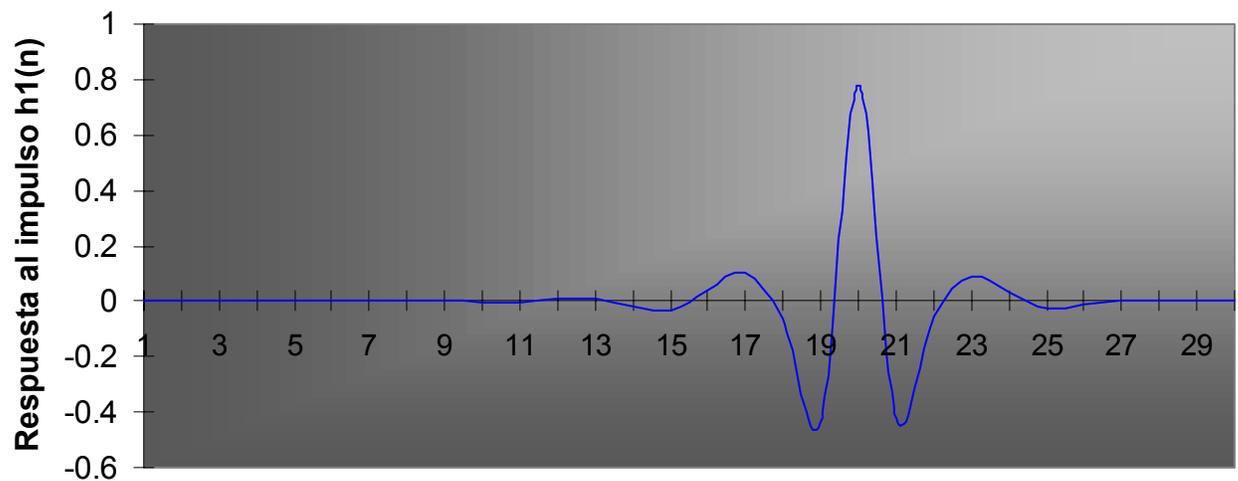
*Filtro Escala Coiflet 4.**Filtro Wavelet Coiflet 4.*

WAVELETS COIFLETS



Número de sampleo (n)

Filtro Escala Coiflet 5.

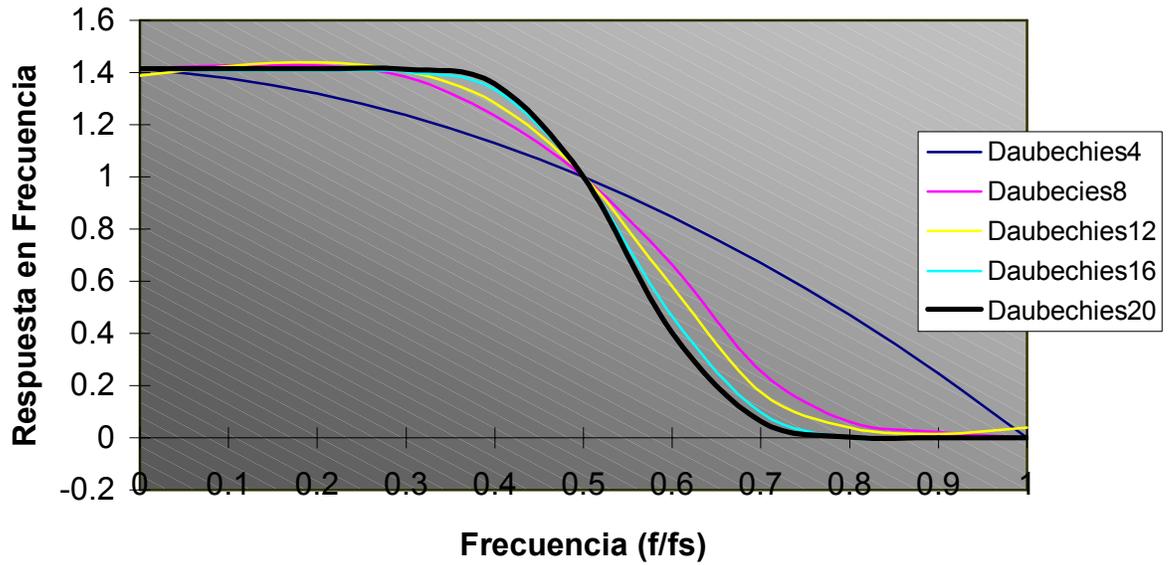


Número de sampleo (n)

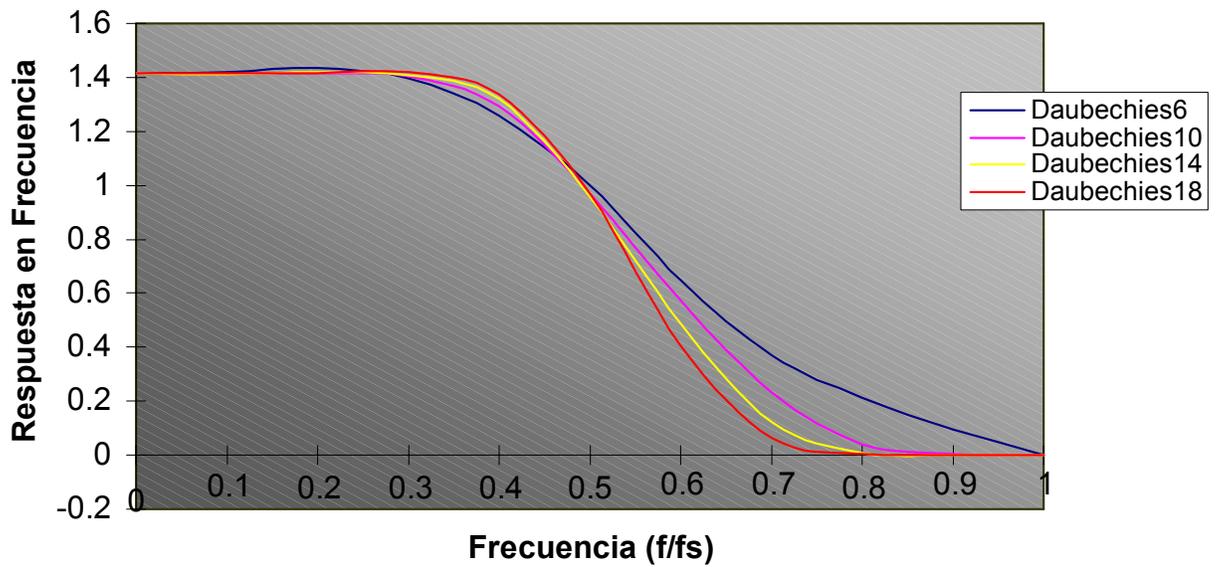
Filtro Wavelet Coiflet 5.

RESPUESTAS EN FRECUENCIA DE FILTROS ESCALA Y WAVELET

FAMILIA DAUBECHIES ESCALA

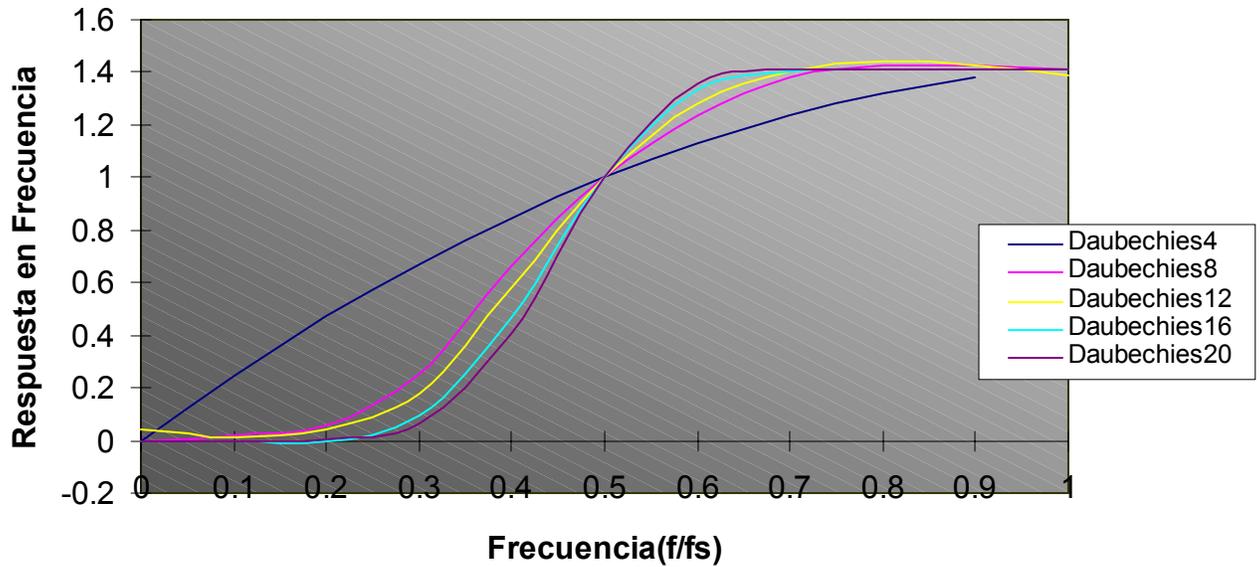


Respuestas en Frecuencia. Filtros Daubechies Escala.

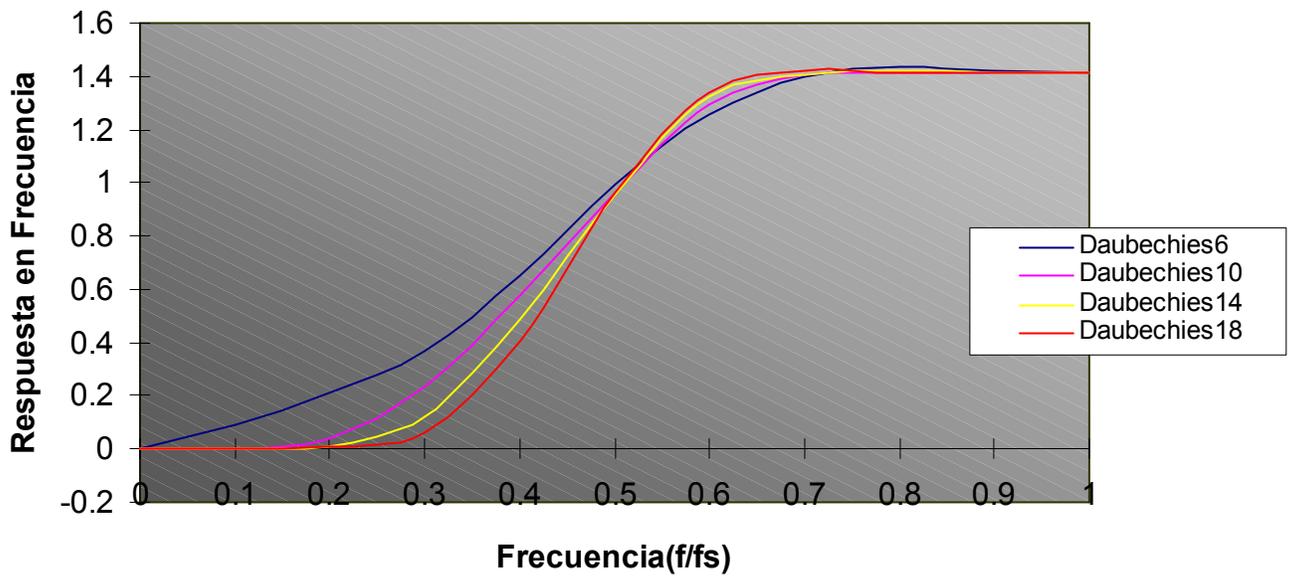


Respuestas en Frecuencia. Filtros Daubechies Escala.

FAMILIA DAUBECHIES WAVELET

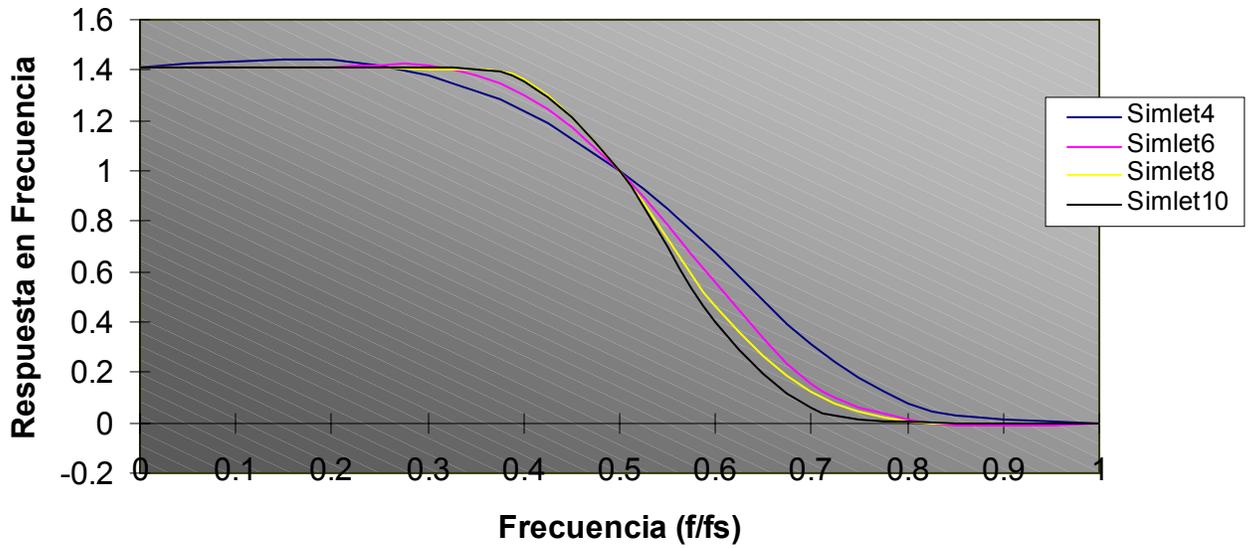


Respuestas en Frecuencia. Filtros Daubechies Wavelet.

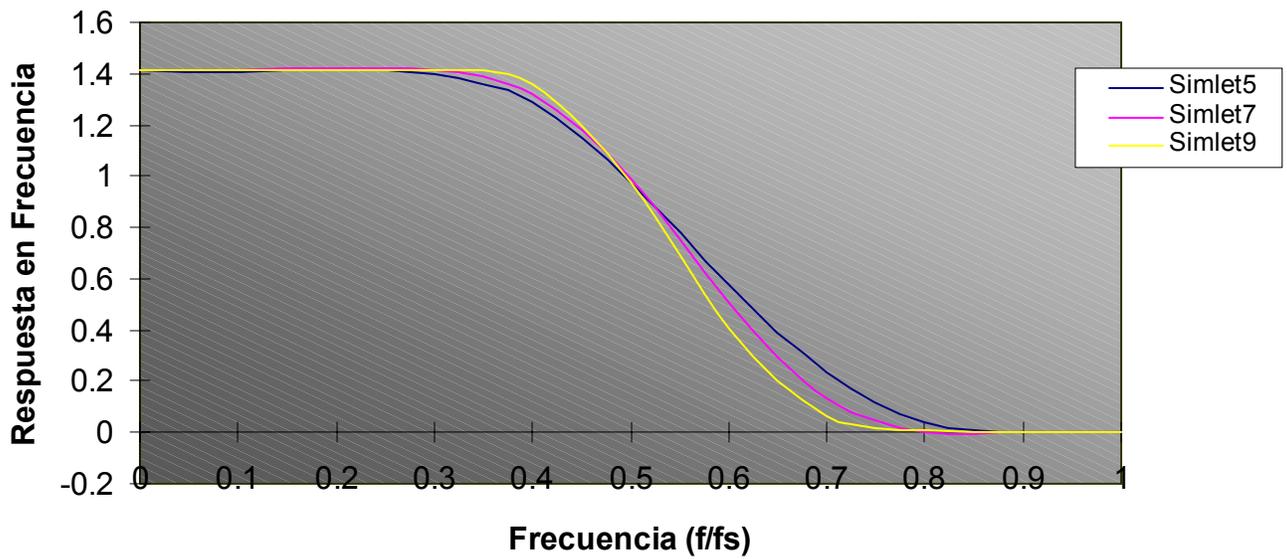


Respuestas en Frecuencia. Filtros Daubechies Wavelet.

FAMILIA SIMLET ESCALA

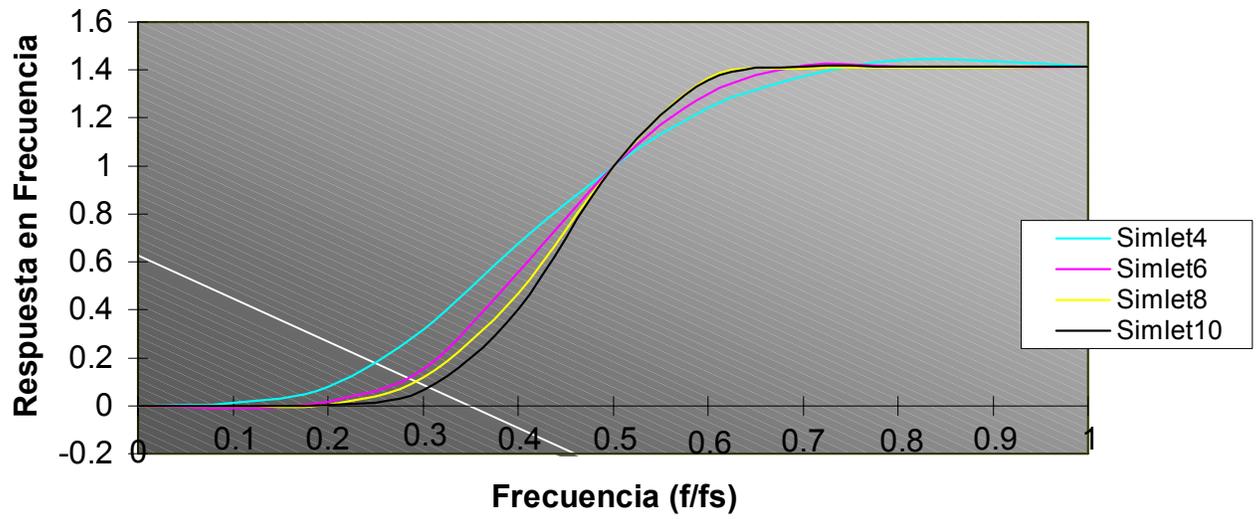


Respuestas en Frecuencia. Filtros Simlet Escala.

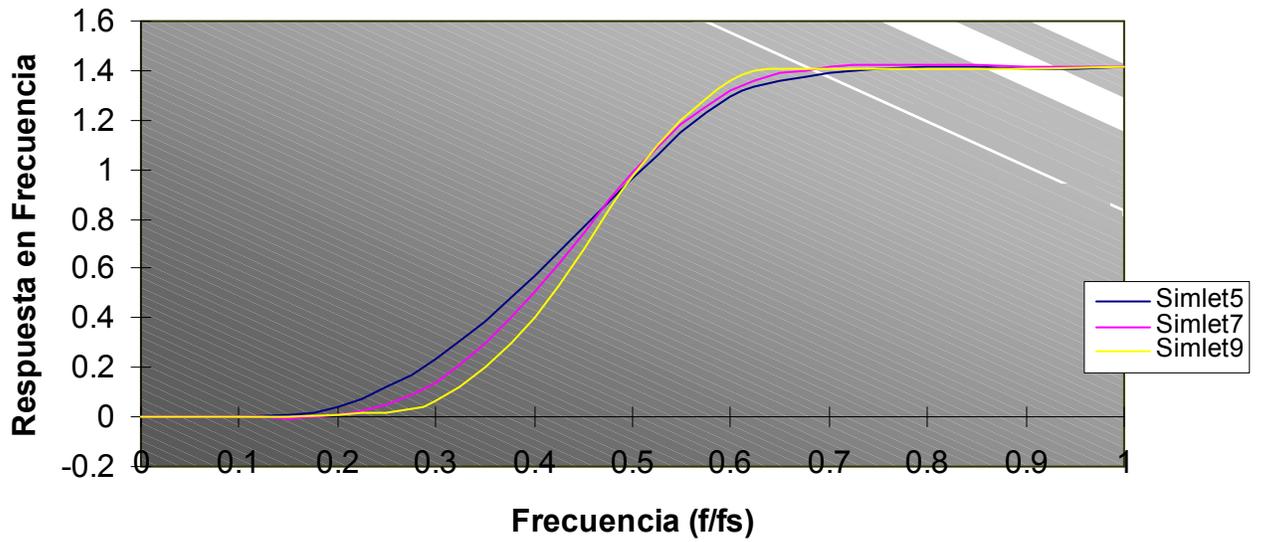


Respuestas en Frecuencia. Filtros Simlet Escala.

FAMILIA SIMLET WAVELET

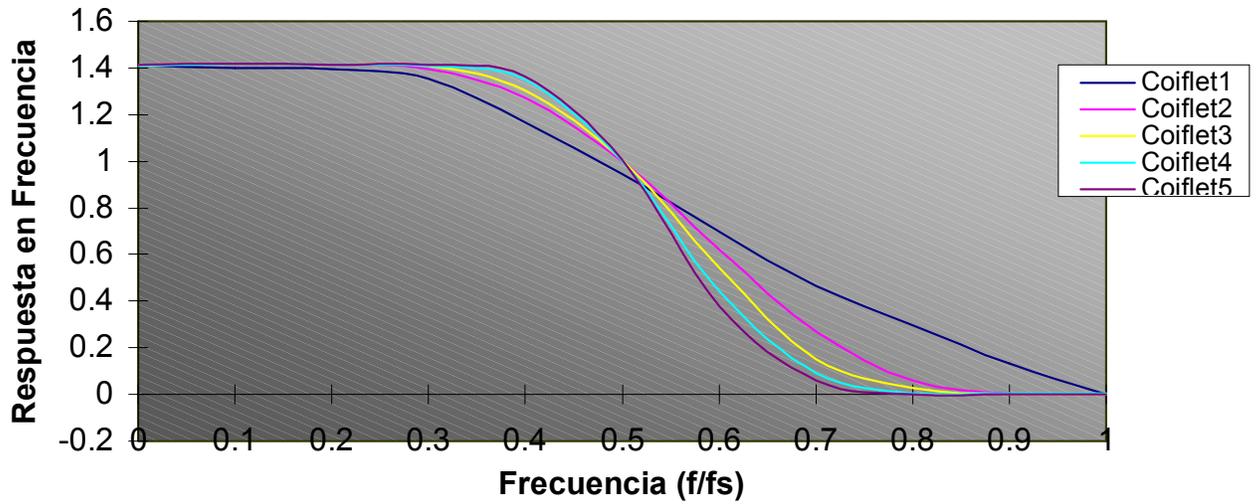


Respuestas en Frecuencia. Filtros Simlet Wavelet.



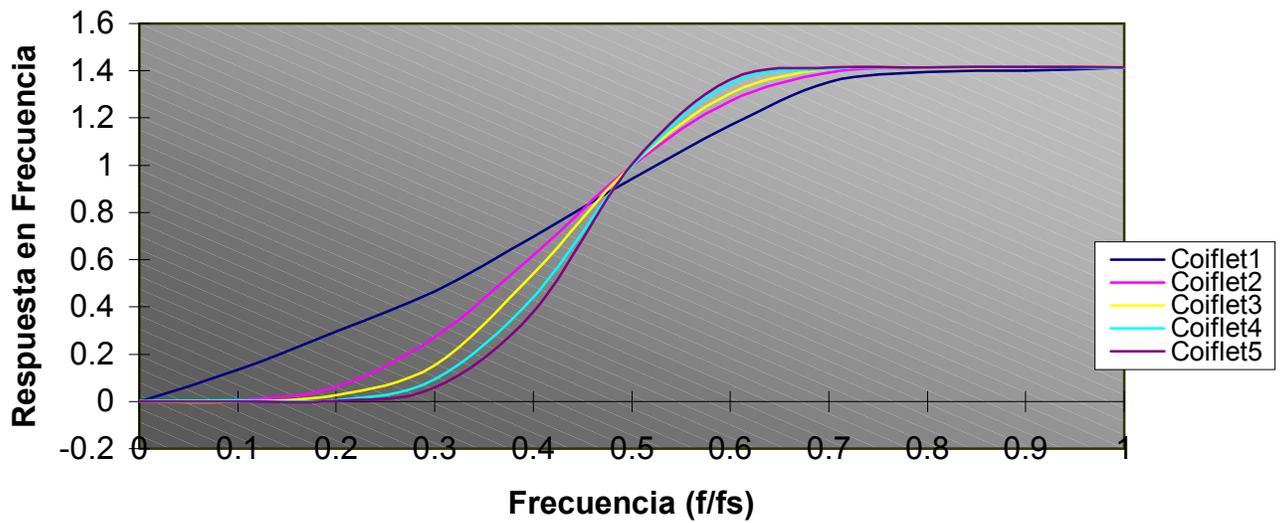
Respuestas en Frecuencia. Filtros Simlet Wavelet.

FAMILIA COIFLET ESCALA



Respuestas en Frecuencia. Filtros Coiflet Escala.

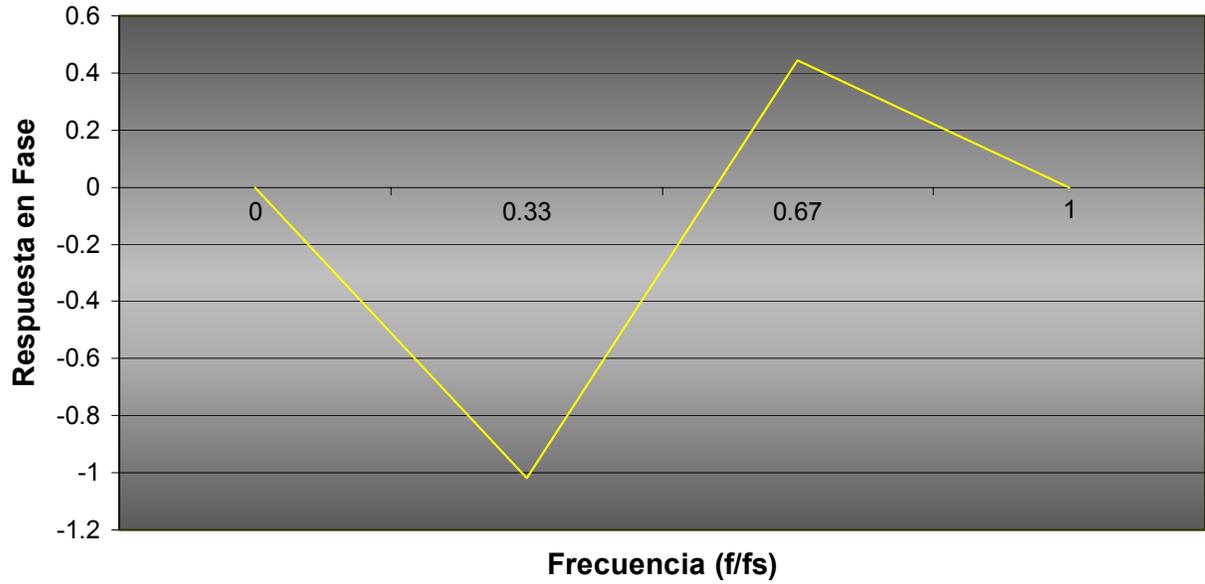
FAMILIA COIFLET WAVELET



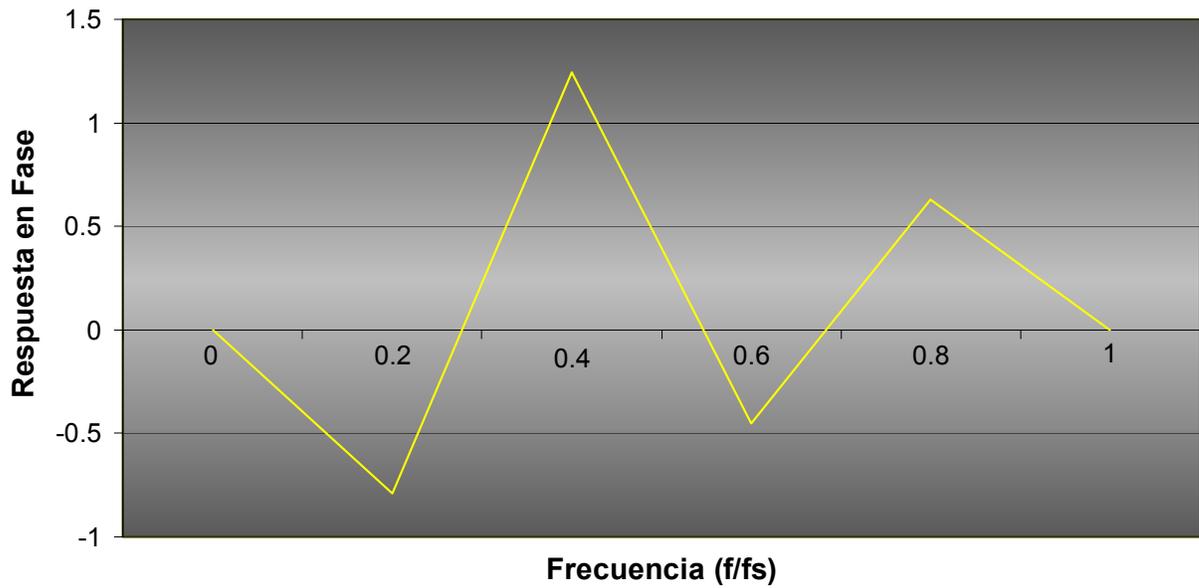
Respuestas en Frecuencia. Filtros Coiflet Wavelet.

APENDICE. RESPUESTAS EN FASE DE FILTROS ESCALA Y WAVELET

FAMILIA DAUBECHIES

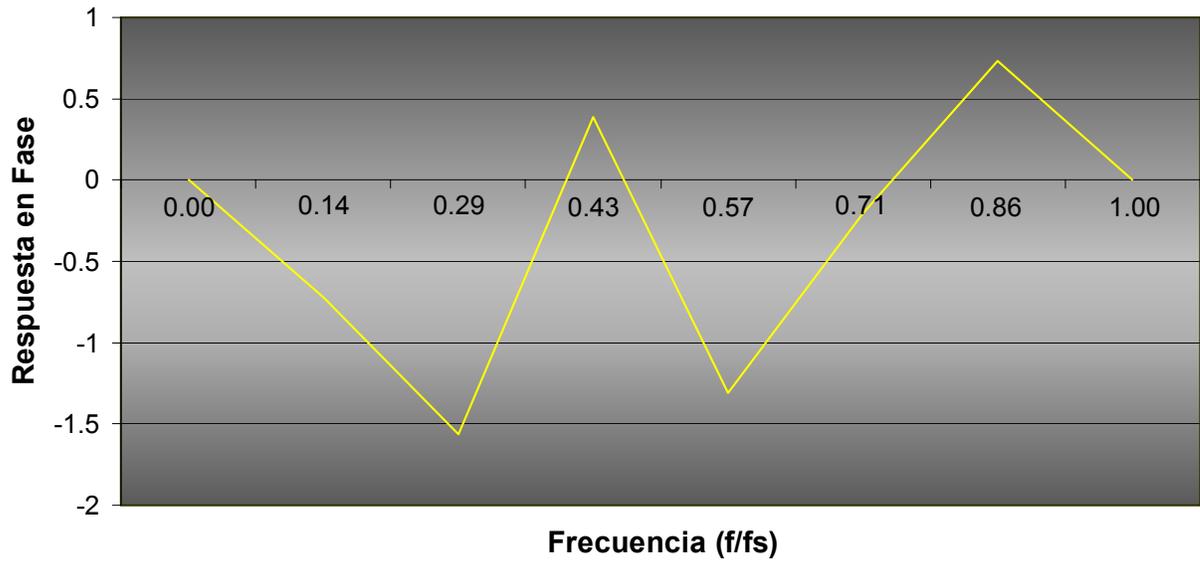


Respuesta en Fase. Filtro Daubechies6 Escala.

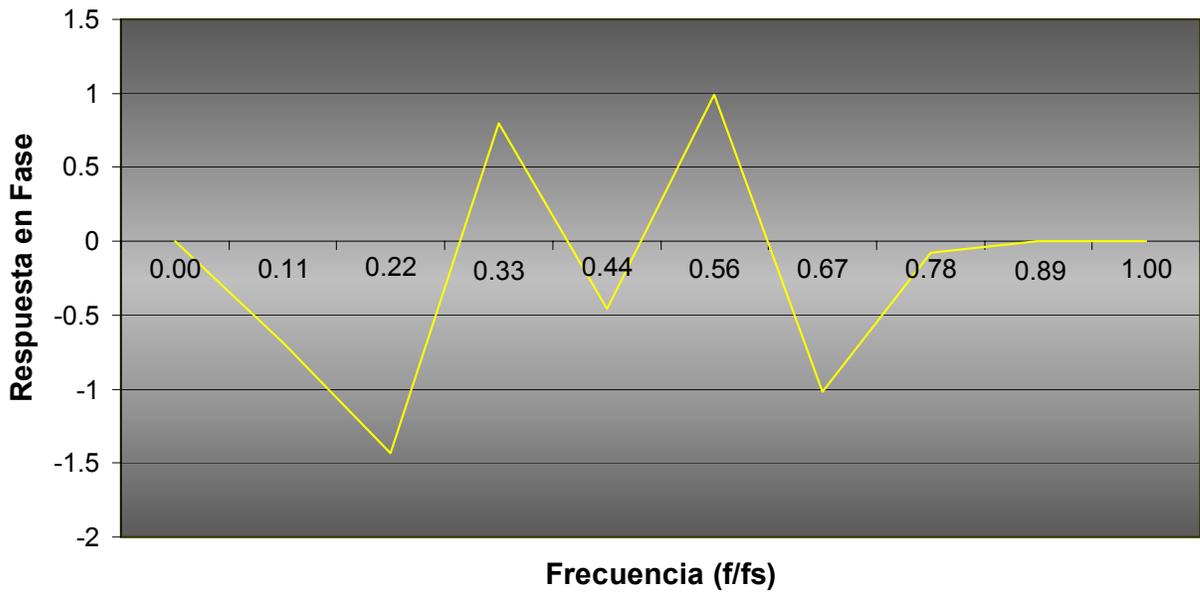


Respuesta en Fase. Filtro Daubechies10 Escala.

FAMILIA DAUBECHIES

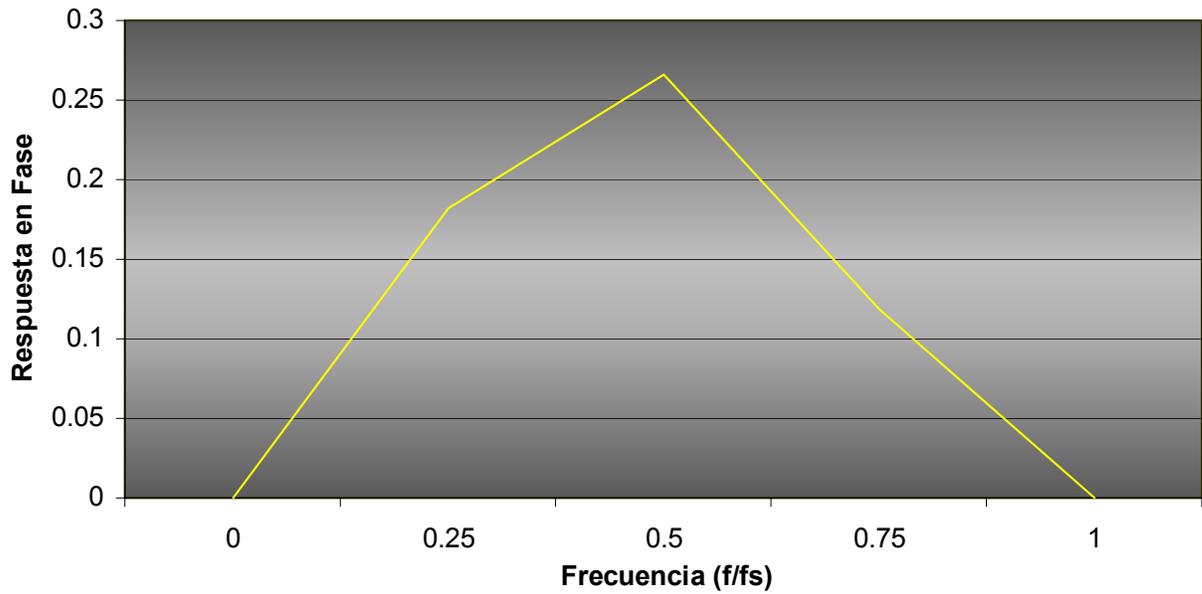


Respuesta en Fase. Filtro Daubechies14 Escala.

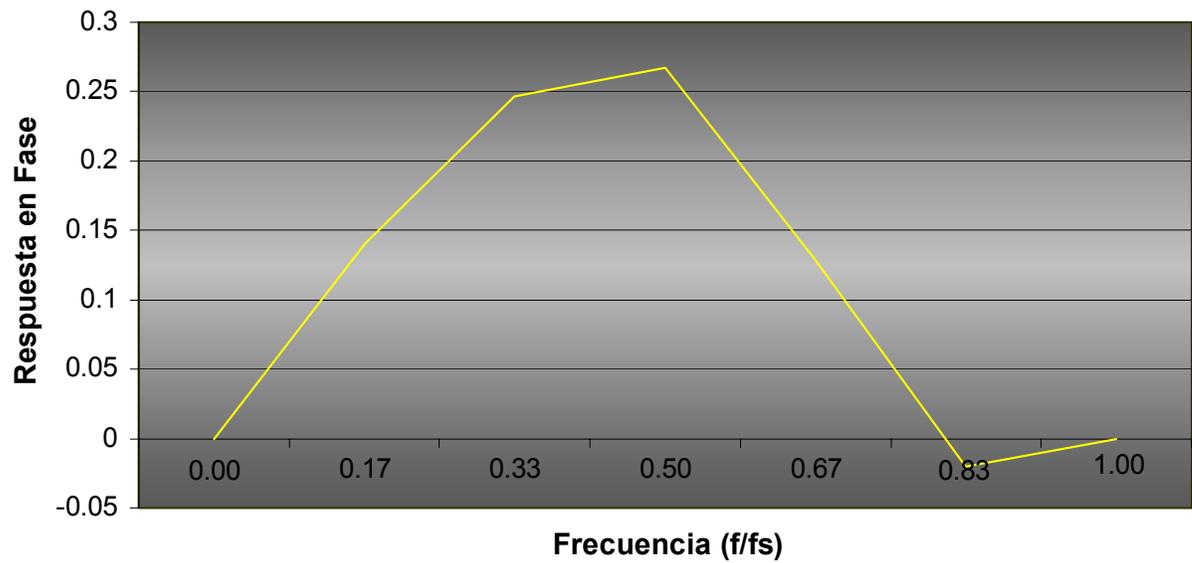


Respuesta en Fase. Filtro Daubechies18 Escala.

FAMILIA SIMLETS

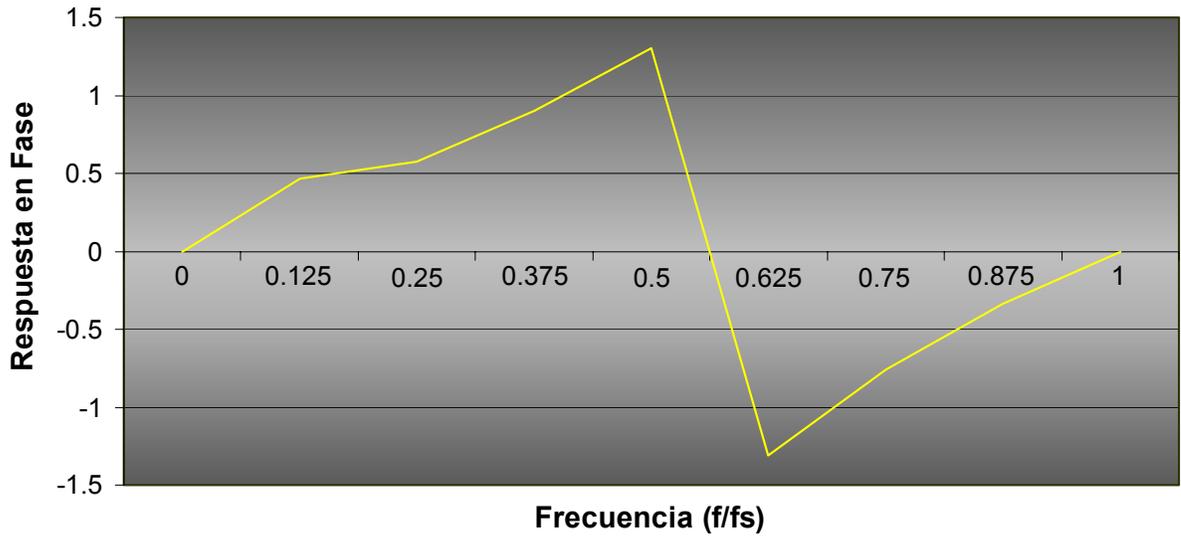


Respuesta en Fase. Filtro Simlet 4 Wavelet.

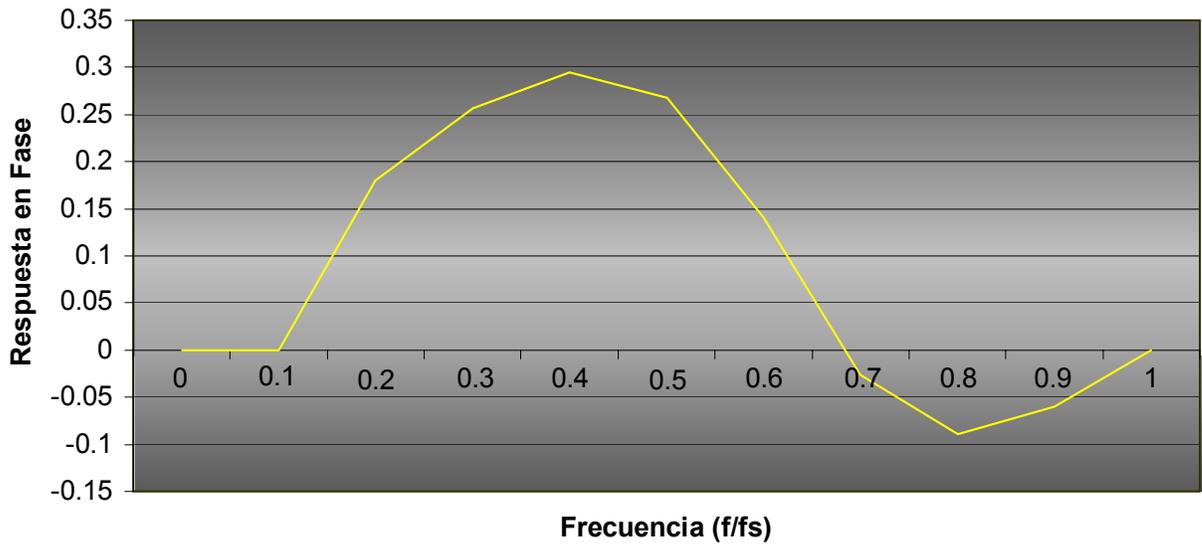


Respuesta en Fase. Filtro Simlet 6 Wavelet.

FAMILIA SIMLETS

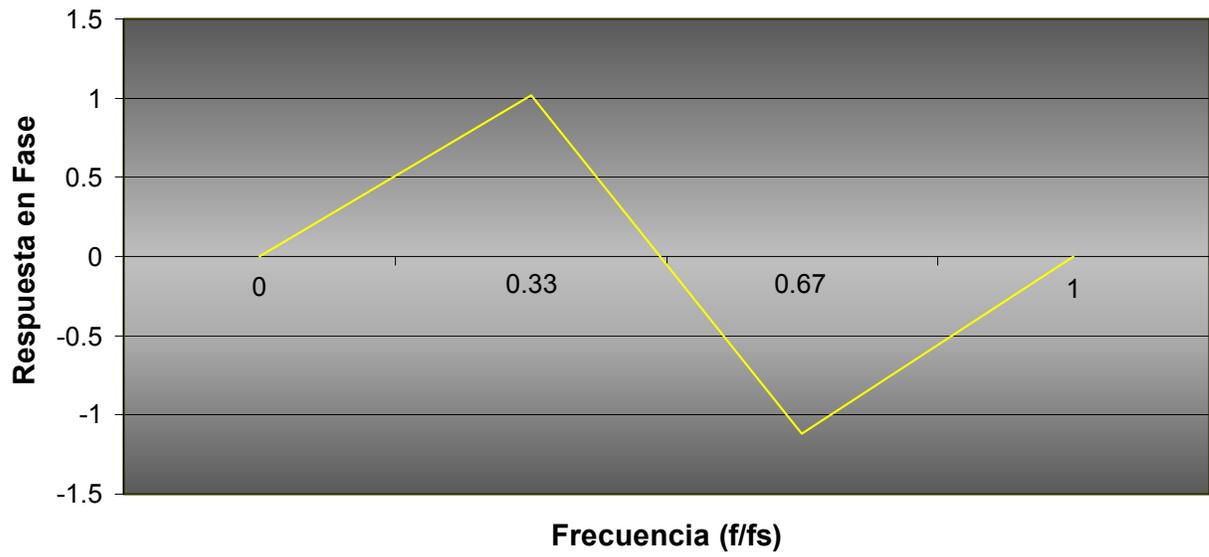


Respuesta en Fase. Filtro Simlet 8 Wavelet.

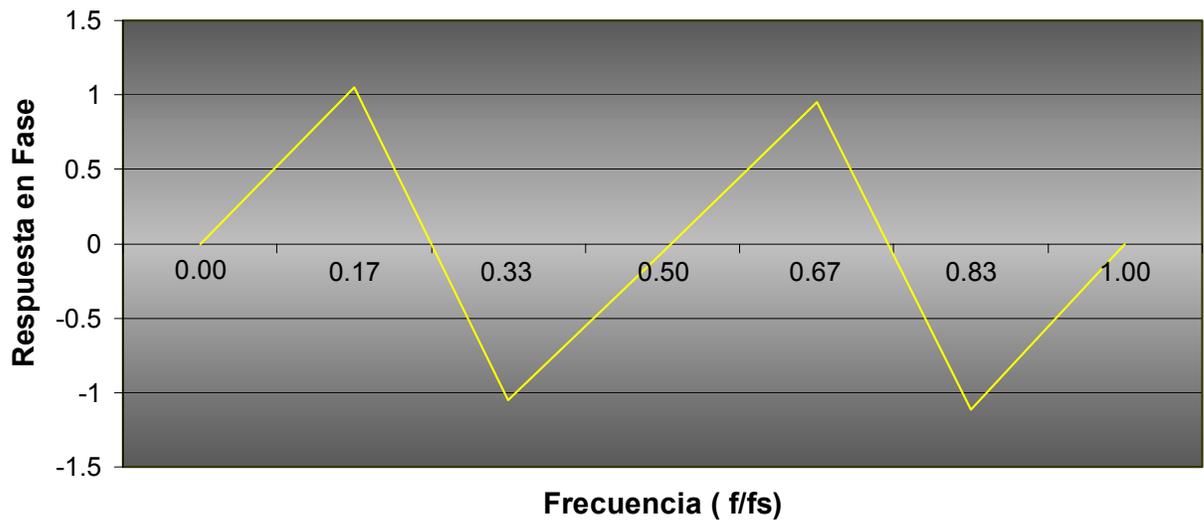


Respuesta en Fase. Filtro Simlet 10 Wavelet.

FAMILIA COIFLETS

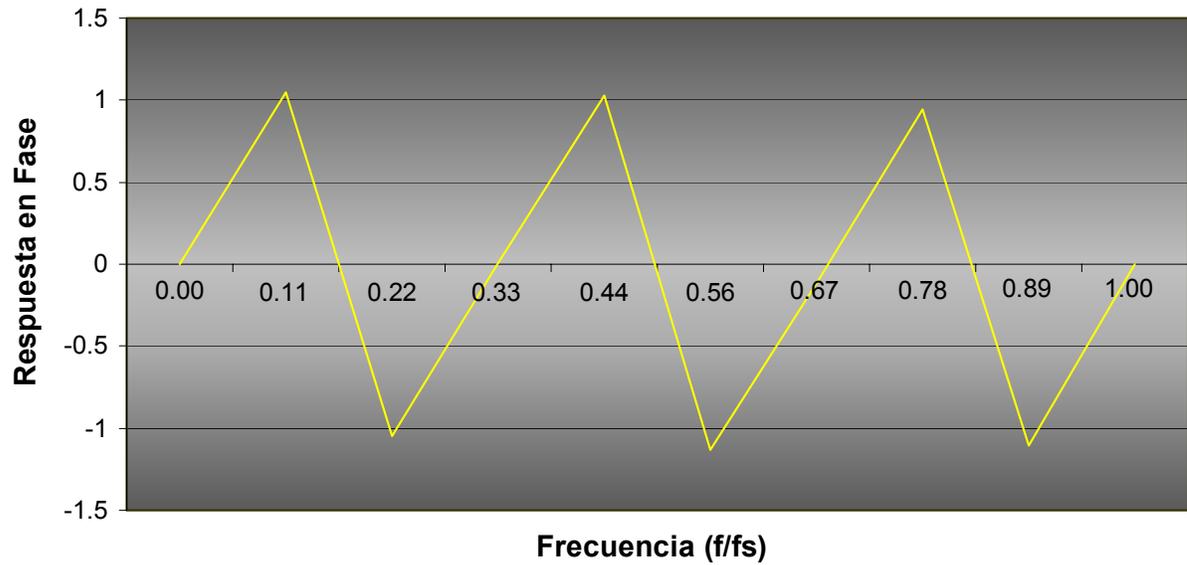


Respuesta en Fase. Filtro Coiflet1 Escala.

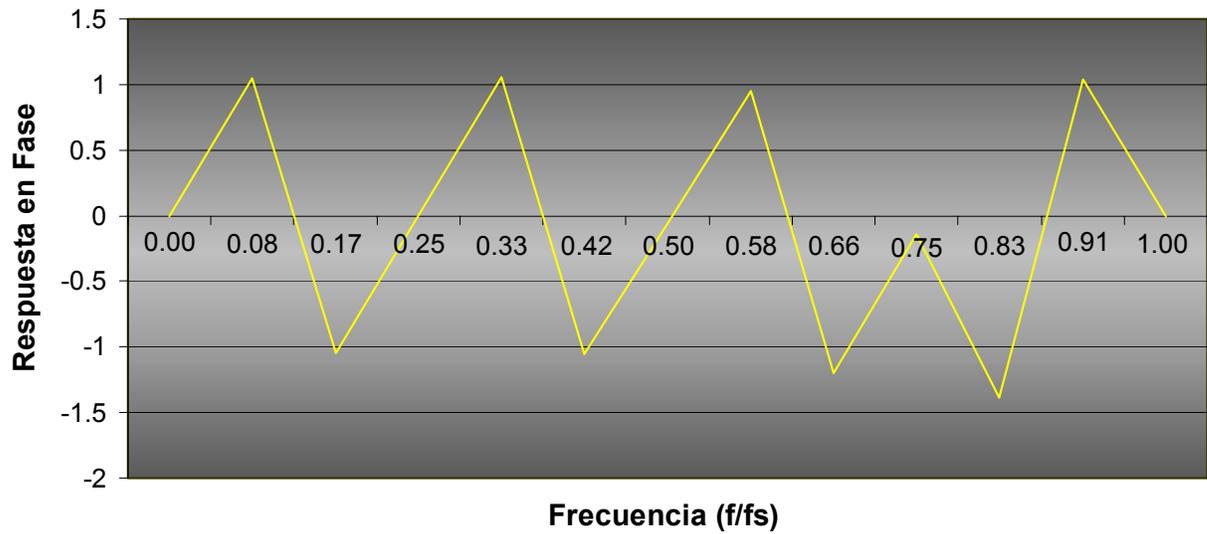


Respuesta en Fase. Filtro Coiflet 2 Escala.

FAMILIA COIFLETS



Respuesta en Fase. Filtro Coiflet 3 Escala.



Respuesta en Fase. Filtro Coiflet 4 Escala.

PROGRAMAS IMPLEMENTADOS EN MATLAB

1.- Versiones trasladadas y escaladas de la función logaritmo

% Familia de funciones de logaritmo.

```
for j=-3:3
for i=-5:5

    t= -50:0.05:50;
    y=log(t*2^j-i);
    plot(t,y,'b');
hold on
end
end
```

2.- Filtros por convolución

```
% contiene ruido de alta frecuencia
N=500;
T=2*pi;
D=T/(N-1);
x=0:D:T;

y=sin(x)+sin(10*x)+sin(50*x)+sin(200*x)+sin(500*x);
h=[0;0;0.3;0.5;0.7;0.9;0.7;0.5;0.3;0;0];
f=conv(y,h);
t=1:length(f);
subplot(2,1,2), plot(t,f);
subplot(2,1,1), plot(x,y);
```

3.- Filtros de Daubechies

```
% hn = daub(N2)
% no usar n mayor que 20

%input : N2=N/2, donde N es largo del filtro
% salida : hn=h(n) coeficientes de largo N con fase minima

N2=15;
a=1; p=1; q=1; % inicialización de variables
hn=[1 1]; % inicialización de factores a -1
for j = 1:N2-1,
    hn= conv(hn, [1,1]);
    a= -a*0.25*(j+N2-1)/j;
    p= conv(p, [1,-2,1]);
```

```

q= [0 q 0]+a*p;
end;

q = sort(roots(q));      %
hn = (conv(hn,real(poly(q(1:N2-1)))));
hn = hn*sqrt(2)/sum(hn);

% wavelet ^*****fin del programa*****
for j = 1: 2*N2,
wn(j)=(-1)^j*hn(2*N2+1-j);
end;
% variable tiempo
t = 0:2*N2-1;
t1=0 : N2;
% espectro
f= fft(hn);
m= abs(f);
g= fft(wn);
m1= abs(g);
end;
% gráficas
subplot(4,1,1),plot(t,hn);
subplot(4,1,2),plot(t,m);
subplot(4,1,3),plot(t,wn);
subplot(4,1,4),plot(t,m1);
end

```

4.- FFT para filtros de Daubechies de orden 4

```

% Ejemplo para D4
D4(1)=0.482963
D4(2)=0.856516
D4(3)=0.224144
D4(4)=-0.12941

z4=fft(D4(1:4));
m4=abs(z4);
r4=real(z4(1:4)); % Entrega la parte real de la FFT
i4=imag(z4(1:4)); % Entrega la parte imaginaria de la FFT
end

```

5.- Aplicación Método de iteración de Newton

```

% filtros de orden 8
clear,fprintf('\n')
da1=0.01; da2=0.01; da3=0.01; da4=0.01; da5=0.01; da6=0.01; da7=0.01; da8=0.01;
a1=input('Estimación inicial de a1:');
a2=input('Estimación inicial de a2:');
a3=input('Estimación inicial de a3:');

```

```
a4=input('Estimación inicial de a4:');
```

```
a5=input('Estimación inicial de a5:');
```

```
a6=input('Estimación inicial de a6:');
```

```
a7=input('Estimación inicial de a7:');
```

```
a8=input('Estimación inicial de a8:');
```

```
for n=1:50
```

```
    s= [a1,a2,a3,a4,a5,a6,a7,a8];
```

```
    a1p=a1+da1;
```

```
    a2p=a2+da2;
```

```
    a3p=a3+da3;
```

```
    a4p=a4+da4;
```

```
    a5p=a5+da5;
```

```
    a6p=a6+da6;
```

```
    a7p=a7+da7;
```

```
    a8p=a8+da8;
```

```
% Jacobianos
```

```
J(1,1) = (f_8_1(a1p,a2,a3,a4,a5,a6,a7,a8) - f_8_1(a1,a2,a3,a4,a5,a6,a7,a8))/da1;
```

```
J(1,2) = (f_8_1(a1,a2p,a3,a4,a5,a6,a7,a8) - f_8_1(a1,a2,a3,a4,a5,a6,a7,a8))/da2;
```

```
J(1,3) = (f_8_1(a1,a2,a3p,a4,a5,a6,a7,a8) - f_8_1(a1,a2,a3,a4,a5,a6,a7,a8))/da3;
```

```
J(1,4) = (f_8_1(a1,a2,a3,a4p,a5,a6,a7,a8) - f_8_1(a1,a2,a3,a4,a5,a6,a7,a8))/da4;
```

```
J(1,5) = (f_8_1(a1,a2,a3,a4,a5p,a6,a7,a8) - f_8_1(a1,a2,a3,a4,a5,a6,a7,a8))/da5;
```

```
J(1,6) = (f_8_1(a1,a2,a3,a4,a5,a6p,a7,a8) - f_8_1(a1,a2,a3,a4,a5,a6,a7,a8))/da6;
```

```
J(1,7) = (f_8_1(a1,a2,a3,a4,a5,a6,a7p,a8) - f_8_1(a1,a2,a3,a4,a5,a6,a7,a8))/da7;
```

```
J(1,8) = (f_8_1(a1,a2,a3,a4,a5,a6,a7,a8p) - f_8_1(a1,a2,a3,a4,a5,a6,a7,a8))/da8;
```

```
J(2,1) = (f_8_2(a1p,a2,a3,a4,a5,a6,a7,a8) - f_8_2(a1,a2,a3,a4,a5,a6,a7,a8))/da1;
```

```
J(2,2) = (f_8_2(a1,a2p,a3,a4,a5,a6,a7,a8) - f_8_2(a1,a2,a3,a4,a5,a6,a7,a8))/da2;
```

```
J(2,3) = (f_8_2(a1,a2,a3p,a4,a5,a6,a7,a8) - f_8_2(a1,a2,a3,a4,a5,a6,a7,a8))/da3;
```

```
J(2,4) = (f_8_2(a1,a2,a3,a4p,a5,a6,a7,a8) - f_8_2(a1,a2,a3,a4,a5,a6,a7,a8))/da4;
```

```
J(2,5) = (f_8_2(a1,a2,a3,a4,a5p,a6,a7,a8) - f_8_2(a1,a2,a3,a4,a5,a6,a7,a8))/da5;
```

```
J(2,6) = (f_8_2(a1,a2,a3,a4,a5,a6p,a7,a8) - f_8_2(a1,a2,a3,a4,a5,a6,a7,a8))/da6;
```

```
J(2,7) = (f_8_2(a1,a2,a3,a4,a5,a6,a7p,a8) - f_8_2(a1,a2,a3,a4,a5,a6,a7,a8))/da7;
```

```
J(2,8) = (f_8_2(a1,a2,a3,a4,a5,a6,a7,a8p) - f_8_2(a1,a2,a3,a4,a5,a6,a7,a8))/da8;
```

```
J(3,1) = (f_8_3(a1p,a2,a3,a4,a5,a6,a7,a8) - f_8_3(a1,a2,a3,a4,a5,a6,a7,a8))/da1;
```

```
J(3,2) = (f_8_3(a1,a2p,a3,a4,a5,a6,a7,a8) - f_8_3(a1,a2,a3,a4,a5,a6,a7,a8))/da2;
```

```
J(3,3) = (f_8_3(a1,a2,a3p,a4,a5,a6,a7,a8) - f_8_3(a1,a2,a3,a4,a5,a6,a7,a8))/da3;
```

```
J(3,4) = (f_8_3(a1,a2,a3,a4p,a5,a6,a7,a8) - f_8_3(a1,a2,a3,a4,a5,a6,a7,a8))/da4;
```

```
J(3,5) = (f_8_3(a1,a2,a3,a4,a5p,a6,a7,a8) - f_8_3(a1,a2,a3,a4,a5,a6,a7,a8))/da5;
```

```
J(3,6) = (f_8_3(a1,a2,a3,a4,a5,a6p,a7,a8) - f_8_3(a1,a2,a3,a4,a5,a6,a7,a8))/da6;
```

```
J(3,7) = (f_8_3(a1,a2,a3,a4,a5,a6,a7p,a8) - f_8_3(a1,a2,a3,a4,a5,a6,a7,a8))/da7;
```

```
J(3,8) = (f_8_3(a1,a2,a3,a4,a5,a6,a7,a8p) - f_8_3(a1,a2,a3,a4,a5,a6,a7,a8))/da8;
```

$$\begin{aligned}
J(4,1) &= (f_{8_4}(a1p,a2,a3,a4,a5,a6,a7,a8) - f_{8_4}(a1,a2,a3,a4,a5,a6,a7,a8))/da1; \\
J(4,2) &= (f_{8_4}(a1,a2p,a3,a4,a5,a6,a7,a8) - f_{8_4}(a1,a2,a3,a4,a5,a6,a7,a8))/da2; \\
J(4,3) &= (f_{8_4}(a1,a2,a3p,a4,a5,a6,a7,a8) - f_{8_4}(a1,a2,a3,a4,a5,a6,a7,a8))/da3; \\
J(4,4) &= (f_{8_4}(a1,a2,a3,a4p,a5,a6,a7,a8) - f_{8_4}(a1,a2,a3,a4,a5,a6,a7,a8))/da4; \\
J(4,5) &= (f_{8_4}(a1,a2,a3,a4,a5p,a6,a7,a8) - f_{8_4}(a1,a2,a3,a4,a5,a6,a7,a8))/da5; \\
J(4,6) &= (f_{8_4}(a1,a2,a3,a4,a5,a6p,a7,a8) - f_{8_4}(a1,a2,a3,a4,a5,a6,a7,a8))/da6; \\
J(4,7) &= (f_{8_4}(a1,a2,a3,a4,a5,a6,a7p,a8) - f_{8_4}(a1,a2,a3,a4,a5,a6,a7,a8))/da7; \\
J(4,8) &= (f_{8_4}(a1,a2,a3,a4,a5,a6,a7,a8p) - f_{8_4}(a1,a2,a3,a4,a5,a6,a7,a8))/da8;
\end{aligned}$$

$$\begin{aligned}
J(5,1) &= (f_{8_5}(a1p,a2,a3,a4,a5,a6,a7,a8) - f_{8_5}(a1,a2,a3,a4,a5,a6,a7,a8))/da1; \\
J(5,2) &= (f_{8_5}(a1,a2p,a3,a4,a5,a6,a7,a8) - f_{8_5}(a1,a2,a3,a4,a5,a6,a7,a8))/da2; \\
J(5,3) &= (f_{8_5}(a1,a2,a3p,a4,a5,a6,a7,a8) - f_{8_5}(a1,a2,a3,a4,a5,a6,a7,a8))/da3; \\
J(5,4) &= (f_{8_5}(a1,a2,a3,a4p,a5,a6,a7,a8) - f_{8_5}(a1,a2,a3,a4,a5,a6,a7,a8))/da4; \\
J(5,5) &= (f_{8_5}(a1,a2,a3,a4,a5p,a6,a7,a8) - f_{8_5}(a1,a2,a3,a4,a5,a6,a7,a8))/da5; \\
J(5,6) &= (f_{8_5}(a1,a2,a3,a4,a5,a6p,a7,a8) - f_{8_5}(a1,a2,a3,a4,a5,a6,a7,a8))/da6; \\
J(5,7) &= (f_{8_5}(a1,a2,a3,a4,a5,a6,a7p,a8) - f_{8_5}(a1,a2,a3,a4,a5,a6,a7,a8))/da7; \\
J(5,8) &= (f_{8_5}(a1,a2,a3,a4,a5,a6,a7,a8p) - f_{8_5}(a1,a2,a3,a4,a5,a6,a7,a8))/da8;
\end{aligned}$$

$$\begin{aligned}
f(1) &= f_{8_1}(a1,a2,a3,a4,a5,a6,a7,a8); \\
f(2) &= f_{8_2}(a1,a2,a3,a4,a5,a6,a7,a8); \\
f(3) &= f_{8_3}(a1,a2,a3,a4,a5,a6,a7,a8); \\
f(4) &= f_{8_4}(a1,a2,a3,a4,a5,a6,a7,a8); \\
f(5) &= f_{8_5}(a1,a2,a3,a4,a5,a6,a7,a8);
\end{aligned}$$

ds=-Jf;

$$\begin{aligned}
a1 &= a1 + ds(1); \\
a2 &= a2 + ds(2); \\
a3 &= a3 + ds(3); \\
a4 &= a4 + ds(4); \\
a5 &= a5 + ds(5); \\
a6 &= a6 + ds(6); \\
a7 &= a7 + ds(7); \\
a8 &= a8 + ds(8);
\end{aligned}$$

$$\begin{aligned}
h(1) &= a1; \\
h(2) &= a2; \\
h(3) &= a3; \\
h(4) &= a4; \\
h(5) &= a5; \\
h(6) &= a6; \\
h(7) &= a7; \\
h(8) &= a8;
\end{aligned}$$

```

fprintf('n=%2.0f, a1=%12.5e, a2=%12.5e' , n,a1,a2)
fprintf('a3=%10.2e, a4=%10.2e\n' ,a3, a4)

if (abs(f(1))<1.0e-9 & abs(f(2))<1.0e-9 & abs(f(2))<1.0e-9 ) , break; end

z=fft(h(1:8));
m=abs(z);

end

```

6.- Descomposición y reconstrucción usando la DWT

```

N=500;
T=2*pi;
D=T/(N-1);
x=0:D:T;
s=sin(x)+sin(10*x)+sin(50*x)+sin(200*x)+sin(500*x);

% designar wavelet
wname = 'db4';

% vector de aproximación y de detalle
[sa, sd] = dwt (s, wname, 'mode', 'ppd');
[sa1,sd1]= dwt (sa,wname, 'mode', 'ppd');
[sa2,sd2]= dwt (sa1,wname, 'mode', 'ppd');

% reconstrucción
f2(1:68)=0
r1 = idwt (sa2,f2 , wname, 'mode', 'ppd');
f3(1:130)=0
r2 = idwt (r1,f3 , wname, 'mode', 'ppd');
f4(1:254)=0
r3 = idwt (r2,f4 , wname, 'mode', 'ppd');
f5(1:502)=0
r4 = idwt (r3,f5 , wname, 'mode', 'ppd');

% gráficas
subplot (211)
plot (s)
subplot (212)
plot (r3(1:500))

```


COEFICIENTES ESCALA SIMLETS

S4	S5	S6	S7	S8
-0,0757657	0,0273331	0,0154041	0,00268181	0,00188995
-0,0296355	0,0295195	0,00349071	-0,00104738	-0,00030292
0,497619	-0,0391342	-0,11799	-0,0126363	-0,0149523
0,803739	0,199398	-0,0483117	0,0305155	0,00380875
0,297858	0,723408	0,491055	0,0678927	0,0491372
-0,0992195	0,633979	0,787641	-0,0495528	-0,027219
-0,012604	0,0166021	0,337929	0,0174413	-0,0519458
0,0322231	-0,175328	-0,0726375	0,536102	0,364442
	-0,0211018	-0,0210603	0,767764	0,777186
	0,0195389	0,0447249	0,28863	0,48136
		0,00176771	-0,140047	-0,0612734
		-0,00780071	-0,107808	-0,143294
			0,00401024	0,00760749
			0,0102682	0,0316951
				-0,00054213
				-0,00338242
S10				
0,00077016				
9,56E-05				
-0,00864133				
-0,00146538				
0,0459272				
0,0116099				
-0,159494				
-0,0708805				
0,471691				
0,76951				
0,383827				
-0,0355367				
-0,0319901				
0,049995				
0,00576491				
-0,0203549				
-0,00080435				
0,00459317				
5,70E-05				
-0,00045932				

COEFICIENTES ESCALA COIFLETS

C1	C2	C3	C4	C5
-0,0727326	0,0163873	-0,00379351	0,00089231	-0,00021208
0,337898	-0,0414649	0,0077826	-0,00162949	0,00035859
0,852572	-0,0673726	0,0234527	-0,00734617	0,00217824
0,384865	0,38611	-0,0657719	0,0160689	-0,00415936
-0,072733	0,812724	-0,0611234	0,0266823	-0,0101311
-0,0156557	0,417005	0,405177	-0,0812667	0,0234082
	-0,0764886	0,793777	-0,0560773	0,028168
	-0,0594344	0,428483	0,415308	-0,09192
	0,0236802	-0,0717998	0,782239	-0,0520432
	0,00561143	-0,0823019	0,434386	0,421566
	-0,00182321	0,034555	-0,0666275	0,77429
	-0,00072055	0,0158805	-0,0962204	0,437992
		-0,00900798	0,0393344	-0,062036
		-0,00257452	0,0250823	-0,105574
		0,00111752	-0,0152117	0,0412892
		0,00046622	-0,00565829	0,0326836
		-7,10E-05	0,00037514	-0,0197618
		-3,46E-05	0,00126656	-0,00916423
			-0,00058902	0,00676419
			-0,00025998	0,00243337
			6,23E-05	-0,00166286
			3,12E-05	-0,00063813
			-3,26E-06	0,00030226
			-1,78E-06	0,00014054
				-4,13E-05
				-2,13E-05
				3,73E-06
				2,06E-06
				-1,67E-07
				-9,52E-08