

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA
ESCUELA DE ELECTRÓNICA



PLATAFORMA DE SIMULACIÓN PARA LA REDUCCIÓN DE RUIDO
EN SEÑALES ELÉCTRICAS HACIENDO USO DE LA TEORÍA DE
WAVELETS

TRABAJO DE GRADUACIÓN

PRESENTADO POR:

Francisco José Robles Andrade

PARA OPTAR AL GRADO DE

Ingeniero en Electrónica

Asesor:

Ing. Juan Carlos Castro Chávez

Septiembre de 2006

Soyapango – El Salvador – Centro América

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA
ESCUELA DE ELECTRÓNICA

AUTORIDADES:

RECTOR
ING. FEDERICO HUGUET RIVERA

VICERRECTOR ACADÉMICO
PBRO. VÍCTOR BERMÚDEZ, sdb

SECRETARIO GENERAL
LIC. MARIO RAFAEL OLMOS

DECANO DE LA FACULTAD DE INGENIERÍA
ING. GODOFREDO GIRÓN

DIRECTOR DE ESCUELA DE ELECTRÓNICA
ING. OSCAR DURÁN VIZCARRA

ASESOR DEL TRABAJO DE GRADUACIÓN
ING. JUAN CARLOS CASTRO CHÁVEZ

JURADO EVALUADOR
ING. OSCAR WENCESLAO RIVAS ZALDAÑA
ING. EDUARDO RIVERA
ING. JUAN CARLOS CRUZ DADA

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA
ESCUELA DE ELECTRÓNICA

JURADO EVALUADOR DEL TRABAJO DE GRADUACIÓN

Ing. Oscar Wenceslao Rivas
JURADO

Ing. Eduardo Rivera
JURADO

Ing. Juan Carlos Cruz Dada
JURADO

Ing. Juan Carlos Castro Chávez
ASESOR

Agradecimientos

Agradezco en primer lugar a Dios Todopoderoso por todas las bendiciones recibidas. Por darme la fuerza y determinación necesarias para dar por finalizado una etapa de gran importancia en mi vida. Por colocar grandes y maravillosas personas a mí alrededor de las cuales he aprendido muchísimas cosas y de las cuales he recibido un apoyo incondicional.

A la Virgen María Auxiliadora por guiar nuestros caminos como la verdadera Madre que es y por confortarnos en los momentos más difíciles de nuestras vidas. Porque siempre nos ha acompañado y escuchado nuestras oraciones.

A mi Madre, Dinora Elizabeth Andrade, por ser el ángel que Dios puso en la vida de mi hermana y la mía para guiarnos y darnos la fortaleza necesaria para afrontar con coraje y amor todo lo que la vida nos tiene preparado. Por enseñarnos, con su ejemplo de vida, a ser personas responsables y luchar por lo que queremos. Por que con su amor, ternura, cariño y sacrificio, me enseñó cosas que jamás se pueden aprender de otra manera. Por tus noches de desvelo y por tu confianza en mí, GRACIAS MAMI.

A mi papá, Pedro José Robles, por creer en mis posibilidades de salir adelante y enseñarme a ser independiente. Por su apoyo y respeto a todas las decisiones, actividades y retos que a lo largo de mi vida he tomado y adquirido. Por estar a nuestro lado en los momentos importantes y trascendentales de nuestras vidas.

A Dinora Guadalupe Robles Andrade (Lupita), mi hermanita. Porque con el correr del tiempo ha sido un ejemplo de valentía y fuerza para mí. Hasta el punto de llegar a convertirse en una mujer responsable, independiente, muy inteligente y capaz de salir adelante. Por ser mi alegría y orgullo y permitirme disfrutar sus triunfos. Por el amor que me prodiga, por su confianza en mí, por respetar y apoyar mis decisiones, por hacerme reaccionar cuando cometo un error, por su invaluable compañía y por ser mi hermana, gracias.

A mi abuelito, Manuel Antonio Guerra (Q.D.D.G), que si bien no esta físicamente con nosotros para vernos coronar uno de sus sueños, sé que esta allá arriba en el cielo muy feliz y orgulloso. Lo sé porque desde la fecha en que partió a la presencia de Dios, he sentido que gane un ángel de la guarda más en mi vida y su espíritu jamás ha dejado de estar con nosotros para apoyarnos y darnos el aliento necesario. Gracias por enseñarme lo que es tener carácter y afrontar todas las adversidades de frente, por enseñarme la importancia de mantener una familia unida y luchar con mucho amor y firmeza porque eso se mantenga de esa manera, por cuidarnos y apoyarnos tanto tiempo. En especial esto te lo dedico a ti papá Meme, ¡lo logramos!

A mis tíos y tías. Manuel y Yanira, Vilma y Julio, Oscar y Cecilia, Rosario y Mauricio. Por ser como verdaderos padres y madres para mi hermana y para mí, en diferentes facetas de nuestras vidas, por ser ejemplos de abnegación y entrega, por todo su cariño y amor demostrados, por ser mis grandes ejemplos de trabajo duro, por confiar y apoyarme siempre.

A mis primos y primas. Stephany, Alejandro, Erick, Walter, Javier, Angélica, Jaqueline, Oscar, José, Vilma, Moisés, Oti y Walter Orlando. Por ser mis otros hermanos y por hacerme la vida feliz. A los menores, por demostrarme y recordarme que siempre la esperanza de un mundo mejor son ustedes y enseñarme a mantenerme siempre alegre. A los mayores, por permitirme compartir tiempo con ustedes y por sus consejos, ayuda, amistad, confianza y cariño.

A mi abuela, María Lidia Cortez (Q.D.D.G). Por sus consejos y atenciones. Porque con su forma de ser me enseñó la fuerza de carácter con la que se deben afrontar las diferentes facetas de la vida.

A Claudia Beatriz Molina Morán, mi novia. Por ser la pieza que le da el equilibrio necesario a mi vida. Por permitirme encontrarla en el lugar y momento justos. Por demostrarme con su amor incondicional, confianza, respeto y apoyo, que es la mujer con quien quiero compartir el resto de los días que Dios me regale vida. Por ser además esa

amiga que siempre esta dispuesta a escucharme. A su familia, por abrir las puertas de su hogar y acogerme como un miembro más de ella, brindándome mucha confianza, apoyo y cariño.

A mi gran amigo de toda la vida, Manuel José Merino. Por ser un hermano mayor para mí, por sus consejos, por su apoyo, por su infinita confianza, por su respeto, por su amistad. A su familia por abrir las puertas de su hogar y en especial a su hermana Patricia Merino por su valioso aporte a la conclusión del presente Trabajo de Graduación.

A mis amigos y excelentes compañeros de estudios: Carlos Rodríguez, Juan Carlos Rivera, Francis Rodríguez y Francisco Reyes. Porque con el pasar de los años y la distancia siempre mantenemos el contacto, por todos los días y noches de estudios que compartimos en el Colegio y la Universidad, por todos los logros que disfrutamos juntos, por todas las aventuras que corrimos, por enseñarme e impulsarme a siempre seguir adelante por difíciles que fueran las situaciones, por permitirme absorber algunos de sus valiosos conocimientos y enseñarme a trabajar en equipo, por respetar mis ideas y enseñarme a escuchar puntos de vista distintos, por estar siempre a mi lado. A cada una de sus familias por alojarme en muchísimas ocasiones durante nuestras tertulias de estudio, ya sea para preparar un examen o trabajar en algún proyecto y por el apoyo y confianza que cada uno de ellos nos demostraron al abrirnos las puertas de sus hogares.

Un agradecimiento especial vaya a los profesores e instructores de la Universidad Don Bosco, quienes me permitieron aprender muchas cosas durante mis años de estudio, tanto del tipo académico como del tipo humano y profesional. Gracias por sus consejos y permitirme robarles una pequeña porción de sus conocimientos, los cuales intento e intentaré poner siempre en práctica en mi vida profesional.

Francisco José Robles Andrade

Objetivos Generales

- Introducir a los interesados en el procesamiento de señales e imágenes a una herramienta matemática de gran utilidad y muy poco conocida como es el caso de la Teoría de Wavelets.
- Desarrollar una aplicación que sea capaz de reducir el nivel de ruido eléctrico en señales que se encuentre basada en la Teoría de Wavelets.

Objetivos Específicos

- Respaldo de manera conceptual y matemática la Teoría de Wavelets, presentando todos los principios y parámetros que la rigen.
- Demostrar el por qué para ciertos tipos de aplicaciones la Transformada de Wavelets presenta un mejor desempeño que la Transformada de Fourier.
- Describir la utilización de la Teoría de Wavelets en el proceso de reducción de ruido sobre señales generadas artificialmente a través de software.
- Presentar tres algoritmos que se fundamentan en la Teoría de Wavelets para llevar a cabo el proceso de reducción de ruido en señales.
- Desarrollar las rutinas necesarias en MATLAB® que sean capaces de simular el comportamiento de los diferentes algoritmos de reducción de ruido basados en Wavelets.
- Respaldo a través de gráficos y tablas de datos, el análisis comparativo de los diferentes algoritmos de reducción de ruido que se implementarán en MATLAB®.

Objetivos Generales y Específicos

- Describir las bases para el diseño de los diversos algoritmos que son necesarios en la implementación computacional de la aplicación en reducción de ruido de la Teoría de Wavelets.
- Diseñar y desarrollar una aplicación amigable al usuario que sea capaz de simular el comportamiento de los Wavelets en el proceso de reducción de ruido y que se encuentre basada en Visual Basic 6.0 como lenguaje de programación.

Índice de Tablas

| | | |
|-----|-------------------------------------------------------------------------|------------|
| 3.1 | Variantes del decibel y su uso | 76 |
| 3.2 | Delimitación de variables para simulación en MATLAB® | 91 |
| 3.3 | Ejemplo de valores de MSE y Umbral: Minimax, Umbral suave, Daubechies 5 | 96 |
| 3.4 | Mejores estimaciones Wavelet-Método aplicando el Umbral Duro | 99 |
| 3.5 | Mejores estimaciones Wavelet-Método aplicando el Umbral Suave | 99 |
| 4.1 | Coeficientes escala y wavelet para Familia Daubechies 5 | 117 |
| 4.2 | Coeficientes escala y wavelet para Familia Coiflets 3 | 118 |

Índice de Figuras

| | <i>Figura</i> | <i>Pág.</i> |
|------|---------------------------------------------------------------------------------------------|-------------|
| 1.1 | (a) Señal original; (b) Descomposición en Series de Fourier | 15 |
| 1.2 | Señales senoidales de 5 y 10 Hz | 16 |
| 1.3 | Señal de 50Hz y su Transformada de Fourier | 17 |
| 1.4 | Señal estacionaria y su espectro de frecuencia | 21 |
| 1.5 | Señal no estacionaria y su espectro de frecuencia | 22 |
| 1.6 | Representación gráfica de la STFT | 26 |
| 1.7 | STFT de la señal no estacionaria de la Fig. 1.5 con ventana Gaussiana $a= 180$ | 27 |
| 1.8 | Enrejado resultante de la STFT en el plano tiempo-frecuencia | 28 |
| 1.9 | Representación de señal cuadrada y su Transformada Discreta de Fourier | 30 |
| | | |
| 2.1 | Traza sísmica típica | 32 |
| 2.2 | Discretización de una señal continua | 40 |
| 2.3 | Gráfica de la función escala Haar | 43 |
| 2.4 | Señal de voltaje de línea con sobre impulsos | 43 |
| 2.5 | Aproximación de la señal de voltaje de línea con sobre impulsos, utilizando la Wavelet Haar | 44 |
| 2.6 | Gráfica de un elemento típico en V_0 | 45 |
| 2.7 | Representación gráfica de $\phi(x-j)$ y $\phi(x-k)$ | 48 |
| 2.8 | Representación gráfica de la función Wavelet Haar $\psi(x)$ | 49 |
| 2.9 | Interpretación gráfica de la resolución en el tiempo y la frecuencia de la WT | 54 |
| 2.10 | Proceso de escalamiento y traslación de una señal | 56 |
| 2.11 | Procedimiento de cálculo de la CWT | 58 |
| 2.12 | Algoritmo de codificación sub-banda | 61 |
| 2.13 | Representación gráfica Wavelet Haar | 64 |
| 2.14 | Representación gráfica de Wavelet del Sombrero Mejicano | 65 |
| 2.15 | Representación gráfica de Wavelet Daubechies 5 | 66 |
| 2.16 | Representación gráfica de Wavelet Symlets 2 | 67 |

| | <i>Figura</i> | <i>Pág.</i> |
|------|--------------------------------------------------------------------------------------------------------------------|-------------|
| 2.17 | Representación gráfica de Wavelet Coiflet 1 | 68 |
| 2.18 | Representación gráfica de Wavelet Morlet | 69 |
| 2.19 | Representación gráfica de Wavelet Gaussiana de orden 1 | 70 |
| 3.1 | (a) Sinusoidal pura, (b) Sinusoidal con ruido aditivo | 72 |
| 3.2 | Clasificación del ruido eléctrico | 73 |
| 3.3 | Modelo de Ruido térmico, blanco o Gaussiano | 74 |
| 3.4 | Esquema de proceso de reducción de ruido con wavelets | 78 |
| 3.5 | Comportamiento de métodos no lineales de reducción de ruido. (a) Señal original, (b) Umbral duro, (c) Umbral suave | 80 |
| 3.6 | Señales sintéticas de prueba con $n = 4096$ muestras | 90 |
| 3.7 | Señales sintéticas de prueba con ruido, $n = 4096$ y $SNR = 3$ | 95 |
| 3.8 | Señales de prueba recuperadas: Minimax, Umbral suave, Daubechies 5 | 97 |
| 3.9 | Representación gráfica de las mejores estimaciones – Umbral duro | 100 |
| 3.10 | Representación gráfica de las mejores estimaciones – Umbral suave | 101 |
| 4.1 | Proceso de Descomposición Wavelet. Banco de Filtro de dos canales | 105 |
| 4.2 | Proceso de Reconstrucción Wavelet. Banco de Filtro de dos canales | 105 |
| 4.3 | Banco de filtros de 5 niveles de descomposición | 107 |
| 4.4 | Ventana de bienvenida e informativa de aplicación desarrollada | 108 |
| 4.5 | Interfase de configuración principal | 109 |
| 4.6 | Resultados obtenidos con aplicación diseñada en VB 6.0 para la limpieza/reducción de ruido con Wavelets | 119 |

Índice General

| <i>Contenido</i> | <i>Página</i> |
|--------------------------------------------------------------------------------|---------------|
| Introducción | 10 |
| Capítulo 1: Fundamentos del análisis de señales | 12 |
| 1.1 Introducción | 12 |
| 1.2 La Transformada de Fourier | 13 |
| 1.2.1 Propiedades de la Transformada de Fourier | 18 |
| 1.2.2 Limitaciones de la Transformada de Fourier | 19 |
| 1.3 El Principio de Indeterminación de Heisenberg y el plano tiempo-frecuencia | 22 |
| 1.4 La Transformada de Fourier de Período Corto | 24 |
| 1.5 La Transformada Discreta de Fourier | 29 |
| Capítulo 2: Teoría de Wavelets | 31 |
| 2.1 Introducción | 31 |
| 2.2 Antecedentes Históricos de la teoría de Wavelets | 32 |
| 2.3 Principios matemáticos que rigen la teoría de Wavelets | 36 |
| 2.3.1 Espacios con Producto Interno | 36 |
| 2.3.2 El espacio cuadrado integrable L^2 | 39 |
| 2.4 La Wavelet haar | 43 |
| 2.4.1 Función Haar escala | 45 |
| 2.4.2 Función Haar Wavelet | 48 |
| 2.5 Análisis Multi-resolución | 50 |
| 2.6 La Transformada Wavelet | 52 |
| 2.6.1 Variables de escala (a) y traslación (b) | 54 |
| 2.6.2 Transformada Wavelet Continua | 57 |
| 2.6.3 Transformada Wavelet Discreta | 60 |

| <i>Contenido</i> | <i>Página</i> |
|-----------------------------------------------------------------------------------|---------------|
| 2.7 Familias de Wavelets | 63 |
| 2.7.1 Wavelet Haar | 63 |
| 2.7.2 Función del Sombrero Mejicano | 64 |
| 2.7.3 Wavelet Daubechies | 65 |
| 2.7.4 Wavelet Symlets | 66 |
| 2.7.5 Wavelet Coiflet | 67 |
| 2.7.6 Wavelet Morlet | 68 |
| 2.7.7 Wavelet Gaussiana | 69 |
| Capítulo 3: Reducción de ruido en señales utilizando Wavelets | 71 |
| 3.1 Introducción | 71 |
| 3.2 Definición de ruido eléctrico | 72 |
| 3.3 Procedimiento general de reducción de ruido | 76 |
| 3.3.1 Criterios de umbral | 78 |
| 3.4 Algoritmos de reducción de ruido basados en la teoría de Wavelets | 80 |
| 3.4.1 Algoritmo Minimax | 82 |
| 3.4.2 Algoritmo Fixed form Threshold | 84 |
| 3.4.3 Algoritmo Rigorous SURE | 85 |
| 3.5 Simulación de algoritmos para reducción de ruido | 86 |
| 3.5.1 Definición de señales de prueba | 86 |
| 3.5.2 Definición de parámetros de simulaciones | 90 |
| 3.5.3 Simulación de algoritmos utilizando MATLAB | 92 |
| 3.5.4 Análisis de resultados y comparación entre algoritmos | 97 |
| Capítulo 4: Desarrollo de aplicación d Wavelets basada en Visual Basic 6.0 | 102 |
| 4.1 Introducción | 102 |
| 4.2 Transformada Rápida Wavelet y banco d filtros | 103 |
| 4.2.1 Múltiples niveles de descomposición - reconstrucción | 106 |
| 4.3 Descripción de aplicación desarrollada en Visual Basic 6.0 | 108 |
| 4.3.1 Generación de señales de prueba utilizando Visual Basic 6.0 | 109 |
| 4.3.2 Generación de señal interferente | 111 |
| 4.3.3 Selección de la Familia Wavelet a utilizar en el análisis | 112 |

| <i>Contenido</i> | <i>Página</i> |
|----------------------------------------------------------------------------|---------------|
| 4.3.4 Algoritmo de reducción de ruido implementado | 118 |
| 4.3.5 Visualización de Resultados | 118 |
| Conclusiones | 121 |
| Recomendaciones | 124 |
| Apéndice A: Programas implementados en MATLAB® | 125 |
| Apéndice B: Resultados completos de simulaciones utilizando MATLAB® | 132 |
| Referencias Bibliográficas | 157 |

Introducción

A raíz de la acelerada expansión de las tecnologías de información y sistemas de comunicación de hoy en día, surge la necesidad de conocer herramientas que nos garanticen obtener la información que estos sistemas procesan con la mayor certeza de que dicha información sea recibida libre de errores.

Debido a que, sin importar el medio de transmisión a través del cual se propaguen los mensajes (señales), estos siempre se ven afectados por perturbaciones eléctricas de naturaleza interna y externa. Fenómenos que al final son los responsables de la corrupción de los datos que son recibidos. Teniendo esto en mente es importante desarrollar sistemas que busquen el disminuir el impacto de dichas perturbaciones.

Es por ello que en el presente estudio se presentará una teoría alternativa y poco conocida (si es comparada con el Análisis de Fourier) en el desarrollo de filtros para reducción de ruido en señales eléctricas. La teoría a la cual nos referimos es la Teoría de Wavelets, la cual como se demostrará en los siguientes capítulos corrige algunas limitantes con las que cuenta la Teoría de Fourier.

Así es como en el Capítulo 1 nos introducimos de forma muy general al análisis de señales definiendo la tan ampliamente utilizada Teoría de Fourier, en el cual se definen sus propiedades, aplicaciones y limitantes. Limitantes que radican primordialmente en la incapacidad de la Teoría de Fourier ante las representaciones tiempo-frecuencia debido al principio de indeterminación de Heisenberg. Se estudia además la Transformada de Gabor o Transformada de Fourier de Período Corto; la cual corrige parcialmente las deficiencias de la Transformada de Fourier clásica.

Luego en el Capítulo 2, se introduce al lector en el mundo de la teoría de Wavelets, desde los antecedentes históricos que dieron origen a dicha teoría hasta los fundamentos matemáticos que la respaldan son expuestos en este capítulo. El análisis Multi-resolución, base sobre la cual se fundamenta el estudio de Wavelets, debido a la capacidad del mismo

de obtener muy buenas representaciones tiempo-frecuencia y sin contradecir el principio de indeterminación de Heisenberg, también se estudia aquí.

Luego, en el Capítulo 3 se lleva a la práctica una de las principales aplicaciones de los Wavelets, la reducción de ruido, definiendo algoritmos que se encargan de llevar a cabo con muy buenos resultados esta tarea, tal y como se podrá observar en el análisis de resultados y en el apéndice B. Se desarrollan además los algoritmos necesarios, haciendo uso de MATLAB®, para conseguir la reducción del ruido en las señales de prueba que son generadas dentro de la misma aplicación, dichos algoritmos se encuentran plasmados por completo en el apéndice A.

Luego de comparar los resultados de desempeño de los diferentes algoritmos de reducción de ruido basados en Wavelets, se seleccionó aquel que presentó los mejores resultados, el cual es implementado en una aplicación desarrollada en Visual Basic 6.0 en la forma de un sistema que simula el comportamiento de la Teoría de Wavelets para la reducción de ruido eléctrico en señales. Esto es expuesto en el Capítulo 4. En este mismo capítulo se describe la plataforma de simulación y se demuestra su comportamiento.

Finalmente, se presentan las conclusiones, que según el criterio del autor validan la importancia de prestar atención a la Teoría de Wavelets y desarrollar una investigación a este respecto. Así como también se presentan un par de recomendaciones para trabajo futuro y seguir de esta manera encontrando nuevas aplicaciones de esta potente herramienta matemática.

Capítulo I

Fundamentos del análisis de señales

1.1 Introducción

El presente capítulo tiene el objetivo primordial de presentar una visión generalizada de los conceptos básicos en el análisis de señales y en particular de aquellos que servirán como antecedentes para fundamentar la teoría de Wavelets, la cual es la columna vertebral del presente documento.

En un primer momento se define y establece la teoría de la Transformada de Fourier, que aun siendo la transformada de más amplio uso en el procesamiento de señales, se demostrará que para casos muy puntuales no es la técnica más adecuada, sin que ello reste la importancia y potencia de la misma.

Con la finalidad de corregir las desventajas o deficiencias que presenta la Transformada de Fourier en el análisis de señales no estacionarias, se desarrolla la Transformada Rápida de Fourier (STFT) ó Transformada de Gabor, de la cual también se hará un breve repaso en este capítulo, y se demostrará su empleo en las representaciones tiempo-frecuencia de señales no estacionarias. La total comprensión de la STFT por parte del lector es de vital importancia para la posterior comprensión de la Transformada Wavelet, ya que esta última fue desarrollada como una alternativa para resolver algunos problemas que presenta la STFT.

1.2 La Transformada de Fourier

Un proceso físico puede ser descrito en el dominio del tiempo mediante valores representados por una cantidad como función del tiempo. También es posible describir el mismo proceso en el dominio de la frecuencia mediante una serie de amplitudes representadas como función de la frecuencia. La transformada de Fourier como veremos a continuación es una herramienta con la capacidad de representar este tipo de procesos, o cualquier otro, tanto en el dominio del tiempo como en el dominio de la frecuencia. Esto hace que la Transformada de Fourier sea ampliamente utilizada en aplicaciones en el campo de la ciencia e ingeniería.

En esencia, la Transformada de Fourier descompone o expande una señal en senos y cosenos de diferentes frecuencias, cuya suma corresponde a la señal original, es decir, es capaz de distinguir las diferentes componentes de frecuencia de la señal, y sus respectivas amplitudes. Ahora bien, desde el punto de vista matemático, la transformada de Fourier de una función del tiempo $f(t)$ queda definida por la ecuación 1.1:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{j\omega t} dt \quad (1.1)$$

y la transformada inversa de Fourier definida por la ecuación 1.2:

$$f(t) = \int_{-\infty}^{\infty} F(\omega)e^{-j\omega t} d\omega \quad (1.2)$$

En la ecuación 1.1 se observa que la señal de interés $f(t)$, es multiplicada por un término sinusoidal de frecuencia $\omega = 2\pi f$. De lo cual se concluye que si la señal tiene una alta componente de frecuencia f entonces el producto de la señal y del término sinusoidal es relativamente grande, lo cual indica que la señal $f(t)$ presenta una fuerte componente de frecuencia f . Sin embargo, si la señal no tiene una componente de frecuencia f dicho producto tiende a cero.

Además, vale la pena hacer notar que la información proporcionada por la integral corresponde a todos los instantes de tiempo, ya que el intervalo de integración esta comprendido entre $-\infty$ y $+\infty$, lo cual implica que no importa el instante de tiempo en el que aparece la componente de frecuencia f porque no afectará el resultado de la integración. Por lo tanto, la Transformada de Fourier solamente es capaz de entregar información de la existencia o no de ciertas componentes de frecuencia.

De acuerdo a lo que se puede observar a partir de la pareja de ecuaciones de la transformada de Fourier (ecuaciones 1.1 y 1.2), con ellas se puede obtener una representación en el dominio de la frecuencia de una señal que se encuentra originalmente en el dominio del tiempo. La relación existente entre la representación de la señal original a través de funciones sinusoidales y la exponencial que se expresan en las ecuaciones (1.1) y (1.2) proviene de la definición de la identidad de Euler mostrada en la ecuación 1.3..

$$\begin{aligned}e^{j\omega t} &= \text{Cos}(\omega t) + j\text{Sen}(\omega t) \\e^{-j\omega t} &= \text{Cos}(\omega t) - j\text{Sen}(\omega t)\end{aligned}\tag{1.3}$$

A partir de esta función exponencial es posible formular un conjunto de funciones ortogonales del tipo:

$$\{e^{jn\omega t} : n = 0, \pm 1, \pm 2, \dots\}$$

Sobre un intervalo $\{t_0, t_0+T\}$, y por lo tanto podemos descomponer o expandir la señal original (en el dominio del tiempo) según se detalla en las ecuaciones 1.4 y 1.5.

$$f(t) = F_0 + F_1 e^{-j\omega t} + F_2 e^{-j2\omega t} + \dots + F_{-1} e^{j\omega t} + F_{-2} e^{j2\omega t} + \dots\tag{1.4}$$

$$f(t) = \sum_{-\infty}^{\infty} F_n e^{-jn\omega t}\tag{1.5}$$

Dichas funciones exponenciales pueden ser referenciadas como las funciones base de la Transformada de Fourier, y debido a su propiedad de ortogonalidad, es posible obtener los coeficientes F_n como términos de semejanza entre la función original y la función exponencial, tal y como se detalla en las ecuaciones 1.6 y 1.7 a continuación:

$$F_n = \frac{\int_{t_0}^{t_0+T} f(t)e^{-jn\omega t} dt}{\int_{t_0}^{t_0+T} e^{jn\omega t} e^{-jn\omega t} dt} \quad (1.6)$$

$$F_n = \frac{1}{T} \int_{t_0}^{t_0+T} f(t)e^{-jn\omega t} dt \quad (1.7)$$

Aunque matemáticamente la función exponencial resulta más fácil de manipular, generalmente se suele trabajar con las funciones seno y coseno, ya que desde un punto de vista físico, resulta más fácil comprender el paso de la señal del dominio del tiempo al dominio de la frecuencia y en forma inversa. Por lo tanto, es posible realizar una transformación de la ecuación (1.5) a la forma expresada en la ecuación 1.8.

$$f(t) = \frac{a_0}{2} + \sum \{a_n \text{Cos}(n\omega t) + b_n \text{Sen}(n\omega t)\} \quad (1.8)$$

$$\begin{aligned} \therefore \\ a_n &= \text{Re}[F_n] \\ b_n &= \text{Im}[F_n] \end{aligned}$$

A partir de la ecuación 1.8 se puede decir que la función en el dominio del tiempo ha sido representada como una combinación lineal de todas las componentes de frecuencia presentes en la señal $f(t)$, donde los coeficientes a_n y b_n representan la cantidad de energía que aporta cada componente de frecuencia a la señal original, tal y como se puede observar en la figura 1.1.

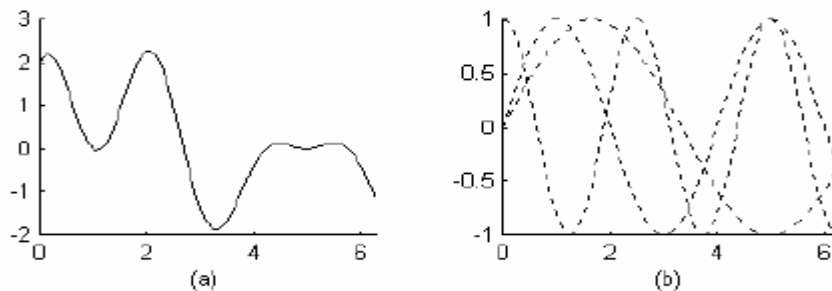


Fig. 1.1: (a) Señal original; (b) Descomposición en series de Fourier

Capítulo 1. Fundamentos del Análisis de Señales

Muchos de los fenómenos físicos pueden describirse mediante una señal en el dominio del tiempo, es decir, una de las variables es el tiempo (variable independiente) y la otra la amplitud (variable dependiente). Cuando se representa gráficamente dicha señal se obtiene una función tiempo-amplitud. Sin embargo, la información que se puede obtener directamente de esta representación no siempre es la más apropiada, puesto que la información que caracteriza a la señal, en muchos casos, puede observarse más claramente en el dominio de la frecuencia, es decir, mediante un espectro que muestre las frecuencias existentes en la señal. Por lo tanto, para una mejor representación de la señal se hace necesario disponer de su representación en el dominio del tiempo y de la frecuencia de forma simultánea.

En la figura 1.2 se pueden observar dos señales en el dominio del tiempo. Para determinar el espectro de frecuencia de cada una de estas señales, se puede hacer uso de la Transformada de Fourier, tal y como se mencionó anteriormente dicha transformada parte de una representación en el dominio del tiempo de la señal y obtiene la representación en frecuencias de la misma; si se representara esto gráficamente, en un eje se mostraría la frecuencia y en el otro la amplitud. Así pues, las señales que se observan en la figura 1.2 presentan una sola componente de frecuencia, 5 Hz y 10 Hz respectivamente.

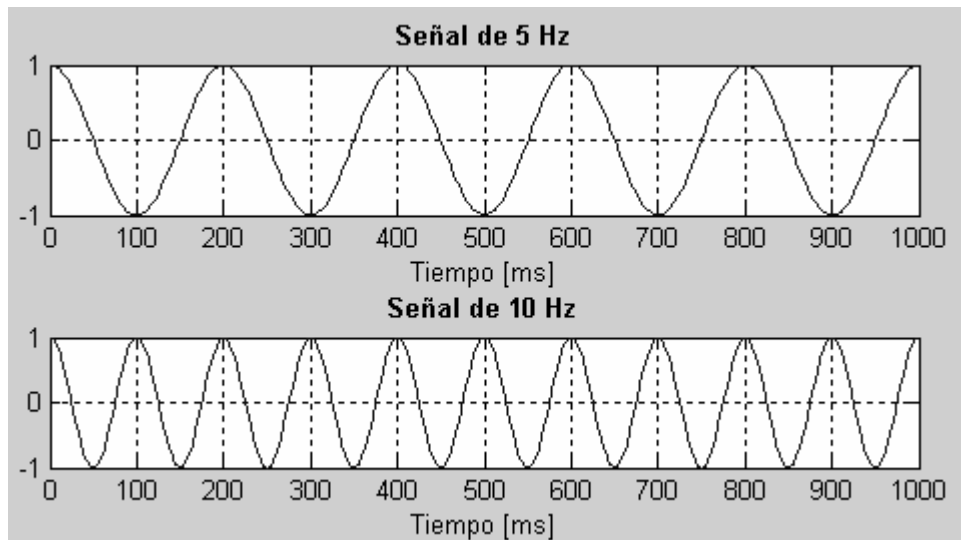


Fig. 1.2: Señales senoidales de 5 y 10Hz.

Capítulo 1. Fundamentos del Análisis de Señales

En la figura 1.3 se muestra una señal sinusoidal con una frecuencia de 50 Hz, a la que se le aplica la transformada de Fourier, representada por el espectro de frecuencias de dicha figura; en la cual se aprecia perfectamente que a lo largo de todo el espectro de frecuencias solamente existe una componente frecuencial de 50 Hz.

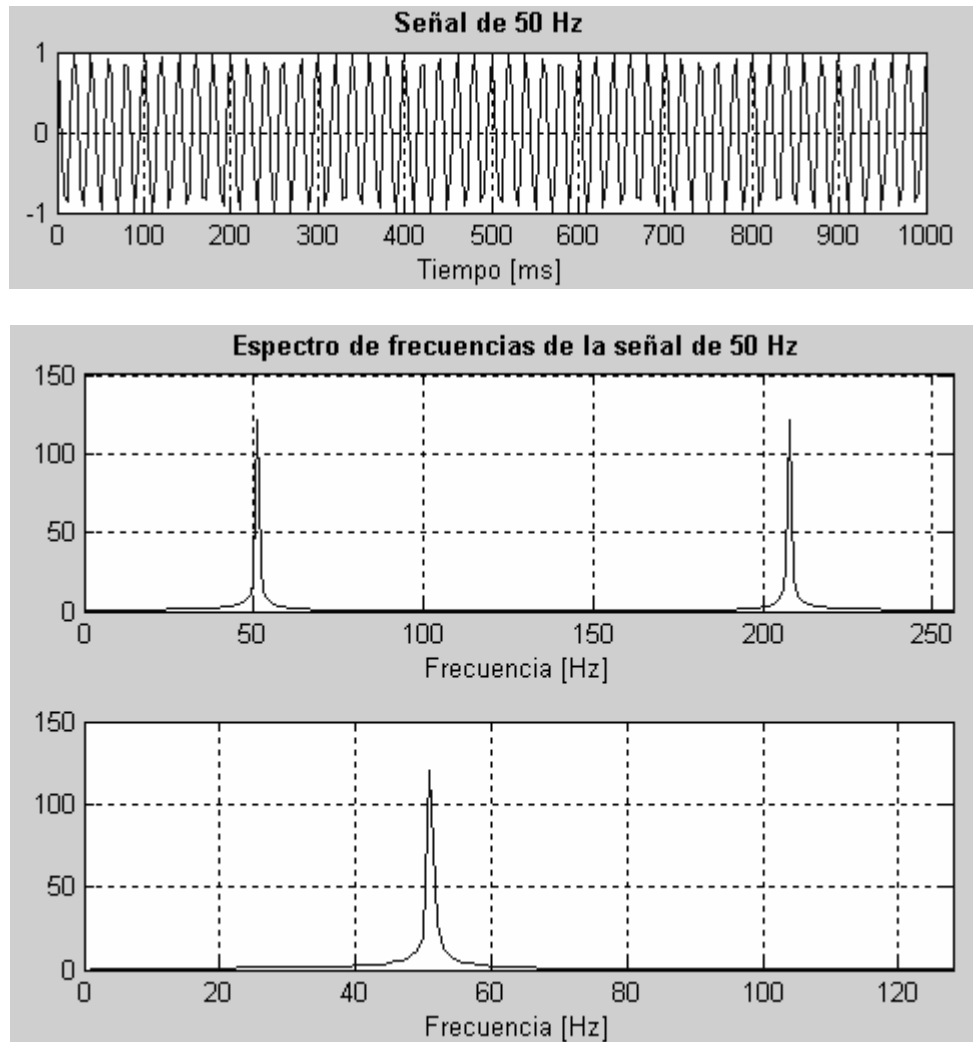


Fig. 1.3: Señal de 50Hz y su Transformada de Fourier

Del espectro de frecuencias mostrado en la figura 1.3 es importante hacer notar que es simétrico y por lo tanto la segunda mitad del espectro es redundante. Además, también es de hacer notar que a través de la Transformada de Fourier se obtiene la frecuencia de la señal, pero no indica el instante de tiempo en el que aparece, esta información no es necesaria cuando la señal es *estacionaria*, sin embargo es de crucial importancia al analizar señales *no estacionarias*. Más adelante en la sección 1.2.2 se profundizará en estos dos diferentes tipos de señales.

1.2.1 Propiedades de la Transformada de Fourier [7]

Con el fin de descubrir lo que ocurre en un dominio cuando en el otro se efectúa una operación elemental sobre la función, se han desarrollado una serie de propiedades de la transformada de Fourier para facilitar el análisis. Dichas propiedades se presentan en esta sección de una manera general.

- **Linealidad (superposición).** La transformación de Fourier es una operación lineal basada en las propiedades de la integración, por lo que puede aplicarse la superposición descrita en la ecuación 1.9.

$$a_1 f_1(t) + a_2 f_2(t) \Leftrightarrow a_1 F_1(\omega) + a_2 F_2(\omega) \quad (1.9)$$

- **Escalamiento en el tiempo.** La expansión/compresión de una onda en el tiempo, afecta su espectro de frecuencias, tal y como lo describe la ecuación 1.10.

$$f(at) \Leftrightarrow \frac{1}{|a|} F\left(\frac{\omega}{a}\right) \quad (1.10)$$

- **Escalamiento en frecuencia.** Esta propiedad es la recíproca a la que se describe en la ecuación 1.10.

$$\frac{1}{|b|} f\left(\frac{t}{b}\right) \Leftrightarrow F(b\omega) \quad (1.11)$$

- **Desplazamiento en el tiempo (retraso).** Si una señal $f(t)$ se retrasa en el tiempo en t_0 , su espectro no se altera, y a cada componente de frecuencia se le agrega una fase negativa ($-\omega t_0$).

$$f(t - t_0) \Leftrightarrow F(\omega)e^{j\omega t_0} \quad (1.12)$$

- **Desplazamiento en frecuencia (modulación).** Al multiplicar una señal por una función armónica (seno o coseno) se provoca un desplazamiento del espectro en frecuencia, a este proceso se le conoce como *modulación*. Es de hacer notar que durante la modulación una mitad del espectro se mueve a frecuencias más altas y la otra a frecuencias menores.

$$f(t)e^{-j\omega_0 t} \Leftrightarrow F(\omega - \omega_0) \quad (1.13)$$

- **Teorema de Convolución.** La convolución en el dominio del tiempo corresponde a la multiplicación en el dominio de la frecuencia.

$$f(t) * h(t) = \int_{-\infty}^{\infty} f(t)h(t - \tau)d\tau \quad (1.14)$$

$$f(t) * h(t) \Leftrightarrow F(\omega)H(\omega)$$

- **Teorema de la Correlación.** La ecuación 1.15 describe el nivel de semejanza entre una señal $f(t)$ con otra señal comparadas con un corrimiento relativo.

$$Corr(f, h) = \int_{-\infty}^{\infty} f(\tau + t)h(t)d\tau \quad (1.15)$$

$$Corr(f, h) \Leftrightarrow F(\omega)H^*(\omega)$$

- **Teorema de Parseval.** Este establece que la energía de la señal es siempre la misma sin depender si se encuentra en el dominio del tiempo o en el dominio de la frecuencia.

$$E = \int_{-\infty}^{\infty} |f(t)|^2 dt = \int_{-\infty}^{\infty} |F(\omega)|^2 d\omega \quad (1.16)$$

1.2.2 Limitaciones de la Transformada de Fourier

A pesar de la enorme utilidad y amplio campo de aplicación de la Transformada de Fourier, esta presenta ciertas limitaciones, tal y como se menciono brevemente en la introducción de este capítulo, específicamente cuando se aplica a señales *no estacionarias*.

Antes de continuar, diremos que una señal *estacionaria* es aquella en la cual todas las componentes de frecuencia existentes en la señal aparecen a lo largo de toda su duración. Mientras que en una señal *no estacionaria*, la frecuencia cambia constantemente a lo largo del tiempo que dura la señal.

Si por ejemplo se tienen dos señales diferentes, ambas con las mismas componentes espectrales, pero con la diferencia de que una de las señales es estacionaria y todas sus componentes frecuenciales están presentes en todo instante de tiempo, tal y como ocurre con la señal definida por la función matemática que se muestra en la ecuación 1.17; y cuya gráfica en el dominio del tiempo y la frecuencia se observan en la figura 1.4.

$$x(t) = \text{Cos}(2\pi \cdot 10t) + \text{Cos}(2\pi \cdot 25t) + \text{Cos}(2\pi \cdot 50t) + \text{Cos}(2\pi \cdot 100t) \quad (1.17)$$

Mientras que por otra parte, tenemos una segunda señal que presenta las mismas cuatro componentes frecuenciales (10, 25, 50 y 100 Hz) de la señal estacionaria definida por la ecuación 1.17, pero en este caso, cada una de estas cuatro componentes de frecuencia son diferentes para cuatro intervalos de tiempo diferentes de la señal, tal y como se ilustra en la figura 1.5, a este tipo de señal se le conoce como *no estacionaria*.

Si se comparan los espectros de frecuencia de las señales mostradas en la figura 1.4 y figura 1.5, se puede concluir que ambos presentan las cuatro componentes espectrales de 10, 25, 50 y 100 Hz. Y a parte del nivel de rizado y la diferencia de amplitud (que siempre puede ser normalizado) ambos espectros son prácticamente idénticos, aunque las señales en el dominio del tiempo son completamente diferentes. Así pues, ambas señales contienen las mismas componentes de frecuencia pero la de la figura 1.4 contiene estas frecuencias para todo el tiempo y la de la figura 1.5 presenta estas frecuencias para diferentes intervalos de tiempo. Este fenómeno se debe a que la Transformada de Fourier solo proporciona el contenido espectral de la señal y no la localización temporal de las componentes espectrales. Este es el motivo por el cual la Transformada de Fourier no es una técnica adecuada para señales *no estacionarias* cuando se desea obtener una correspondencia tiempo-frecuencia.

Con el fin de solventar el inconveniente presentado por la Transformada de Fourier es que fue desarrollada la Transformada de Fourier de Periodo Corto (STFT, *Short Time Fourier Transform*), la cual se expondrá su comportamiento más adelante en la sección 1.4 y la que a su vez es la que da paso al estudio de la Transformada Wavelet, que se presentará en el capítulo 2.

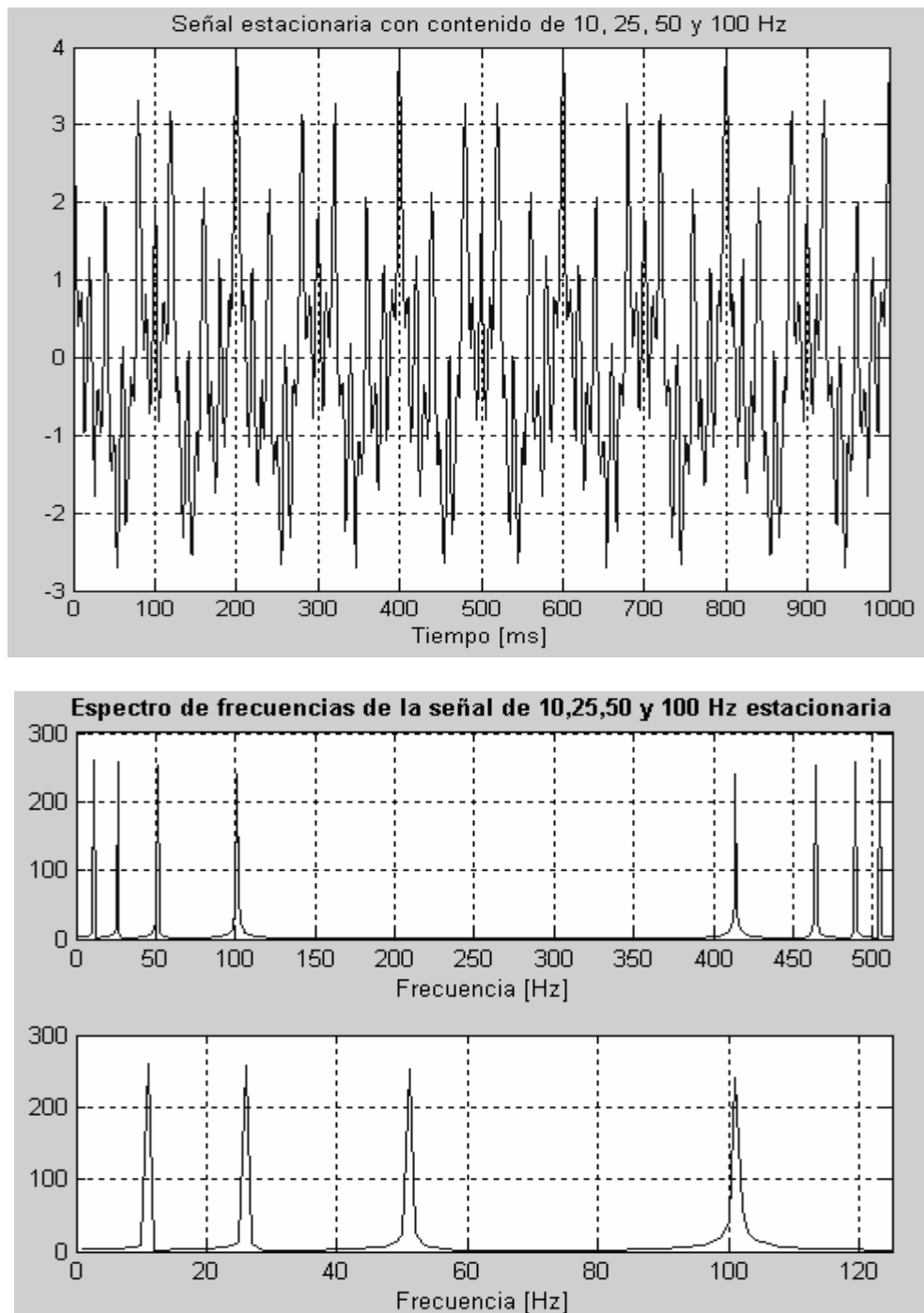


Fig. 1.4: Señal estacionaria y su espectro de frecuencia

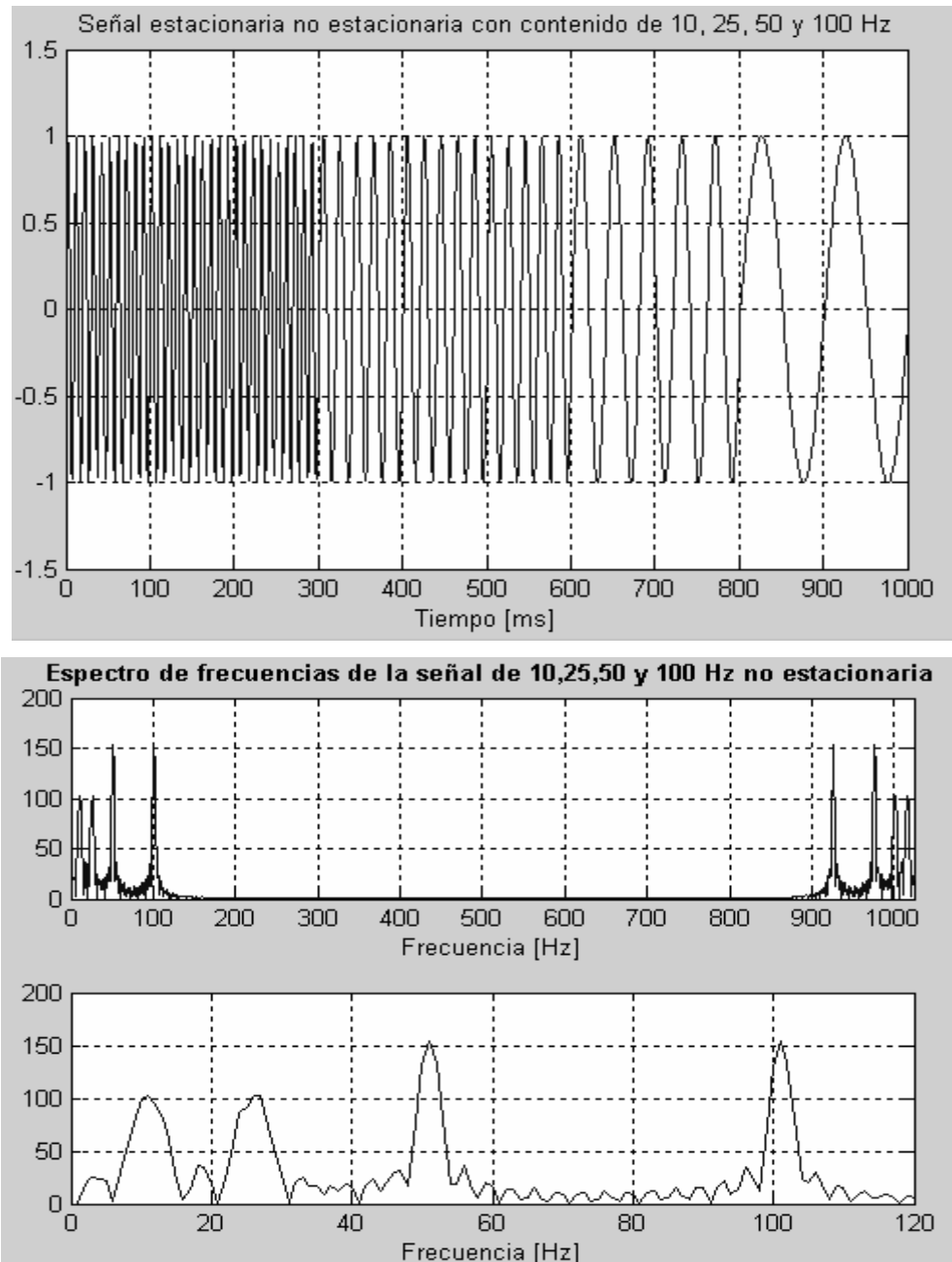


Fig. 1.5: Señal no estacionaria y su espectro de frecuencia

1.3 El principio de Indeterminación de Heisenberg y el Plano Tiempo-Frecuencia

En el ámbito de las ciencias aplicadas usualmente se suele representar una señal física mediante una función del tiempo $f(t)$, o alternativamente, en el dominio de la frecuencia por su Transformada de Fourier $F(\omega)$. Ambas representaciones son en cierto sentido “natural”,

resultantes de enfocar el universo real. Las mismas contienen exactamente la misma información sobre la señal, respondiendo a enfoques distintos y complementarios.

En el campo del procesamiento de señales es común encontrar fenómenos oscilantes localizados en el tiempo. Y también lo es, encontrar ondas perdurables, que con amplitudes casi estacionarias, exhiben patrones de frecuencia variables en el tiempo [5].

Surge entonces, la noción de la existencia de fenómenos localizados en tiempo y frecuencia, es decir que existen objetos que para su descripción requieren información conjunta de ambos dominios. Por supuesto, el par de Transformadas de Fourier no es la herramienta idónea para expresar este tipo de información conjunta; por los motivos ya explicados anteriormente.

En contraposición, para la representación de dichos fenómenos se requiere de patrones elementales capaces de localizar conjuntamente la información de ambos dominios. Lo cual plantea el problema de las representaciones tiempo-frecuencia.

En principio, este es un problema antinatural ya que, de acuerdo al *Principio de Indeterminación de Heisenberg*, en su formulación original, expresa que no pueden conocerse simultáneamente el momento y la posición de una partícula en movimiento [13]. Si ahora lo aplicamos a nuestro estudio, implica que no se puede conocer que componentes espectrales existen un determinado instante de tiempo. Y lo más que se puede hacer es investigar que componentes espectrales existen en un cierto intervalo de tiempo, lo cual nos plantea un problema de *resolución*. Y se imponen por tanto soluciones de compromiso, ya que el principio de indeterminación afirma que no es posible reducir arbitrariamente a la vez la resolución en el tiempo (Δt) y la resolución en frecuencia ($\Delta \omega$) debido a que su producto esta acotado inferiormente por la desigualdad de Heisenberg, expresada en la ecuación 1.18.

$$\Delta t \Delta \omega \geq \frac{1}{2} \quad (1.18)$$

Esta desigualdad implica que se debe sacrificar la resolución en el tiempo por la resolución en frecuencia o viceversa.

Tratándose de fenómenos aislados en el tiempo, la cuestión es relativamente sencilla. Podemos analizarlos separadamente, y sobre sus respectivos dominios temporales extraer la información en frecuencia que nos provee la Transformada de Fourier. Análogamente podemos tratar el caso de ondas con patrones de frecuencia relativamente simples, moduladas por funciones relativamente largas.

Pero el problema se vuelve arduo cuando nos encontramos frente a señales donde conviven múltiples fenómenos localizados en el tiempo y en la frecuencia, superponiéndose bajo complejas estructuras.

Claramente, por las razones arriba expuestas, no existe una respuesta única. Entonces, el desafío consiste en definir una apropiada representación acorde con las características de la señal y a los objetivos de su procesamiento.

1.4 La Transformada de Fourier de Período Corto

Como ya se discutió, la Transformada de Fourier constituye una herramienta mediante la cual podemos obtener información sobre como esta distribuida la energía de una señal a través de sus distintas componentes de frecuencia, es decir, podemos conocer todas las componentes de frecuencia existentes en la señal y sus respectivos aportes energéticos. Lo cual indica que la Transformada de Fourier tiene una perfecta resolución en frecuencia, lo que la hace una herramienta matemática de muchísima utilidad para el análisis de señales estacionarias. Sin embargo, ella no puede ser aplicada con el objeto de obtener información de cuándo o dónde las diferentes componentes de frecuencia se encuentran en las señales *quasi-estacionarias* o *no estacionarias*, cuyo contenido espectral varía con el tiempo. Es decir, la Transformada de Fourier posee una muy pobre resolución en el tiempo.

Con el afán de superar los inconvenientes que presenta la Transformada de Fourier frente a señales no estacionarias, en 1946 Denis Gabor realiza una adaptación de la Transformada

de Fourier de forma tal que pueda obtenerse una representación tiempo-frecuencia [1], y que en la actualidad es ampliamente utilizada en el área del procesamiento de señales.

A esta adaptación de la Transformada de Fourier realizada por Gabor se le conoce como *Transformada de Fourier de Periodo Corto (STFT)*, la cual consiste básicamente en dividir la señal de interés en diferentes partes donde se pueda asumir que la señal es estacionaria.

La forma de dividir la señal se realiza mediante una función ventana $h(t)$ cuyo ancho o soporte corresponde a la longitud de cada segmentación de la señal. Así pues, con la función ventana encuadramos la señal alrededor de un instante de tiempo τ y calculamos su transformada de Fourier, luego trasladamos la función ventana hasta que no se sobrepona con la anterior cubriendo una nueva porción de la señal a la que volvemos a calcular su transformada de Fourier. Este proceso es repetido hasta que se ha cubierto la totalidad de la señal.

Partiendo de lo expresado anteriormente, en forma matemática, la Transformada de Fourier de Periodo Corto se expresa como se muestra en la ecuación 1.19:

$$STFT(t, \omega) = \int_{-\infty}^{\infty} f(t)h^*(\tau - t)e^{-j\omega t} dt \quad (1.19)$$

Como lo expresa la ecuación 1.19, la Transformada de Fourier de Periodo Corto no es más que la Transformada de Fourier de la señal previamente multiplicada por una función ventana. Por lo tanto se obtiene una representación real de la señal en tiempo y frecuencia. La Transformada de Fourier de Periodo Corto es bidimensional (en un eje aparece el tiempo y en el otro la frecuencia) o tridimensional, si consideramos la amplitud.

Y considerando, como ya se dijo, a $h(t)$ como una función ventana de valores sólo reales, no complejos, de tal manera que se cumpla la relación establecida en la ecuación 1.20,

$$h(-t) = h^*(t) \quad (1.20)$$

y si se considera lo expresado por la ecuación 1.20, esta puede reescribirse de la forma que se presenta en la ecuación 1.21,

$$STFT(t, \omega) = \int_{-\infty}^{\infty} f(t)h(t - \tau)e^{-j\omega\tau} dt \quad (1.21)$$

la cual calcula el producto interno entre la señal y la función ventana trasladada y modulada.

En la figura 1.6 se muestra un función ventana de tipo Gaussiana, la primera porción de la señal muestra la ventana localizada en $t = \tau_1$, la del medio en $t = \tau_2$ y la de más a la derecha en $t = \tau_3$. Estas ventanas comprenden a tres Transformadas de Fourier en tres tiempos distintos. Por lo tanto se obtendrá una buena representación tiempo-frecuencia de la señal.

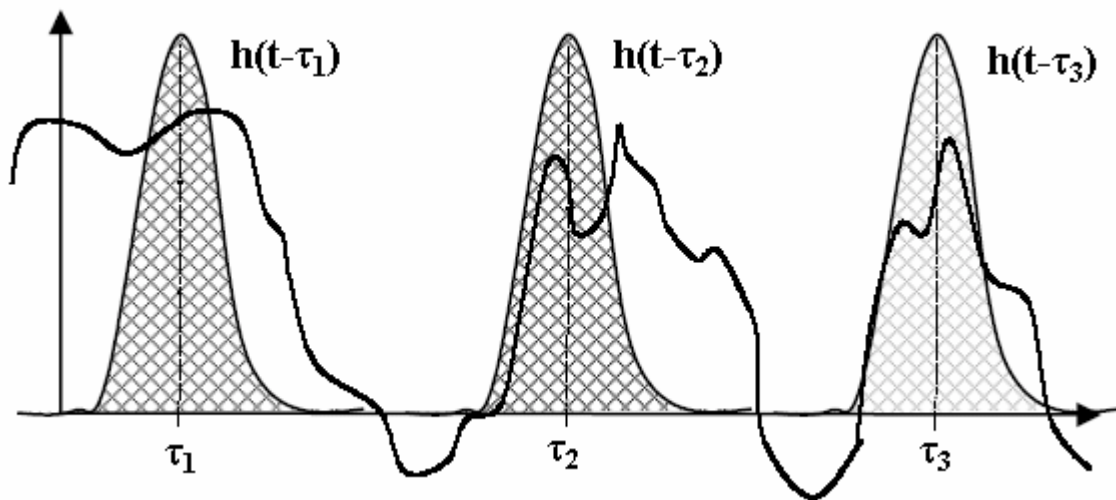


Fig. 1.6: Representación gráfica de la STFT

Una forma de comprender como trabaja la STFT es a partir de un ejemplo, para ello consideraremos la forma de onda de la figura 1.5. Al obtener la STFT de dicha señal (la cual se muestra en la figura 1.7) se observa que dicha señal se puede representar en tres dimensiones (tiempo, frecuencia y amplitud) y se advierte que la gráfica es simétrica con respecto al punto medio del eje de frecuencia, puesto que la STFT es la versión en ventanas de la Transformada de Fourier Clásica, la cual como se comentó en secciones anteriores, siempre es simétrica.

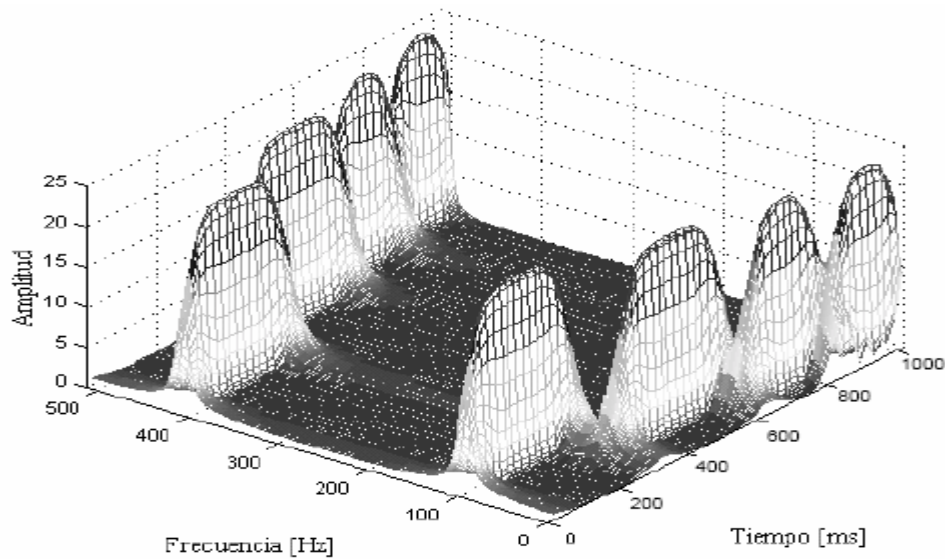


Fig. 1.7: STFT de la señal no estacionaria de la Fig. 1.5, con ventana Gaussiana $a=180$

Lo importante de hacer notar en la figura 1.7 es que existen cuatro picos que corresponden a las cuatro componentes de frecuencia de la señal ilustrada en la figura 1.5 y que además están ubicados en diferentes intervalos de tiempo. Por lo tanto se cuenta con una representación tiempo-frecuencia de la señal.

Entonces, podría parecer que el problema de representación tiempo-frecuencia de una señal estaría resuelto. Sin embargo, existe un problema que se remonta al principio de indeterminación de Heisenberg [1], que en este caso, tal y como se estudió en la sección 1.3, se traduce en que no es posible conocer la representación exacta tiempo-frecuencia de una señal, sino tan solo los intervalos de tiempo en los cuales existen determinadas bandas de frecuencia, por lo tanto aparece un problema de resolución.

En la transformada de Fourier no existe problema de resolución en el dominio de la frecuencia, pues se sabe exactamente las frecuencias que existen, de manera similar no existe problema de resolución en el dominio del tiempo, ya que se conoce el valor de la señal para cada instante de tiempo.

Lo que proporciona la perfecta resolución en frecuencia en la transformada de Fourier es el hecho de que la ventana empleada es la función exponencial $e^{j\omega t}$, la cual existe para todo

instante de tiempo $[-\infty, +\infty]$. En la STFT la ventana es de longitud finita, es decir solo se aplica a una parte de la señal, causando una disminución de la resolución en frecuencia, con lo cual solo es posible conocer una banda de frecuencias y no un valor exacto de frecuencia.

En consecuencia, existe un compromiso entre buena resolución en el tiempo o buena resolución en frecuencia. Para obtener la estacionalidad se elige una ventana lo suficientemente estrecha en la cual la señal sea estacionaria. Cuanto más estrecha sea la ventana se obtendrá mejor resolución en el tiempo y por lo tanto una mejor representación de la estacionalidad y peor resolución en frecuencia. Por tanto, el problema consiste en la selección de una ventana para el análisis, dependiendo de la aplicación. Si las componentes frecuenciales están bien separadas unas de otras en la señal original, se puede sacrificar resolución en la frecuencia y tratar de mejorar la resolución en el tiempo. Tal y como se ejemplifica en la figura 1.8, donde se muestran dos posibilidades, dependiendo de la resolución deseada en tiempo y frecuencia.

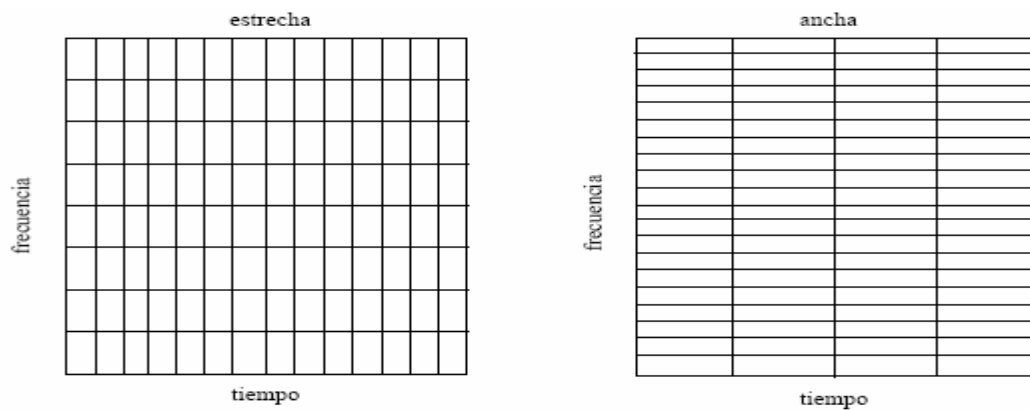


Fig. 1.8: Enrejado resultante de la STFT en el plano tiempo - frecuencia

En conclusión se puede afirmar que con una *ventana estrecha* existe buena resolución en el tiempo y pobre resolución en el dominio de la frecuencia. Mientras que con una *ventana ancha* se tiene buena resolución en el dominio de la frecuencia y pobre resolución en el dominio del tiempo.

1.5 La Transformada Discreta de Fourier

La utilización cada día más creciente de los sistemas computacionales en todos los ámbitos de la vida cotidiana, y en particular en el procesamiento digital de señales, ha provocado que la versión discreta de la transformada de Fourier sea de mucho interés [7]. Ello se debe al hecho de que las computadoras trabajan únicamente con valores discretos o muestras. Por lo que el cálculo numérico de la transformada discreta de Fourier también se basa en las muestras obtenidas de una señal $f(t)$; los cuales son de la forma f_k , con $k = 0, 1, 2, \dots$

Y si se mantiene la relación tiempo-frecuencia existente en la transformada de Fourier, se intuye entonces que es posible calcular la transformada $F(\omega)$ solo para valores discretos de ω , con lo que se obtienen valores de la transformada de la forma F_n con $n = 0, 1, 2, \dots$

Por lo tanto, sea $f(t)$ una señal periódica de período T y sólo se conocen sus valores en N puntos igualmente espaciados en el tiempo. Entonces, si $f(kT_s)$ corresponde a la k -ésima muestra de $f(t)$ y $F(n\omega_s)$, donde $\omega_s = 2\pi f_s$ (f_s es la frecuencia con la que se realiza el proceso de muestreo) corresponde a la n -ésima muestra de $F(\omega)$, y además definimos a N como el número de muestras de la señal o longitud de la señal, podemos reescribir la Transformada de Fourier, de una señal de período T , en su forma discreta como se muestra en la ecuación 1.22. La cual es conocida como la Transformada Discreta de Fourier.

$$F_n = \sum_{k=0}^{N-1} f_k e^{j2\pi kn/N} \quad n, k = 0, 1, 2, \dots, N-1 \quad (1.22)$$

Ahora bien, si se parte de los coeficientes F_n también se pueden obtener los valores de f_k en forma análoga a lo expresado en la ecuación 1.22, así:

$$f_k = \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{-j2\pi kn/N} \quad n, k = 0, 1, 2, \dots, N-1 \quad (1.23)$$

A la ecuación 1.23 se le conoce como la Inversa de la Transformada Discreta de Fourier. Como se puede observar, la TFD y la TIFD forman un par exacto de transformada [17].

Capítulo 1. Fundamentos del Análisis de Señales

En el presente capítulo únicamente se brindó un muy breve repaso a los principales tópicos que dieron origen al procesamiento de señales y no se extendió en mucho detalle matemático de temas tan importantes como es la Transformada de Fourier, ya que no es el principal objetivo de este documento. Para mayor información referente a este tema el lector puede consultar literatura del área de comunicaciones o procesamiento digital de señales. En los siguientes capítulos nos enfocaremos en la teoría que da origen a la aplicación que se desarrollará.

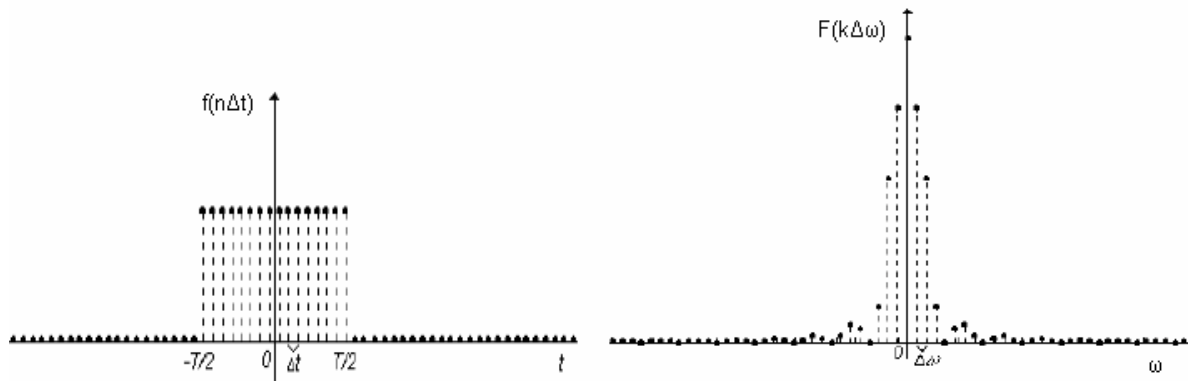


Fig. 1.9: Representación de señal cuadrada y su Transformada Discreta de Fourier

Capítulo II

Teoría de Wavelets

2.1 Introducción

Tal y como se demostró en el capítulo 1, la transformada de Fourier presenta problemas de resolución tiempo-frecuencia y que son explicados perfectamente por el principio de indeterminación de Heisenberg, sin embargo, ante estos inconvenientes, es posible analizar cualquier señal empleando una técnica alternativa llamada *Análisis Multiresolución*. El análisis multiresolución es el pilar sobre el cual se fundamenta la teoría de Wavelets, teoría que es el principal interés del presente capítulo y sobre la cual reside el eje principal del presente trabajo de graduación.

El análisis multiresolución, tal y como se verá a lo largo del capítulo, está diseñado para proporcionar una buena resolución temporal y pobre resolución en frecuencia para las altas frecuencias y buena resolución en frecuencia y pobre en tiempo para las bajas frecuencias. Por lo que el principio de indeterminación de Heisenberg no interfiere o no es contradicho en este tipo de análisis. Esta forma de llevar a cabo el análisis de una señal adquiere un sentido especial cuando dichas señales tienen componentes de alta frecuencia de corta duración y componentes de baja frecuencia de larga duración.

Además, se presenta también una descripción de los aspectos básicos, características, tipos, principios matemáticos, etc., que dan soporte a la teoría de Wavelets y la forma de trabajar de dicha teoría. Además, se explica como es que la transformada Wavelet resuelve los problemas inherentes a la STFT.

2.2 Antecedentes históricos de la teoría de Wavelets

La teoría de Wavelets fue aplicada en un primer momento en el área de la geofísica para analizar específicamente datos obtenidos en estudios de sismología. De hecho, los geofísicos redescubrieron dicha teoría; ya que los matemáticos la habían desarrollado unos veinte años atrás para solucionar problemas abstractos, aunque nunca se imaginaron que podría ser aplicada en el campo del procesamiento de señales [2].

Los estudios sismológicos son construidos a partir de muchas imágenes en dos dimensiones de porciones de roca o de suelo, que son unidas para brindar una imagen tridimensional de la estructura de la roca debajo de la superficie del suelo. Cada una de las imágenes es obtenida colocando geófonos (“micrófonos” sísmicos) a intervalos igualmente espaciados a lo largo de una línea, llamada la “línea del sismo”. Luego, utilizando dinamita colocada al final de dicha línea provocan una onda sísmica en el suelo. Así pues, cada uno de los geófonos colocados a lo largo de la línea graba el movimiento de la tierra debido a la explosión provocada por la dinamita, desde el principio hasta el fin de la misma, esta grabación es conocida como la *traza sísmica* y tiene la forma que se observa en la figura 2.1 [2].

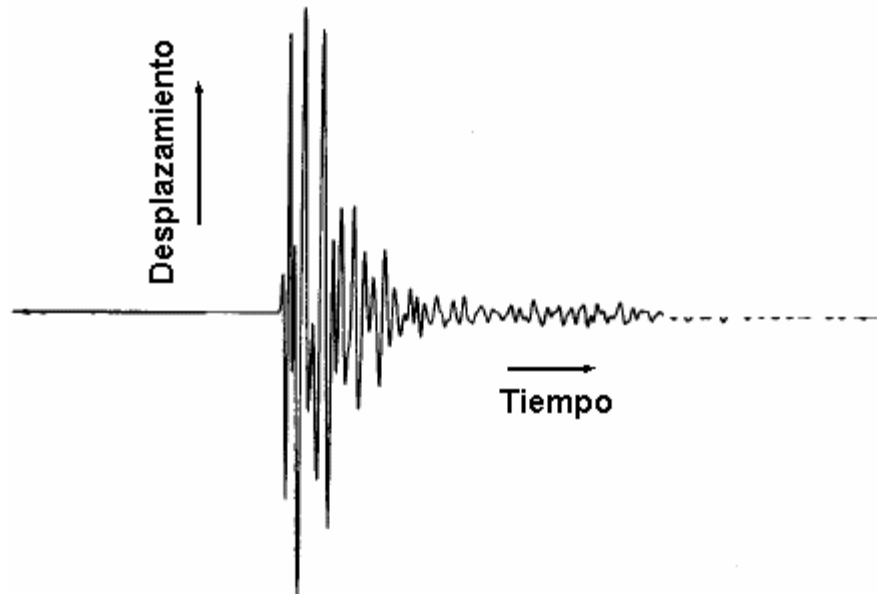


Fig. 2.1: Traza sísmica típica

En la figura 2.1 se puede observar que la traza es descrita por una gráfica desplazamiento-tiempo, en la cual son de vital importancia tanto la oscilación como el momento en que ella ocurre.

La primera onda grabada por el geófono es la onda directa, que viaja a lo largo de toda la superficie del suelo, y que usualmente no es de mucha importancia. Las subsiguientes ondas son reflejadas por las capas de rocas debajo del suelo, estas son las verdaderamente importantes. El conocimiento del momento en que la onda golpea el geófono brinda información que indica la posición de la capa que reflejo dicha onda. El movimiento del suelo que la onda produce, brinda algunos detalles muy específicos de la capa de rocas. Las trazas obtenidas a partir de todos los geófonos juntos a lo largo de una línea pueden ser combinadas para dar una muy buena “imagen” del suelo directamente por debajo de la línea sísmica [2].

El principal factor que determina la exactitud y precisión de un estudio sismológico es la utilización de una herramienta de análisis apropiada para el estudio de cada una de las trazas. Así pues, la Transformada de Fourier no es una buena herramienta de análisis en este caso. Ya que solo puede brindarnos información frecuencial (las oscilaciones que componen la onda) tal y como se describió en el capítulo 1. Dicha transformada no brinda información de cuando la oscilación ocurrió. Por otro lado, la Transformada de Fourier de Periodo Corto (STFT), es mejor. En dicha técnica, como ya se ha explicado, el intervalo de tiempo total es dividido en varios pequeños intervalos de tiempo igualmente espaciados, los cuales posteriormente son analizados por separado haciendo uso de la Transformada de Fourier. El resultado de dicho análisis ahora contiene información en el dominio del tiempo y de la frecuencia. Sin embargo, existe un problema con esta aproximación. Los pequeños intervalos de tiempo en que se dividió la señal entera no son ajustables, en los intervalos de muy corta duración, componentes de muy alta frecuencia ocurren repentinamente las cuales son muy difíciles de detectar.

Es aquí donde se desvela la utilidad de la teoría de Wavelets. Los Wavelets son capaces de rastrear la información de la onda en el dominio del tiempo y de la frecuencia

simultáneamente. Esto implica que esta teoría puede utilizarse para realizar un acercamiento (zoom in) de una onda durante el apareamiento brusco de las altas frecuencias, o bien, realizar una toma lejana (zoom out) de la señal y de esta forma detectar oscilaciones de muy baja frecuencia.

Luego de describir brevemente los fenómenos que dieron origen a la teoría de Wavelets, a continuación se presenta de manera resumida su desarrollo a lo largo del tiempo y sus principales precursores, esto con el fin de ir adentrándonos poco a poco al análisis formal de dicha teoría.

Después de que en 1807 Joseph Baptiste Fourier desarrollara toda la teoría del análisis de señales que se estudió en el capítulo anterior, los matemáticos gradualmente fueron trascendiendo en su conocimiento del *análisis en frecuencia* al *análisis en escala*. Es así como en 1873 Karl Weierstrass describió una familia de funciones que son formadas por una superposición de copias *escaladas* de una función base dada. Las funciones que él definió son *fractales*, en el sentido de que son continuas en todos sus puntos y diferenciables en ninguno. Sin embargo, la teoría de Wavelets como tal, fue mencionada por primera vez en uno de los apéndices de la tesis de Alfred Haar en 1909, quien desarrolló el primer sistema ortonormal de funciones con soporte compacto, ahora llamada *base de Haar*, y que será tratada con detenimiento más adelante en la sección 2.5. Sin embargo, desafortunadamente dicha Wavelet no es continuamente diferenciable, lo cual limita seriamente sus aplicaciones. Es importante hacer notar que esta base aún sirve como fundamento para la teoría moderna de Wavelets [9].

Ya para los años 30's, muchos grupos de investigadores trabajaban de manera independiente en el estudio de la representación de funciones utilizando las *funciones base de escala variable* [3]. Es importante la comprensión completa de dichas funciones para comprender de una mejor manera la teoría de Wavelets.

Haciendo uso de las funciones base de escala variable, llamada función base Haar, el físico Paul Levy en 1930 investigó el movimiento Browniano (señal aleatoria). Y llegó a la

conclusión de que la función base Haar es superior a las funciones base de Fourier en el estudio de muy pequeños y complicados detalles en el movimiento Browniano. Por otra parte, siempre en los años 30's en una investigación realizada por Littlewood, Paley y Stein, la cual involucraba el cálculo de la energía de una señal $f(x)$, descrita por la ecuación 2.1,

$$Energía = \frac{1}{2} \int_0^{2\pi} |f(x)|^2 dx \quad (2.1)$$

observaron que dicho cálculo producía diferentes resultados si la energía se encontraba concentrada alrededor de unos pocos puntos o si estaba distribuida sobre un intervalo bastante grande. Dichos resultados perturbaron a los científicos de aquella época, debido a que esto implicaba claramente un incumplimiento del principio de conservación de la energía. Así pues, fue como los investigadores descubrieron una función que puede variar en escala y que al calcular la energía no violaba dicho principio [3].

Luego, otro avance significativo se dio en 1946, cuando Dennis Gabor describió una base no-ortonormal de lo que ahora se llaman Wavelets con soporte no acotado, basado en funciones gaussianas trasladadas. El término Wavelet proviene del campo de la sismología, el cual fue bautizado por Ricker en 1940 para describir el disturbio resultante de un impulso sísmico agudo o una carga explosiva. En 1982, Morlet mostró como estos wavelets sísmicos podrían ser modelados eficientemente con las funciones matemáticas que Gabor había definido casi cuarenta años atrás [3], [9].

Posteriormente en 1980, Grossman y Morlet un físico y un ingeniero, definieron de forma general los Wavelets en el campo de la física cuántica y demostraron como señales arbitrarias pueden ser analizadas en términos de escalamientos y traslaciones de una función *wavelet madre*. Yves Meyer y Stephane Mallat ampliaron esta noción a una teoría llamada *análisis multiresolución*. Y en 1985 Mallat demostró como esta teoría se puede utilizar en el procesamiento digital de imágenes y en el análisis de señales. A diferencia de la Wavelet Haar, las Wavelets de Meyer son continuamente diferenciables en cualquier punto, sin embargo, ellas no son de soporte compacto. Un par de años más tarde, Ingrid

Daubechies se baso en el trabajo de Yves Mallat para definir un conjunto de Wavelets basadas en funciones ortonormales que son tal vez las Wavelets más elegantes y la piedra angular de las aplicaciones basadas en Wavelets de hoy en día [12].

2.3 Principios matemáticos que rigen la teoría Wavelets [2]

En la presente sección se presentará la base matemática que brindará un sólido respaldo a la definición de las funciones Wavelets como base de los espacios de funciones lineales de cuadrado integrable $L^2(R)$.

Sean dos vectores $X = (x_1, x_2, x_3)$, $Y = (y_1, y_2, y_3)$ en un espacio tridimensional (R^3) el producto interno entre ambos vectores queda definido por la ecuación 2.2 de la siguiente forma:

$$\langle X, Y \rangle = x_1 y_1 + x_2 y_2 + x_3 y_3 \quad (2.2)$$

Esta definición parcialmente tiene su origen en la necesidad de medir la longitud de un vector, longitud que tiene su origen en el Teorema de Pitágoras expresado por la ecuación 2.3.

$$\text{Longitud de } X = \sqrt{x_1^2 + x_2^2 + x_3^2} = \sqrt{\langle X, X \rangle} \quad (2.3)$$

Partiendo de este punto, se puede decir que el principal interés de esta sección será definir el concepto de un producto interno en un sentido más general que incluya una amplia variedad de espacios vectoriales.

2.3.1 Espacios con producto interno

La definición de un espacio producto interno real de dimensión finita (*espacio Euclidiano*) en tres dimensiones, naturalmente puede ser generalizado a R^n para cualquier dimensión n . Así pues, para dos vectores $X = (x_1, x_2, x_3, \dots, x_n)$, $Y = (y_1, y_2, y_3, \dots, y_n)$ en R^n , el producto interno Euclidiano esta dado por la ecuación 2.4, que se presenta a continuación:

$$\langle X, Y \rangle = \sum_{j=1}^n x_j y_j \quad (2.4)$$

Al estudiar las Series y la Transformada de Fourier, se hace mucho uso del factor exponencial complejo. Así pues, ahora se considerarán tanto los espacios vectoriales complejos como los espacios vectoriales reales. De esta manera la definición anterior del producto interno para R^n puede ser modificada para vectores en C^n conjugando el segundo factor. Hay que recordar que el conjugado de un número complejo $z = x + jy$ está definido como $\bar{z} = x - jy$. Y es importante hacer notar la relación expresada en la ecuación 2.5.

$$z\bar{z} = x^2 + y^2 \quad (2.5)$$

Lo cual por definición es $|z|^2$ (el cuadrado de la longitud de $z = x + jy$) considerando el vector que va desde el punto (0,0) hasta el punto (x,y).

Ahora bien, si se tienen dos vectores definidos en C^n , $Z = (z_1, z_2, \dots, z_n)$ y $W = (w_1, w_2, \dots, w_n)$, entonces el producto interno entre ambos vectores puede ser expresado por la ecuación 2.6.

$$\langle Z, W \rangle = \sum_{j=1}^n z_j \bar{w}_j \quad (2.6)$$

El propósito de la existencia del conjugado presente en la ecuación 2.6 es asegurar que la longitud de un vector en C^n sea real y positiva, tal y como lo demuestra la ecuación 2.7.

$$\text{Longitud de } Z = \sqrt{\langle Z, Z \rangle} = \sqrt{\sum_{j=1}^n z_j \bar{z}_j} = \sqrt{\sum_{j=1}^n |z_j|^2} \quad (2.7)$$

Por definición, un producto interno de un espacio vectorial V es una función que cumple con la siguiente relación establecida por la ecuación 2.8, y que se presenta a continuación:

$$\langle \cdot, \cdot \rangle: V \times V \rightarrow C \quad (2.8)$$

Y que a su vez satisface las propiedades que se presentan a continuación:

- Bilinealidad: $\langle X + Y, Z \rangle = \langle X, Z \rangle + \langle Y, Z \rangle$ y $\langle X, Y + Z \rangle = \langle X, Y \rangle + \langle X, Z \rangle$.
- Positividad: $\langle v, v \rangle > 0$ para todo $v \in V$.
- Simetría conjugada: $\overline{\langle v, w \rangle} = \langle w, v \rangle$ para todo vector v y w en V .
- Homogeneidad: $\langle cv, w \rangle = c \langle v, w \rangle$ para todo vector v y w en V y todo escalar $c \in C$.
- Adición: $\langle u + v, w \rangle = \langle u, w \rangle + \langle v, w \rangle$ para todo vector $u, v, w \in V$.

Las definiciones anteriores también son de utilidad para definir un producto interno real en un espacio vectorial real, excepto que el valor escalar c en la propiedad de homogeneidad es real y no tiene conjugado en la definición de la propiedad de simetría.

Es de hacer notar que la tercera y quinta propiedades implican bilinealidad en el segundo factor. Mientras que la tercera y cuarta propiedades implican que el factor escalar sale del segundo factor como un conjugado, tal y como puede apreciarse en la ecuación 2.9.

$$\langle v, cw \rangle = \overline{\langle cw, v \rangle} = \overline{c \langle w, v \rangle} = \bar{c} \langle v, w \rangle \quad (2.9)$$

La condición de positividad implica que podemos asignar cualquier número diferente de cero, $\|v\| = \sqrt{\langle v, v \rangle}$ como la longitud del vector v . La noción de longitud da idea de la distancia entre dos vectores en V definida por la ecuación 2.10 que se presenta a continuación:

$$\text{Distancia entre } \{v, w\} = \|v - w\| \quad (2.10)$$

Nótese que la propiedad de positividad de los productos internos implica que la única forma de que se cumpla $\|v - w\| = 0$ es cuando $v = w$. Esta noción de distancia también da la idea de una secuencia convergente $\{v_k; k = 1, 2, \dots\}$; por lo que se establece que:

$$v_k \rightarrow v \text{ si } \|v_k - v\| \rightarrow 0$$

Es decir, que $v_k \rightarrow v$ si la distancia entre v_k y v es pequeña comparándola con el valor de k .

2.3.2 El espacio cuadrado integrable L^2

En la presente sección, se discutirán una clase particular de espacios vectoriales dimensionalmente infinitos que son particularmente útiles en el campo del análisis de señales. Así pues, una señal de audio puede ser vista como una función $f(t)$, que representa la intensidad de la señal en el tiempo. Aquí t varía en un intervalo $a \leq t \leq b$ que representa la duración en el tiempo de la señal. Aquí, a podría ser $-\infty$ ó b ser $+\infty$.

Debido a la necesidad de imponer una restricción del crecimiento de una función definida en el intervalo $a \leq t \leq b$, por definición se puede establecer que: *Para un intervalo $a \leq t \leq b$, el espacio $L^2([a, b])$ es el conjunto de todas las funciones cuadrado integrables definidas en $a \leq t \leq b$.* Tal y como se define en la ecuación 2.11,

$$L^2([a, b]) = \left\{ f : [a, b] \rightarrow C; \int_a^b |f(t)|^2 dt < \infty \right\} \quad (2.11)$$

las funciones discontinuas también son incluidas como miembros de este espacio cuadrado integrable. Así pues, en este contexto la integral anterior puede ser interpretada en el sentido de una suma de Riemann. La definición de L^2 incluye funciones cuyo conjunto de discontinuidades es levemente grande, en cuyo caso debe ser utilizada la integral de Lebesgue. La condición $\int_a^b |f(t)|^2 dt < \infty$ físicamente implica que el total de la energía de la señal es finita (la cual es una clase de señal real).

El espacio $L^2([a, b])$ es dimensionalmente infinito. Por ejemplo, si $a = 0$ y $b = 1$, entonces el conjunto de funciones $\{1, t, t^2, t^3 \dots\}$ es linealmente independiente y pertenece a $L^2([0, 1])$. La función $f(t) = 1/t$ es un ejemplo de una función que no pertenece a $L^2([0, 1])$ ya que

$$\int_0^1 (1/t)^2 dt = \infty.$$

Ahora se construirá un apropiado producto interno de $L^2[a, b]$. Para motivar el producto interno L^2 , se discretizará el intervalo $[a, b]$. Con el fin de simplificar el análisis, haremos que $a = 0$ y $b = 1$.

Sea N un entero positivo mucho mayor que cero y sea $t_j = j/N$ para $1 \leq j \leq N$. Si f es continua, entonces los valores de f en el intervalo $[t_j, t_{j+1}]$ pueden ser aproximados por $f(t_j)$. Por lo tanto, f puede ser aproximada por el siguiente vector de la ecuación 2.12. Tal y como se ilustra en la figura. 2.2.

$$f_N = (f(t_1), f(t_2), \dots, f(t_N)) \in R^N \quad (2.12)$$

Ya que el valor de N es muy grande, f_N es una muy buena aproximación de f .

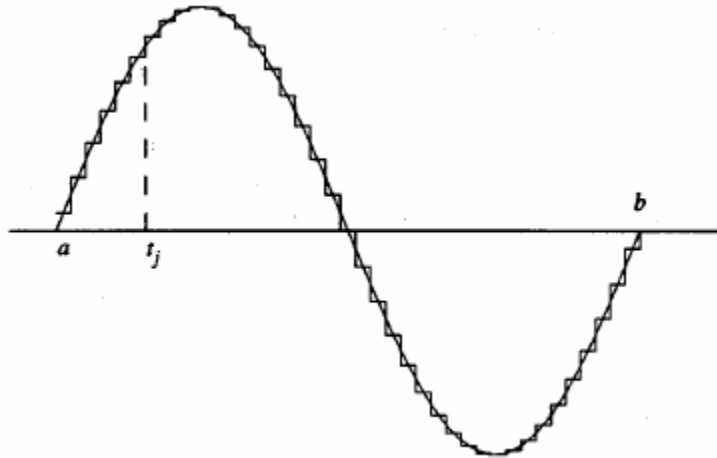


Fig. 2.2: Discretización de una señal continua

Ahora bien, si f y g son dos señales en $L^2[0, 1]$, entonces ambas señales pueden ser discretizadas por f_N y g_N . Una posible definición de $\langle f, g \rangle_{L^2}$ es examinar el producto interno ordinario R^N de f_N y g_N que se detalla en la ecuación 2.13.

$$\langle f_N, g_N \rangle_{R^N} = \sum_{j=1}^N f(t_j) \overline{g(t_j)} = \sum_{j=1}^N f(j/N) \overline{g(j/N)} \quad (2.13)$$

El problema con la aproximación expresada en la ecuación 2.13 radica en que el valor de N es muy grande, por lo que la suma del lado derecho es muy grande. Una mejor opción es considerar el promedio del producto interno expresado en la ecuación 2.14, así:

$$\frac{1}{N} \langle f_N, g_N \rangle_{R^N} = \sum_{j=1}^N f(j/N) \overline{g(j/N)} \frac{1}{N} \quad (2.14)$$

Ya que f_N y g_N se aproximan a f y g cuando el valor de N es grande, una definición razonable de $\langle f, g \rangle_{L^2}$ es tomar el límite del promedio del producto interno cuando $N \rightarrow \infty$.

Así pues, la ecuación anterior puede ser escrita como se muestra en la ecuación 2.15.

$$\frac{1}{N} \langle f_N, g_N \rangle_{R^N} = \sum_{j=1}^N f(j/N) \overline{g(j/N)} \Delta t \quad \text{donde } \Delta t = \frac{1}{N} \quad (2.15)$$

La suma del término de la derecha es una aproximación de la suma de Riemann a $\int_0^1 f(t) \overline{g(t)} dt$ sobre la partición $[0, t_1, t_2, \dots, t_N = 1]$ de $[0, 1]$. Esta aproximación es mucho

más exacta cuanto mayor es N . De aquí se deduce entonces que una acertada definición para el producto interno de L^2 en $L^2([a, b])$ esta dada por la ecuación 2.16 dada a continuación:

$$\langle f, g \rangle_{L^2} = \int_a^b f(t) \overline{g(t)} dt \quad \text{para } f, g \in L^2([a, b]) \quad (2.16)$$

Las propiedades de simetría conjugada, homogeneidad y bilinealidad son cumplidas completamente por este producto interno.

Mientras que para la condición de positividad, si $0 = \langle f, f \rangle = \int_a^b |f(t)|^2 dt$ y si f es

continua, entonces $f(t) = 0$ para todo t . Si ahora a $f(t)$ se le permite que sea discontinua en un número finito de puntos, entonces se puede concluir que $f(t) = 0$ en todo valor, excepto en un número finito de valores de t . Por ejemplo la función:

$$f(t) = \begin{cases} 1 & \text{si } t = 0 \\ 0 & \text{si } t \neq 0 \end{cases}$$

en esta función diferente de cero, aun se cumple que $\int_{-1}^1 |f(t)|^2 dt = 0$. No obstante, anteriormente se estipuló que dos elementos f y g en $L^2([a, b])$ son iguales si $f(t) = g(t)$ para todo valor de t excepto para un número finito de valores t . Esta es una definición muy razonable para el propósito de la integración ya que $\int_a^b f(t)dt = \int_a^b g(t)dt$ para dichas funciones. Con esta convención, la condición de positividad se sigue cumpliendo.

Lo equivalencia anteriormente establecida tiene mucho sentido, ya que desde el punto de vista del análisis de señales. El comportamiento de una señal en un instante de tiempo (digamos $t = 0$) raras veces es de interés. Mientras que el comportamiento de una señal a lo largo de un intervalo de tiempo positivo siempre es de importancia.

En muchas aplicaciones, la señal de interés ya es discreta. Por ejemplo, la señal que se obtiene desde un reproductor de discos compactos puede ser representada por un conjunto de números discretos que representan la intensidad de esta señal de sonido en un intervalo de tiempo regular. En estos casos, las señales pueden ser representadas como una secuencia tal que $X = \dots, x_{-1}, x_0, x_1, \dots$, donde cada x_j es el valor numérico de la señal en un j -ésimo intervalo de tiempo $[t_j, t_{j+1}]$. Teóricamente, la secuencia podría continuar indefinidamente. En realidad, la señal usualmente se detendrá en algún punto, lo cual matemáticamente puede ser representado por $x_j = 0$ para $|j| > N$ para algún entero N .

Para una señal discreta existe un espacio l^2 el cual describe un conjunto de todas las secuencias $X = \dots, x_{-1}, x_0, x_1, \dots$, $x_i \in C$, con $\sum_{n=-\infty}^{\infty} |x_n|^2 < \infty$. Y esta definido por la ecuación 2.17,

$$\langle X, Y \rangle_{l^2} = \sum_{n=-\infty}^{\infty} x_n \overline{y_n} \quad (2.17)$$

para $X = \dots, x_{-1}, x_0, x_1, \dots$, y $Y = \dots, y_{-1}, y_0, y_1, \dots$

2.4 La Wavelet Haar

Existen dos funciones que juegan un papel importante en el análisis de Wavelets, la función *escala* ϕ (wavelet padre) y la función *wavelet* ψ (wavelet madre). Estas dos funciones generan una familia de funciones que pueden ser utilizadas para reconstruir una señal.

El más simple de los análisis wavelet esta basado en la *función de escala Haar*, cuya representación gráfica se muestra en la figura 2.3.

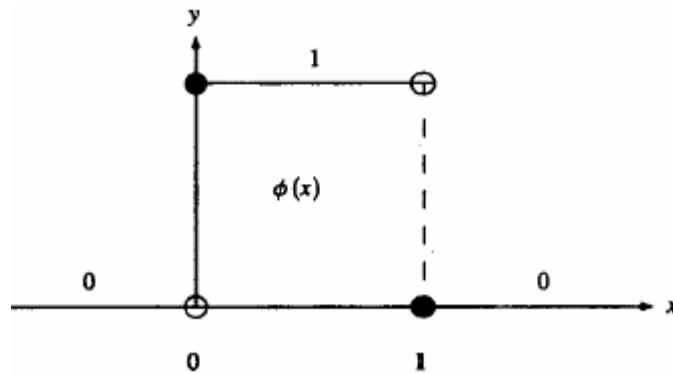


Fig. 2.3: Gráfica de la función escala Haar

Con el fin de ilustrar la idea básica de este tipo de análisis, consideraremos la forma de onda mostrada en la figura 2.4.

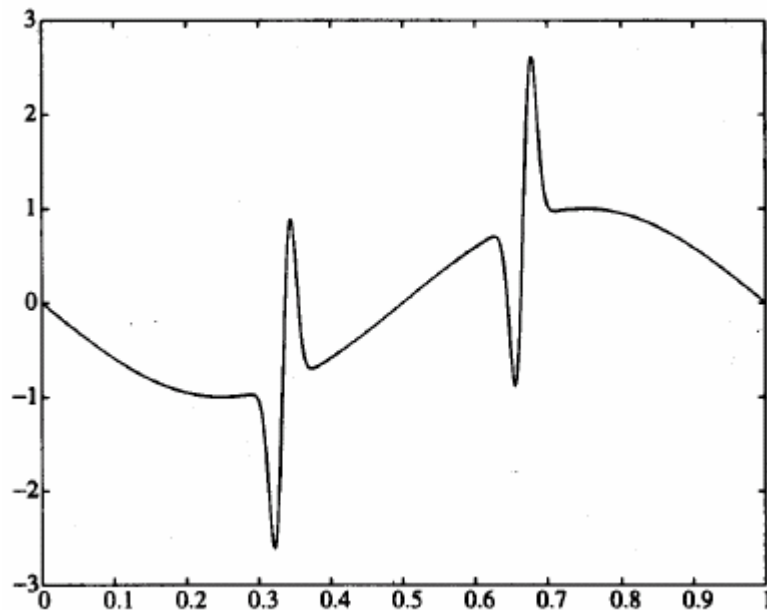


Fig. 2.4: Señal de voltaje de línea con sobreimpulsos

Dicha señal puede ser interpretada como la medida de alguna cantidad física (quizá un ciclo del voltaje de una línea) en función del tiempo. Los dos sobre impulsos que se observan en la figura 2.4 podrían representar ruido eléctrico originado por una pérdida de conexión durante la medición del voltaje.

En la figura 2.5 se puede observar una posible aproximación de la señal de interés utilizando la función de escala Haar. Los escalones generados por la función escala Haar son muy simples e ilustran la idea general que implica un análisis multiresolución, el cual se discutirá en la sección 2.6 con mayor detalle. La principal desventaja de la Wavelet Haar es que ella es discontinua, por lo que no aproxima muy bien señales continuas, es por ello que la señal de la figura 2.5 no aproxima con mucha exactitud a la señal de la figura 2.4. En la sección 2.8 se presentarán otros tipos de Wavelets, las cuales difieren del comportamiento de la Wavelet Haar.

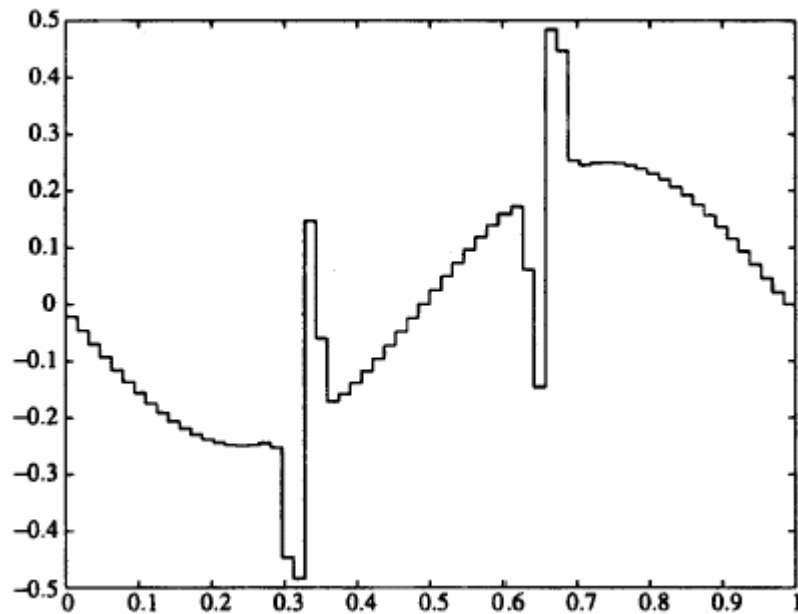


Fig. 2.5: Aproximación de la señal de voltaje de línea con sobre impulsos, utilizando la wavelet Haar

2.4.1 Función Haar escala [2]

La función escala Haar puede ser definida matemáticamente por la expresión de la ecuación 2.18.

$$\phi(x) = \begin{cases} 1, & \text{si } 0 \leq x < 1 \\ 0, & \text{en otro caso} \end{cases} \quad (2.18)$$

La ecuación anterior concuerda completamente con la gráfica que se observa en la figura 2.3, para una función escala Haar.

La función $\phi(x-k)$ representa la misma gráfica que ϕ pero trasladada k unidades a la derecha (asumiendo que k es positiva). Si permitimos que V_0 sea el espacio de todas las funciones de la forma:

$$\sum_{k \in \mathbb{Z}} a_k \phi(x-k) \quad a_k \in \mathbb{R} \quad (2.19)$$

en donde k puede variar sobre cualquier conjunto finito de enteros positivos o negativos. Ya que $\phi(x-k)$ es discontinua en $x = k$ y en $x = k+1$, una descripción alternativa de V_0 es que este consiste de todas las funciones constantes cuyas discontinuidades están contenidas en el conjunto de los números enteros. Debido a que k varía sobre un conjunto finito, cada uno de los elementos de V_0 es cero fuera de los límites del conjunto. A este tipo de funciones se les conoce como funciones de *soporte compacto*. La gráfica de un elemento típico de V_0 se puede observar claramente en la siguiente figura.

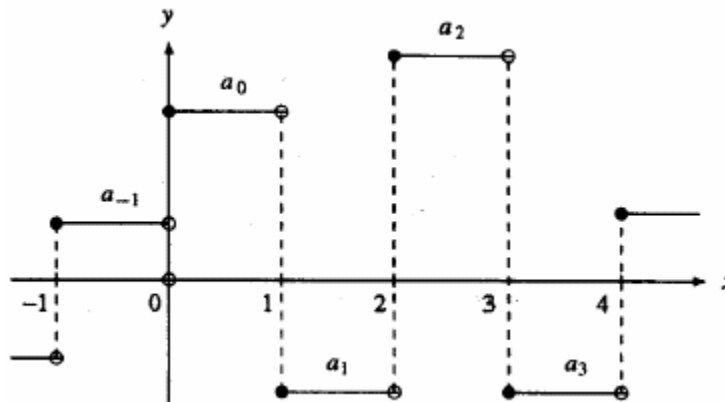


Fig. 2.6: Gráfica de un elemento típico en V_0

Es importante hacer notar que una función en el espacio V_0 puede no presentar discontinuidades en todo el conjunto de enteros (por ejemplo, si en la gráfica de la figura 2.6 hacemos $a_1 = a_2$, entonces la suma es continua en $x = 2$).

A partir de lo anteriormente establecido, y con el fin de tener una definición más general para la función de escala Haar, se puede establecer lo siguiente:

Supóngase que j es cualquier entero no negativo. El espacio de funciones escalón de nivel j , denotado por V_j , está definido por el espacio que abarca el conjunto

$$\left\{ \dots, \phi(2^j x + 1), \phi(2^j x), \phi(2^j x - 1), \phi(2^j x - 2), \dots \right\}$$

y que pertenecen a los números reales. V_j es el espacio de funciones discretas y constantes de soporte compacto cuyas discontinuidades pertenecen al conjunto:

$$\left\{ \dots, \frac{-1}{2^j}, 0, \frac{1}{2^j}, \frac{2}{2^j}, \frac{3}{2^j}, \dots \right\}$$

Una función en V_0 es una función discreta y constante con discontinuidades que pertenecen al conjunto de números enteros. Así pues, cualquier función que pertenece a V_0 también pertenece a V_1 , el cual consiste de funciones discretas y constantes cuyas discontinuidades pertenecen al conjunto $\{\dots, -1/2, 0, 1/2, 1, 3/2, \dots\}$. El mismo análisis aplica para V_1, V_2 y así sucesivamente:

$$V_0 \subset V_1 \subset \dots V_{j-1} \subset V_j \subset V_{j+1} \dots \quad (2.20)$$

Estas relaciones de inclusión son de carácter estricto. Así pues, por ejemplo, la función $\phi(2x)$ pertenece a V_1 pero no pertenece a V_0 (ya que $\phi(2x)$ es discontinua en $x = 1/2$).

Hay que tener en mente que V_j contiene toda la información relevante hasta una escala de resolución del orden de 2^j . Debido a que j es muy grande, la resolución es bastante fina. El hecho de que $V_j \subset V_{j+1}$ implica que no se pierde información cuando la resolución es más fina. Dicha relación de inclusión es también la razón por la cual V_j está definido en términos de $\phi(2^j x)$ en lugar de estar definida como $\phi(ax)$ para algún otro factor a . Es decir, si por ejemplo, se define a V_2 a través de $\phi(3x-j)$ en lugar de $\phi(4x-j)$, entonces V_2 no puede

incluir a V_l (ya que el conjunto de múltiplos de $1/2$ no esta incluido en el conjunto de múltiplos de $1/3$).

Como consecuencia de lo anteriormente establecido, se desprende el siguiente teorema, el cual establece que:

- Una función $f(x)$ pertenece a V_0 si y solo si $f(2^j x)$ pertenece a V_j .
- Una función $f(x)$ pertenece a V_j si y solo si $f(2^j x)$ pertenece a V_0 .

Con el fin de demostrar la primera parte del teorema anterior, diremos que si una función f pertenece a V_0 , entonces $f(x)$ es una combinación lineal de $\{\phi(x-k), k \in Z\}$. Por lo tanto, $f(2^j x)$ es una combinación lineal de $\{\phi(2^j x - k), k \in Z\}$, lo cual implica que $f(2^j x)$ es un miembro de V_j . La segunda parte de este teorema puede ser demostrado de la misma forma.

El gráfico de la función $\phi(2^j x)$ es un impulso de ancho $1/2^j$. Cuando j es grande, el gráfico de $\phi(2^j x)$ es similar a uno de los impulsos de una señal que se desea filtrar. Esto es deseable para tener un algoritmo que descomponga una señal en sus componentes V_j muy eficiente. Una forma de desarrollar dicha descomposición eficientemente es construir una base ortonormal para V_j . (utilizando el producto interno L^2). Tal como se define en la ecuación 2.21.

$$\|\phi(x-k)\|_{L^2}^2 = \int_{-\infty}^{\infty} \phi(x-k)^2 dx = \int_k^{k+1} 1 dx = 1 \quad (2.21)$$

Si j es diferente de k , entonces $\phi(x-j)$ y $\phi(x-k)$ tienen soporte acotado, tal como puede observarse en la figura 2.7. Por lo tanto se cumple que:

$$\langle \phi(x-j), \phi(x-k) \rangle_{L^2} = \int_{-\infty}^{\infty} \phi(x-j)\phi(x-k) dx = 0 \quad j \neq k \quad (2.22)$$

y entonces el conjunto $\{\phi(2^j x - k), k \in Z\}$ es una base ortonormal para V_0 .

El argumento anterior puede extenderse a un caso más general, donde el conjunto de funciones $\{2^{j/2}\phi(2^j x - k), k \in Z\}$ es una base ortonormal de V_j . (El factor $2^{j/2}$ esta presente, ya que: $\int_{-\infty}^{\infty} (\phi(2^j x))^2 dx = \frac{1}{2^j}$).

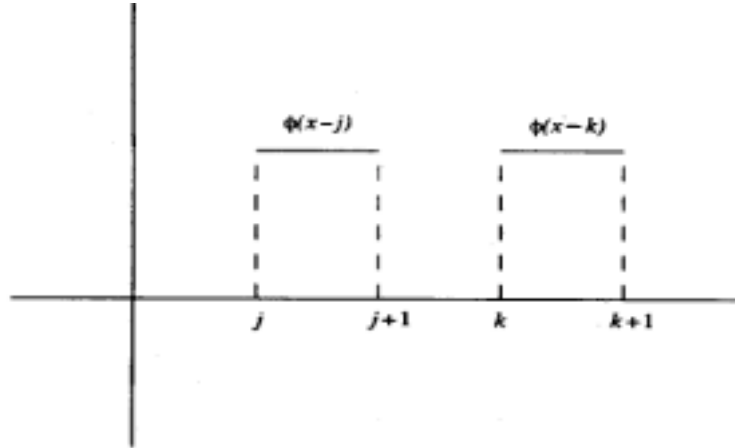


Fig. 2.7: Representación gráfica de $\phi(x-j)$ y $\phi(x-k)$.

2.4.2 Función Haar Wavelet [2]

El obtener la base ortonormal de V_j es solo la mitad del trabajo. Con el fin de resolver nuestro problema de filtrado de ruido, necesitamos conocer una forma de aislar los impulsos que pertenecen a V_j pero que no son miembros de V_{j-1} . Aquí es donde la función Wavelet ψ entra en acción.

La idea es descomponer V_j como una sumatoria ortogonal de V_{j-1} y su complemento. De nuevo, iniciaremos haciendo $j = 1$ e identificaremos el complemento ortogonal de V_0 en V_1 . Dos factores claves necesarios para construir la función Wavelet ψ son:

1. ψ es miembro de V_1 y entonces ψ puede ser expresado como $\psi(x) = \sum_l a_l \phi(2x - l)$ para algunas opciones de $a_l \in R$.
2. ψ es ortogonal a V_0 . Esto es equivalente a $\int \psi(x)\phi(x - k)dx = 0$ para todos los enteros k .

El primero de los requerimientos indica que ψ es construida a partir de una función cuadrada de ancho 1/2. Mientras que el segundo requerimiento en el cual se hace $k = 0$,

implica que $\int_{-\infty}^{\infty} \psi(x)\phi(x)dx = 0$. La función Wavelet (ψ) más simple que satisface ambos requerimientos es la función cuya gráfica se muestra en la figura 2.8. Y que puede ser descrita por la ecuación 2.23,

$$\psi(x) = \phi(2x) - \phi(2(x - 1/2)) = \phi(2x) - \phi(2x - 1) \quad (2.23)$$

con lo cual se satisface el primer requerimiento. Y además se cumple lo establecido por la ecuación 2.24.

$$\int_{-\infty}^{\infty} \psi(x)\phi(x)dx = \int_0^{1/2} 1dx - \int_{1/2}^1 1dx = 1/2 - 1/2 = 0 \quad (2.24)$$

Lo anterior indica que ψ es ortogonal a ϕ . Si $k \neq 0$, entonces el soporte de $\psi(x)$ y el soporte de $\phi(x-k)$ no se traslapan y entonces $\int \psi(x)\phi(x-k)dx = 0$. Por lo tanto, ψ pertenece a V_1 y es ortogonal a V_0 .

Luego de haber presentado el análisis anterior, se puede decir que una función Wavelet Haar puede ser expresada matemáticamente por la expresión que se observa en la ecuación 2.25.

$$\psi(x) = \phi(2x) - \phi(2x - 1) \quad (2.25)$$

Cuya representación gráfica puede observarse en la figura 2.8.

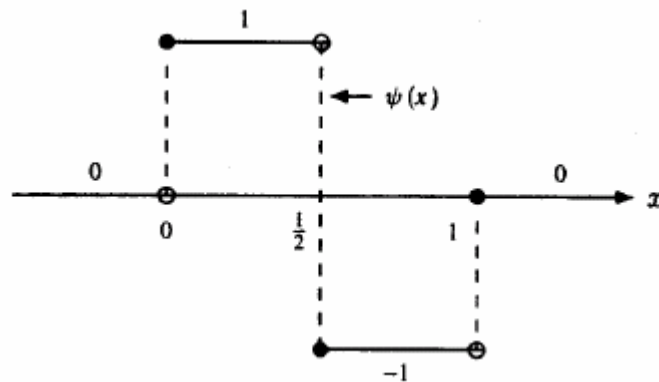


Fig. 2.8: Representación gráfica de la función Wavelet Haar $\psi(x)$

2.5 Análisis Multi-resolución

Aunque los problemas de resolución en tiempo y frecuencia son resultado del principio de indeterminación de Heisenberg y existen independientemente de la transformada utilizada, es posible analizar una señal mediante un enfoque alternativo denominado *análisis multi-resolución*.

Antes de definir los principios que rigen un análisis multi-resolución, necesitamos referirnos al Teorema del Muestreo. La utilidad de dicho teorema radica en que se obtiene una reconstrucción aproximada de una señal f a partir de muestras tomadas de dicha señal uniformemente espaciadas a intervalos de largo T . Si la señal es de banda limitada y su frecuencia de Nyquist es menor que $1/T$, entonces la reconstrucción es perfecta, de otra forma solamente es una aproximación, es decir, que el tamaño de T mide la *resolución* de la señal de interés f . Así pues, un típico análisis de la Transformada Rápida de Fourier (FFT), tomando muestras de la señal f trabaja en *una resolución*, T .

Ahora bien, si la señal f presenta intervalos en los cuales varía rápida y repentinamente, y a la vez períodos donde su variación es muy lenta, el tipo de análisis en una resolución que se describió anteriormente no trabaja bien (debido a las razones descritas en la sección 2.2). Para poder llevar a cabo el análisis de dichas señales, Mallat tuvo la idea de hacer dos cosas. Primero, remplazar el espacio de las funciones de banda limitada obtenidas del teorema del muestreo con una adaptación a la señal. Segundo, analizar la señal utilizando versiones escaladas del mismo espacio, pero ajustadas a resoluciones de $T/2$, $T/2^2$, y así sucesivamente (de aquí el término *análisis multi-resolución*) [1], [2].

El análisis multi-resolución analiza la señal con diferentes resoluciones en diferentes frecuencias. A diferencia de la Transformada de Período Corto de Fourier, cada una de las componentes espectrales se trata de diferente manera.

Además, el análisis multi-resolución está diseñado para brindar una buena resolución temporal y una resolución en frecuencias pobre para las frecuencias altas y una buena resolución en frecuencias junto a una resolución temporal pobre para las frecuencias bajas

de una señal de interés. Este enfoque tiene especial sentido cuando la señal a analizar tiene componentes frecuenciales altas durante períodos de tiempo cortos y componentes frecuenciales bajas durante períodos de tiempo largos.

En las secciones anteriores se presentaron varios principios de multi-resolución, como condiciones para una función escala. Y que a continuación se presentaran como requerimientos básicos del análisis multi-resolución. Si en primer lugar, permitimos que se cumpla que $V_j, j = \dots, -2, -1, 0, 1, 2, \dots$ sea una secuencia de sub-espacios de funciones en $L^2(\mathbb{R})$. El conjunto de espacios $\{V_j, j \in \mathbb{Z}\}$ es llamado análisis multi-resolución con función de escala ϕ si las siguientes condiciones se cumplen:

1. Anidamiento, $V_j \subset V_{j+1}$. Lo cual implica que un análisis multi-resolución requiere un anidamiento de los espacios generados por las funciones escala.
2. Densidad, $\overline{\cup V_j} = L^2(\mathbb{R})$
3. Separación, $\cap V_j = \{0\}$
4. Escalamiento, la función $f(x)$ pertenece a V_j si y sólo si la función $f(2^j x)$ pertenece a V_0 .
5. Base ortonormal, la función ϕ pertenece a V_0 y el conjunto $\{\phi(x - k), k \in \mathbb{Z}\}$ es una base ortonormal (utilizando el producto interno L^2) de V_0 .

Los valores de V_j son llamados *espacios de aproximación*. Pueden existir muchas opciones para la elección de ϕ , correspondiente a un sistema de espacios de aproximación. Elecciones diferentes de la función ϕ pueden conducir a diferentes análisis multi-resolución [1], [2].

Probablemente el tipo más útil de funciones escala son aquellas que tienen soporte compacto. Como ya se mencionó anteriormente, una función presenta soporte compacto si tiene un valor de 0 (cero) fuera de un intervalo finito. La función de escala Haar es un buen ejemplo de una función de soporte compacto. Las funciones de escala asociadas con las Wavelets Daubechies no solamente presentan soporte compacto, sino que también son continuas. Tener estas dos propiedades en una función escala es muy deseable, porque los

algoritmos de descomposición y reconstrucción asociados, son computacionalmente más rápidos y realizan su trabajo de análisis y reconstrucción de señales de una manera más eficiente [2].

2.6 La transformada Wavelet

La transformada Wavelet constituye una técnica relativamente nueva que ha sido utilizada en los últimos años como una muy poderosa herramienta en el análisis del comportamiento local (puntual) de una señal. Dicha transformada fue desarrollada como una alternativa a los problemas de resolución tiempo-frecuencia intrínsecos que presenta la Transformada de Fourier de Período Corto (STFT).

Al igual que la STFT, la transformada Wavelet hace uso de una función ventana que encuadra una señal dentro de un intervalo y focaliza el análisis únicamente en ese segmento de la señal. En principio, lo que hace dicha transformada es filtrar la señal de interés en el dominio del tiempo mediante filtros de paso bajo y de paso alto, los cuales eliminan ciertas componentes de frecuencias (altas o bajas según corresponda) de la señal, luego, el proceso se repite para las señales resultantes del procedimiento de filtrado anterior. Por ejemplo, supóngase que se tiene una señal con componentes de frecuencia de hasta 1 KHz, en la primera etapa de filtrado la señal es dividida en dos partes haciéndola pasar a través de un filtro de paso bajo y un filtro paso alto con lo cual se obtienen dos versiones diferentes de la misma señal: una de dichas versiones corresponde a las frecuencias de 0 – 500 Hz (filtro paso bajo), y otra versión que corresponde a las frecuencias entre 500 – 1,000 Hz (filtro paso alto). A continuación, se toma cualquiera de estas dos versiones resultantes y se realiza nuevamente el mismo procedimiento, al cual se le denomina *descomposición*.

En la mayor parte de los análisis de este tipo, la versión de la señal que se utiliza es la parte que corresponde al filtro de paso bajo, con lo que, se obtendrían tres conjuntos de datos; correspondiente cada uno de ellos a la señal original pero a distintas frecuencias: 0 – 250 Hz, 250 – 500 Hz y 500 – 1,000 Hz. A continuación se vuelve a tomar la señal correspondiente a la parte del filtrado de paso bajo haciéndola pasar nuevamente por los filtros paso bajo y paso alto, de esta forma se obtendrían 4 conjuntos de señales

correspondientes a las frecuencias 0 – 125 Hz, 125 – 250 Hz, 250 – 500 Hz y 500 – 1,000 Hz. El proceso continua hasta que la señal se ha descompuesto en un cierto número de niveles predefinidos. Y finalmente se cuenta con un grupo de señales que representan la misma señal, pero correspondientes a diferentes bandas de frecuencia. Para cada una de estas bandas se conocen sus respectivas señales, si se juntan todas y se presentan en una gráfica tridimensional, se tendría *tiempo* en un eje, *frecuencia* en el segundo y *amplitud* en el tercer eje. De esta manera, es posible establecer que frecuencias existen para un tiempo dado. Sin embargo, el *principio de indeterminación de Heisenberg*, tal y como se explicó en la sección 1.4 del presente documento, establece que no puede conocerse la información de tiempo y frecuencia de una señal en un cierto punto del plano tiempo-frecuencia, es decir, no pueden determinarse con exactitud que frecuencias existen en un instante dado, por lo que solamente es posible conocer que bandas de frecuencias existen en un determinado y bien definido intervalo de tiempo. Esto, como ya se comentó, es un problema de resolución y es de las principales razones por las cuales existe una tendencia a reemplazar la STFT por la Transformada Wavelet (WT), ya que la STFT trabaja con una resolución fija para todos los tiempos, mientras que la WT hace uso de una resolución variable.

Con la WT las altas componentes de frecuencia presentan una mejor resolución en el tiempo, mientras que las bajas frecuencias presentan mejor resolución en el dominio de la frecuencia. Esto implica que una determinada componente de alta frecuencia puede localizarse mejor en el tiempo (con menor error relativo) que una componente de baja frecuencia. Por el contrario, una componente de baja frecuencia puede localizarse mejor en frecuencia comparado con una componente de alta frecuencia.

Con el fin de brindar una mejor interpretación de la resolución en el tiempo y la frecuencia, en la figura 2.9 (a) puede observarse que a altas frecuencias (fila superior) la cantidad de puntos es mayor para un mismo intervalo de tiempo (ΔT); es decir, las altas frecuencias tienen una mejor resolución en el tiempo. Sin embargo, a bajas frecuencias para el mismo intervalo de tiempo existen menos puntos que caracterizan la señal, de lo que se concluye que las frecuencias bajas no tienen buena resolución en el tiempo.

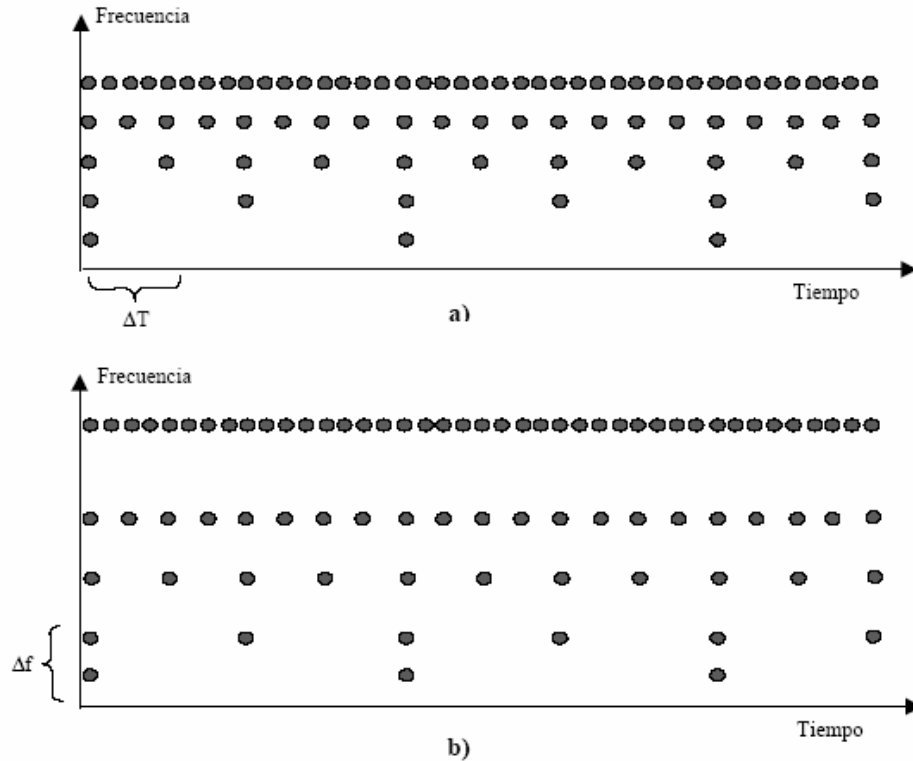


Fig. 2.9: Interpretación gráfica de la resolución en el tiempo y la frecuencia de la WT

Por otra parte, para el caso de una señal discretizada en el tiempo, la resolución en el tiempo puede interpretarse de manera similar a lo comentado en el caso a), pero ahora la información en frecuencia tiene diferentes resoluciones en cada escalón de descomposición tal como puede observarse en la figura 2.9 b). Así pues, puede observarse que para un (Δf) determinado, la resolución en el tiempo es mejor para las bajas frecuencias que para las altas frecuencias, puesto que la separación entre cada escalón de descomposición aumenta a medida que se incrementa la frecuencia.

2.6.1 Variables de escala (*a*) y traslación (*b*)

En el análisis de la Transformada Wavelet, el parámetro *escala* (*a*) es análogo con el parámetro *escala* utilizado en los mapas topográficos. Así pues, las altas escalas corresponden a una visión global no detallada de la señal y las bajas escalas corresponden a una “vista” detallada. De igual forma, en términos de frecuencia, las bajas frecuencias (altas escalas) corresponden a una información global de la señal que comúnmente abarca toda la

señal, mientras que las altas frecuencias (escalas bajas) a una información detallada de una característica oculta en la señal que comúnmente dura un tiempo relativamente pequeño. El escalamiento como operación matemática produce una dilatación ($|a| > 1$) ó una compresión ($|a| < 1$) de una señal. Lo cual nos da el grado de resolución con el cual estamos analizando la señal de interés.

Por su parte, el término *traslación* (b) es utilizado con el mismo significado que tiene en el análisis de la STFT y se relaciona con la localización de la ventana a medida que ésta se desplaza a través de la señal. Es de hacer notar que este término corresponde a la información del tiempo en el dominio transformado. Un punto importante es que la función wavelet ψ (función ventana en STFT) se traslada cubriendo toda la señal para cada valor de a , es decir, si la escala escogida es pequeña habrán más traslaciones de ψ que si la escala escogida es grande; este comportamiento se representa en la figura 2.10. En dicha figura se puede observar con mayor detalle que para una escala grande la wavelet ocupa un mayor segmento de la señal y por lo tanto tiene mejor resolución en frecuencia mientras que para una escala más pequeña el intervalo de tiempo bajo el que se analiza la señal es menor, lo que implica mayor resolución en el tiempo.

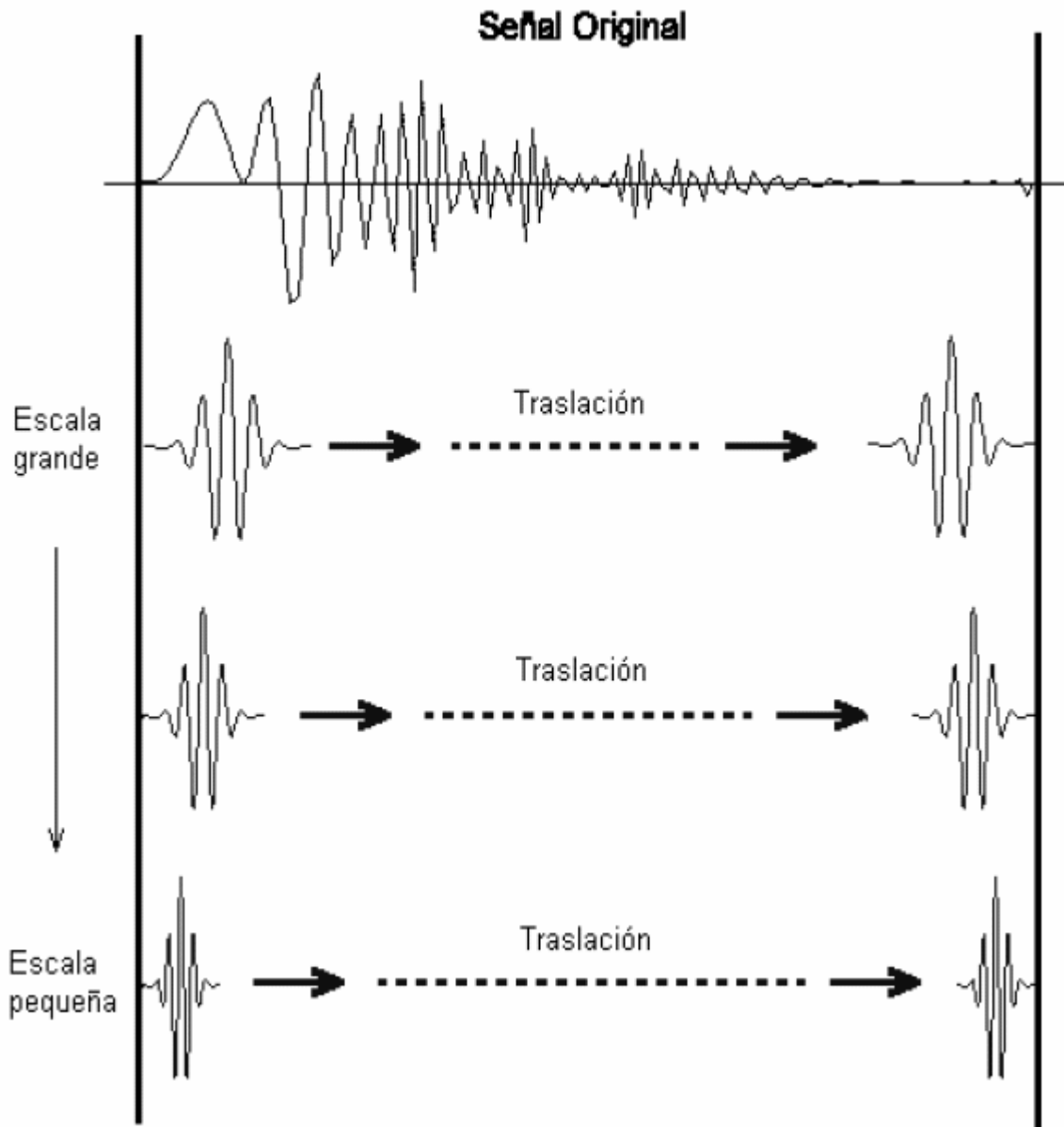


Fig. 2.10: Proceso de escalamiento y traslación de una señal

En resumen, podemos decir que existen dos principales diferencias entre la Transformada Wavelet (WT) y la Transformada de Fourier de Período Corto (STFT):

- La transformada de Fourier de las señales objeto del estudio no son calculadas.
- La anchura de la ventana se varía conforme la transformada Wavelet se calcula para cada componente espectral.

Luego de haber definido los principales componentes a tomar en cuenta a la hora de realizar el cálculo de la Transformada Wavelet, en las siguientes dos secciones se estudiarán con mayor detalle los dos tipos de WT que tienen mayor importancia en el procesamiento de señales: *la transformada wavelet continua y la transformada wavelet discreta* [17].

2.6.2 Transformada Wavelet Continua

La Transformada Wavelet Continua (CWT) pretende expresar una señal $f(t)$ continua en el dominio del tiempo, mediante una expansión de coeficientes proporcionales al producto interno entre la señal de interés $f(t)$ y diversas versiones escaladas y trasladadas de la función *wavelet madre* $\psi(t)$. Si se establece además, que ambas funciones son de energía finita, entonces se puede definir matemáticamente la CWT por la ecuación 2.26 que se muestra a continuación:

$$CWT(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-b}{a}\right) dt \quad (2.26)$$

Si a la ecuación 2.26 le aplicamos el Teorema de Parseval dicha ecuación puede expresarse de la forma descrita en la ecuación 2.27 en términos de la transformada de Fourier de $f(t)$ y de $\psi(t)$, así:

$$CWT(a, b) = \frac{1}{2\pi\sqrt{a}} \int_{-\infty}^{\infty} F(\omega) \Psi(a\omega) e^{-j\omega b} d\omega \quad (2.27)$$

Tanto en la ecuación 2.26 como en la ecuación 2.27 se pueden observar dos nuevas variables (a y b) las cuales fueron definidas anteriormente en la sección 2.6.1. Así pues, la variable a controla el ancho de la función $\psi(t)$, y la variable b nos indica la ubicación en el dominio del tiempo de $\psi(t)$.

Sin embargo, para que este análisis sea posible y lograr a su vez una perfecta reconstrucción de la señal a partir de la transformada, la función $\psi(t)$ debe cumplir con

cierto criterio denominado *condición de admisibilidad* la cual queda definida por la ecuación 2.28.

$$\begin{aligned}\int \psi(t)dt &= 0 \\ \Psi(0) &= 0\end{aligned}\tag{2.28}$$

El cumplimiento de la condición de admisibilidad (ecuación 2.28) implica que el valor medio de $\psi(t)$ es igual a cero, lo que a su vez establece que $\psi(t)$ debe tener valores positivos y negativos. Además, como $\psi(t)$ es una función que ventaniza la señal de interés sobre un intervalo de tiempo dado por a alrededor de un punto $t = b$ se establece entonces que $\psi(t)$ debe ser de soporte compacto, es decir, $\psi(t)$ es una onda definida sobre un intervalo de tiempo finito. Por otra parte, el hecho que se cumpla la ecuación 2.28 significa implícitamente que $\Psi(\omega)$ debe presentar un rápido decaimiento cuando ω tiende a cero, lo que hace suponer entonces que $\psi(t)$ es una función ventana pasa banda en el dominio de la frecuencia (ya que al menos en la frecuencia 0 se detiene).

Desde un punto de vista intuitivo, la CWT consiste en calcular un índice de semejanza entre la señal que esta siendo analizada y la wavelet madre, tal como se muestra en la figura 2.11.

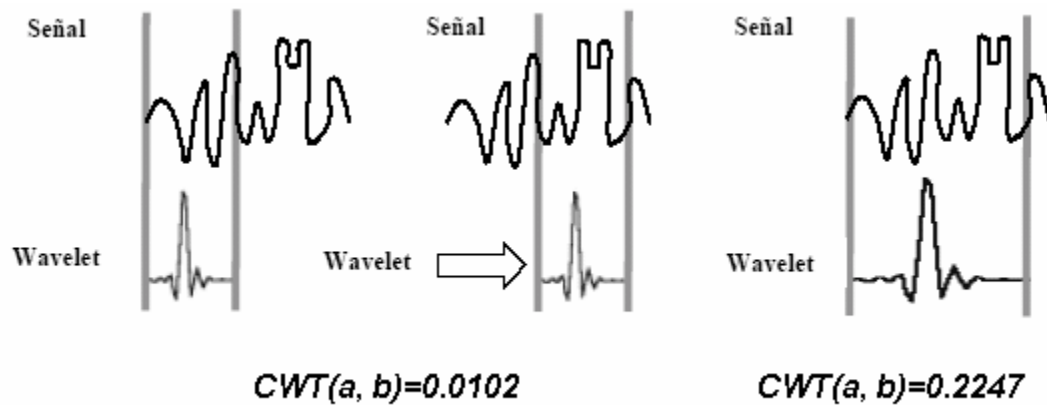


Fig. 2.11: Procedimiento de cálculo de la CWT

El proceso de cálculo de la CWT de una señal en particular puede ser descrito por los siguientes cuatro pasos, los cuales también se ven ilustrados en la figura 2.11 [5], [17].

1. Seleccionar la Wavelet Madre adecuada.
2. Dado los valores de a y b , calcular un coeficiente $CWT(a, b)$ haciendo uso de la ecuación 2.26, que represente la correlación entre la wavelet y la sección de la señal bajo análisis. Cuanto mayor sea el valor de dicho coeficiente, mayor será la similitud entre ambas señales, de lo cual es importante hacer notar que los resultados obtenidos dependerán en gran medida de la forma de la wavelet madre seleccionada en el paso 1.
3. Desplazar la wavelet madre en el sentido positivo del eje del tiempo y repetir los pasos anteriores hasta que se haya cubierto la totalidad de la señal.
4. Escalar la wavelet madre en el tiempo, y repetir los pasos del 1 al 3.

La CWT es una Transformada reversible siempre que se satisfaga la condición de admisibilidad (ecuación 2.28). El cual no es un requerimiento demasiado restrictivo, ya que existen muchas funciones que pueden ser utilizadas como ondas madre y cuya integral es cero. Para ello la función debe ser de carácter oscilatorio.

Así pues, la reconstrucción en el dominio del tiempo de una señal, si se conoce la Transformada Wavelet Continua se logra a través de la ecuación 2.29.

$$f(t) = \frac{1}{c_\psi^2} \int \int CWT(a, b) \frac{1}{a^2} \psi \left(\frac{t-b}{a} \right) db da \quad (2.29)$$

En donde c_ψ es una constante que depende de la onda madre que se utilice y cuyo éxito en la reconstrucción de la señal depende de que esta constante, llamada *constante de admisibilidad*, satisfaga la condición expresada en la ecuación 2.30.

$$c_\psi = \sqrt{2\pi \int_{-\infty}^{\infty} \frac{|\Psi(\xi)|^2}{|\xi|} d\xi} < \infty \quad (2.30)$$

En donde $\Psi(\xi)$ es la Transformada de Fourier de $\psi(t)$. La ecuación 2.30 implica el cumplimiento de las igualdades expresadas en la ecuación 2.28 [13].

2.6.3 Transformada Wavelet Discreta

La principal idea detrás de la Transformada Wavelet Discreta (DWT) en esencia es la misma que en la CWT. Pero en este caso, se hace uso de filtros digitales para obtener una representación tiempo-escala (frecuencia) de una señal digital (discreta). La CWT tal y como se describió en la sección 2.6.2, se calcula variando la escala de la ventana del análisis, desplazando la ventana en el tiempo, multiplicando por la señal de interés e integrando sobre todo el tiempo.

En el caso discreto, se hace uso de filtros con diferentes frecuencias de corte para analizar la señal a diferentes escalas. La señal se pasa por una serie de filtros de paso alto para analizar las frecuencias altas y por una serie de filtros de paso bajo para analizar las bajas frecuencias. La resolución de la señal se cambia mediante operaciones de filtrado y la escala se cambia con operaciones de *submuestreo* y *supermuestreo*.

Submuestrear una señal significa reducir la tasa de muestreo, eliminando algunas muestras de la señal. Mientras que *supermuestrear* una señal es incrementar la tasa de muestreo de la señal, añadiéndole nuevas muestras, normalmente uno, cero o u valor interpolado [1].

Luego de esta pequeña aclaración, se puede establecer el funcionamiento de la DWT sobre una señal discreta $x(n)$ la cual se hace pasar a través de un filtro de media banda de paso alto $g(n)$ y por uno de paso bajo $h(n)$. El filtrado de una secuencia por un filtro digital de respuesta impulsiva $h(n)$ es expresado por la operación de convolución entre ambas señales tal y como se puede ver expresado en la ecuación 2.31 que se muestra a continuación:

$$\begin{aligned}y(n) &= x(n) * h(n) \\ &= \sum_k x(k)h(n - k) \\ &= \sum_k h(k)x(n - k)\end{aligned}\tag{2.31}$$

En donde $y(n)$ es la secuencia a la salida del filtro.

Luego del filtrado pueden eliminarse la mitad de las muestras según el criterio de Nyquist, ya que la máxima frecuencia de la señal (frecuencia de corte) es ahora $\pi/2$ [rad/s] y no π [rad/s]. Puede por tanto, submuestrearse la señal por dos tomando una muestra si y otra no.

La descomposición anterior reduce a la mitad la resolución en el dominio del tiempo, ya que la señal completa esta caracterizada ahora por la mitad de las muestras. Sin embargo, la resolución en frecuencias se ha duplicado, ya que el ancho de banda de la señal es la mitad del ancho de banda de la señal original.

Al procedimiento descrito en el párrafo anterior se le conoce como *codificación subbanda*, el cual se repite sucesivamente a la salida del filtro de paso bajo hasta obtener una secuencia de longitud dos. En la figura 2.12 se ilustran los primeros pasos de dicho procedimiento.

Finalmente, la DWT de la señal de interés se obtiene concatenando todos los coeficientes desde el primer nivel de descomposición. La DWT tendrá el mismo número de coeficientes que la señal original.

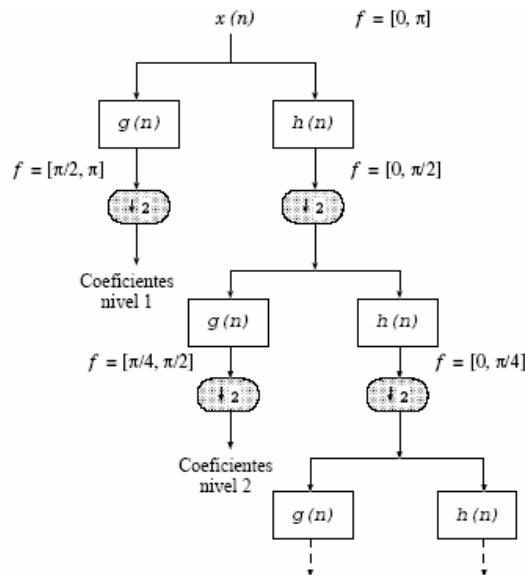


Fig. 2.12: Algoritmo de codificación sub-banda

Las frecuencias más destacadas de la señal original aparecerán con amplitudes grandes en la región de la transformada que incluya esas frecuencias en particular. La diferencia entre esta transformada y la transformada discreta de Fourier es que no se ha perdido la ubicación temporal de dichas frecuencias. Sin embargo, la resolución de esta localización temporal dependerá del nivel en que aparezca la frecuencia. Si la información principal de la señal aparece en las frecuencias altas, como ocurre frecuentemente, la localización temporal de estas frecuencias será más precisa, ya que vienen caracterizadas por un mayor número de muestras. Por otra parte, si la información principal de la señal aparece solo en las frecuencias bajas, su localización temporal no será muy precisa, ya que se utilizan pocas muestras para representar la señal en esas frecuencias. El procedimiento anteriormente descrito, en definitiva ofrece una buena resolución temporal en las frecuencias altas y una buena resolución en frecuencia en las frecuencias bajas. Como ya se menciono antes, este comportamiento es adecuado para el tratamiento de la mayor parte de señales que se encuentran en fenómenos reales.

Las bandas de frecuencias que no sean muy significativas en la señal original, tendrán amplitudes muy bajas y se podrá prescindir de esta sección del análisis de la DWT casi sin perder información.

Es de hacer notar, que debido a los sucesivos submuestreos por dos, la longitud de la señal debe ser potencia de dos, o al menos, un múltiplo de una potencia de dos, para que este esquema sea eficiente [17].

El diseño de una versión discreta de la Transformada Wavelet consiste principalmente indefinir un apropiado conjunto de parámetros $\{(a_j; b_{jk})\}$, de escalas y traslaciones. De forma tal que la familia de wavelets $\psi(a_j, b_{jk})$ cumpla con la *condición de admisibilidad*. Entre las Wavelets que satisfacen esta condición encontramos la *Spline*, la *Daubechies* y aquellas variantes que particularmente generan *bases ortonormales de Wavelets* [6].

En la mayoría de los casos, estas wavelets se asocian a la red diádica establecida a continuación por la ecuación 2.32.

$$\begin{aligned} a_j &= 2^{-j} \\ b_{jk} &= 2^{-jk} \end{aligned} \quad (2.32)$$

Teniendo en cuenta la expresión para las wavelets definida por la ecuación 2.33 y en base a los parámetros definidos en la ecuación 2.32, se obtiene la expresión de la Transformada Discreta Wavelet, mostrada en la ecuación 2.34, la cual relaciona la energía de la señal $f(t)$ y la Wavelet madre, para todos los valores enteros de j y k [6].

$$\psi_{jk}(t) = 2^{j/2} \psi(2^j t - k) \quad j, k \in Z \quad (2.33)$$

$$DWT(j, k) = \int_{-\infty}^{\infty} f(t) \psi_{jk}(t) dt \quad (2.34)$$

2.7 Familias de Wavelets [1]

A lo largo de las siguientes secciones se presentaran brevemente las principales familias de Wavelets que existen.

2.7.1 Wavelet Haar

Dentro de las Wavelets más famosas, antiguas y ampliamente utilizadas encontramos la *wavelet Haar* (la cual ya fue descrita con mayor profundidad en la sección 2.4), sin embargo en esta sección se describirá brevemente sus características. La wavelet Haar se define matemáticamente por la función descrita en la ecuación 2.35 que se presenta a continuación:

$$Haar(t) = \begin{cases} 1; & 0 \leq t < \frac{1}{2} \\ -1; & \frac{1}{2} \leq t < 1 \\ 0; & \text{otro valor} \end{cases} \quad (2.35)$$

y esta representada por la gráfica mostrada en la figura 2.13. En ella se puede observar que este tipo de wavelet es muy sencilla. Más sin embargo, es muy utilizada en el análisis de señales haciendo uso de las transformadas continua y discreta.

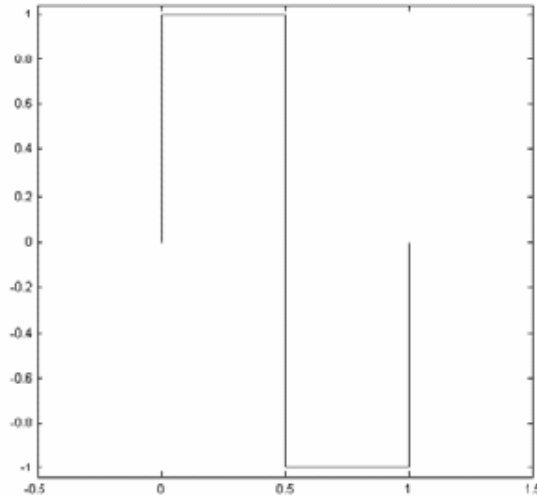


Fig. 2.13: Representación gráfica Wavelet Haar

Entre las características que sobresalen de esta wavelet tenemos:

- Es ortogonal.
- Presenta soporte compacto
- La función *escala Haar* es simétrica
- La función *wavelet Haar* no es simétrica
- Presenta un solo momento de desvanecimiento

De todas las wavelets que se presentarán aquí, la wavelet Haar es la única que es ortogonal, de soporte compacto y posee simetría.

2.7.2 Función del Sombrero Mejicano

Una de las funciones más utilizadas como wavelet madre en el cálculo de la transformada wavelet continua (CWT) es la función del *sombrero mejicano*; la cual obtiene su nombre de la forma que describe su gráfica que puede ser observada en la figura 2.14.

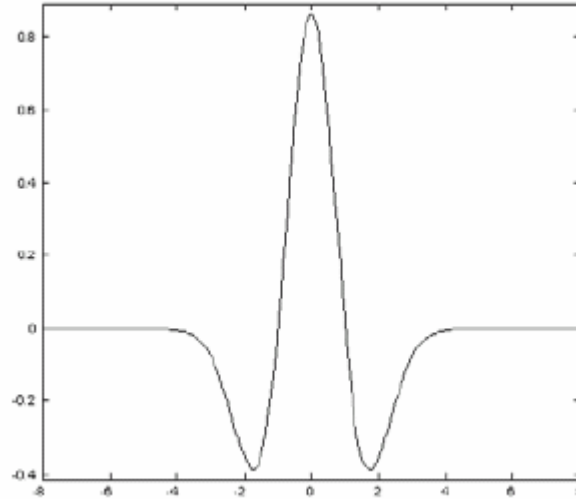


Fig. 2.14: Representación gráfica de Wavelet del Sombrero Mexicano

Esta wavelet es definida como la segunda derivada de la función de distribución Gaussiana y se representa por la ecuación 2.36, de esta forma:

$$Mexh(t) = \frac{2(1-t^2)e^{-\frac{t^2}{2}}}{\pi^{1/4}\sqrt{3}} \quad (2.36)$$

Entre las características de mayor importancia que esta wavelet presenta tenemos:

- Es simétrica, tal y como puede observarse en la figura 2.14. Ello le permite examinar a las señales de un modo simétrico y lineal en la fase, al igual que la wavelet Haar
- Satisface la condición de admisibilidad
- Presenta dos momentos de desvanecimiento

2.7.3 Wavelet Daubechies

La *Wavelet Daubechies* puede ser de orden N dependiendo del número de momentos de desvanecimiento que se requieran en un análisis en particular. N es un entero positivo y denota el número de coeficientes de filtro que tiene esa wavelet; así pues, la *Daubechies* de orden 1 (*db1*) es la wavelet Haar que se estudio en la sección 2.4 y se volvió a revisar brevemente en la sección 2.7.1.

En la figura 2.15 puede observarse la representación gráfica de una wavelet Daubechies de orden 5 (*db5*), donde el número de momentos de desvanecimiento es igual al orden de la wavelet. Además, es de hacer notar que existe una relación del doble entre la cantidad de los coeficientes del filtro y los momentos de desvanecimiento de la wavelet. Con este tipo de wavelet es posible realizar el cálculo de la Transformada Wavelet Continua y Discreta.

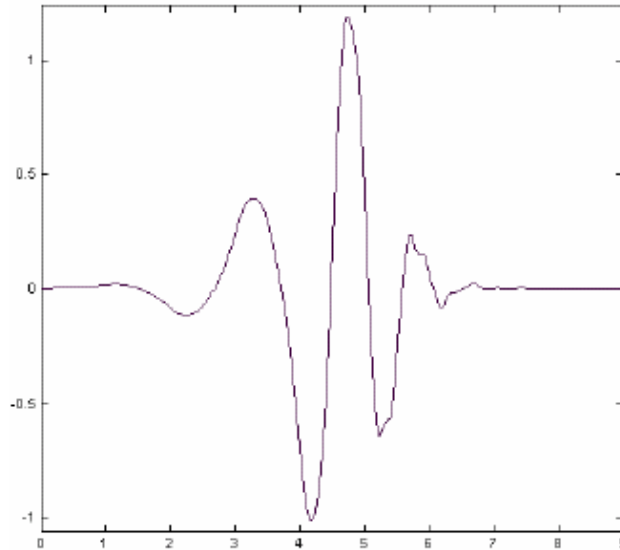


Fig. 2.15: Representación gráfica de Wavelet Daubechies 5

Resumiendo, se puede decir que la Wavelet Daubechies presenta las siguientes características:

- Es ortogonal y biortogonal
- Es de soporte compacto
- No existe simetría para $N > 1$
- Existen N momentos de desvanecimiento
- La longitud del filtro es igual a $2N$

2.7.4 Wavelet Symlets

En la figura 2.16 se puede observar la representación gráfica de la *Wavelet Symlets* de orden 2 (*sym2*). Con esta wavelet es posible realizar la transformada wavelet continua y discreta y posee diferentes órdenes N .

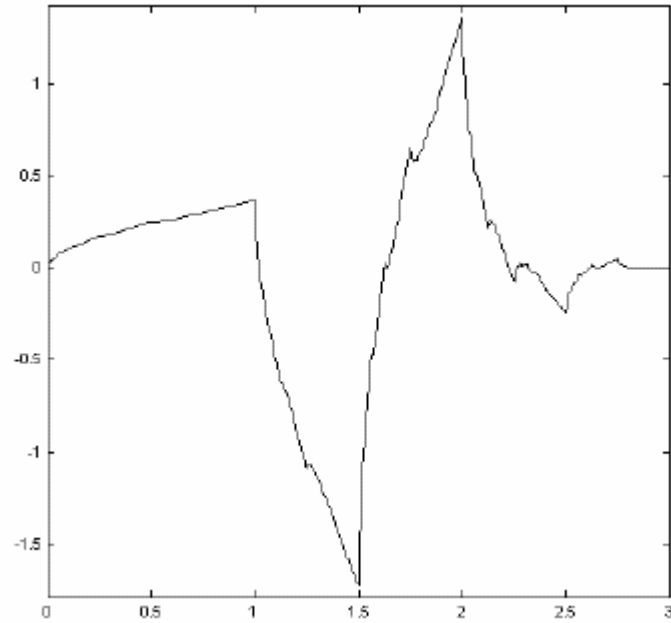


Fig. 2.16: Representación gráfica Wavelet Symlets 2

Entre las propiedades que caracterizan a la Wavelet Symlets podemos mencionar:

- Es ortogonal y biortogonal
- Presenta soporte compacto
- La longitud del filtro es $2N$
- La función escala (ϕ) Symlets es aproximadamente de fase lineal
- La función Wavelet (ψ) Symlets presenta N momentos de desvanecimiento que es directamente proporcional al orden de la wavelet y es asimétrica, tal y como puede observarse en la figura 2.16

2.7.5 Wavelet Coiflet

Una wavelet con un mayor número de momentos de desvanecimiento es la *Coiflet*, cuya gráfica puede observarse en la figura 2.17. Para cada orden diferente de la wavelet se tienen $2N$ momentos de desvanecimiento. Dicha wavelet puede ser simétrica o asimétrica dependiendo del orden N de la wavelet con que se trabaje.

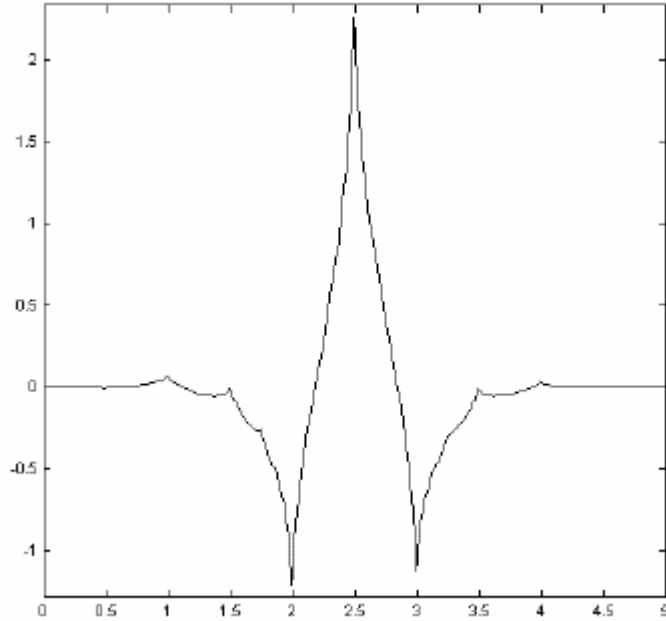


Fig. 2.17: Representación gráfica de Wavelet Coiflet 1

Entre las propiedades de la wavelet Coiflet tenemos:

- Es ortogonal
- Soporte Compacto
- La longitud del filtro es $6N$
- Puede ser simétrica o asimétrica
- La función wavelet Coiflet presenta $2N$ momentos de desvanecimiento, mientras que la función escala Coiflet presenta $2N-1$ momentos de desvanecimiento

2.7.6 Wavelet Morlet

La *Wavelet Morlet* es otra función que única y exclusivamente puede ser utilizada en la CWT. Dicha wavelet es la función de distribución Gaussiana modulada por una señal sinusoidal y puede ser expresada por la ecuación 2.37.

$$morl(t) = e^{-\frac{t^2}{2}} e^{j\alpha t} \quad (2.37)$$

En la figura 2.18 podemos observar la gráfica de la wavelet Morlet, la cual es simétrica y satisface la condición de admisibilidad. Además es importante hacer notar que dicha wavelet no posee propiedades de ortogonalidad ni biortogonalidad.

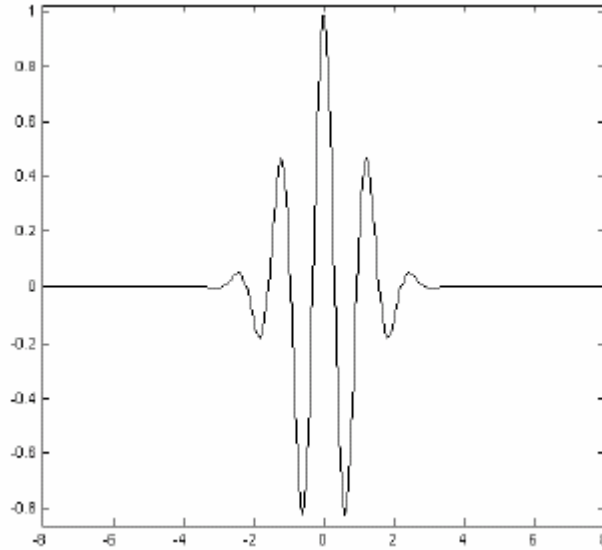


Fig. 2.18: Representación gráfica de Wavelet Morlet

2.7.7 Wavelet Gaussiana

La *Wavelet Gaussiana* es otra muestra muy común de este tipo de funciones finitas. Y se define como la primera derivada de la función de densidad Gaussiana. Y matemáticamente queda definida por la ecuación 2.38 que se presenta a continuación:

$$Gauss(t, n) = C_n \frac{d}{dt} \left(e^{-t^2}, n \right) \quad (2.38)$$

donde C_n es una constante determinada por $2 - norm$ de $Gauss(t, n) = 1$. Con esta wavelet únicamente es posible realizar la CWT y puede ser simétrica o asimétrica según el valor de n . En la figura 2.19 se puede observar la Wavelet Gaussiana de orden 1.

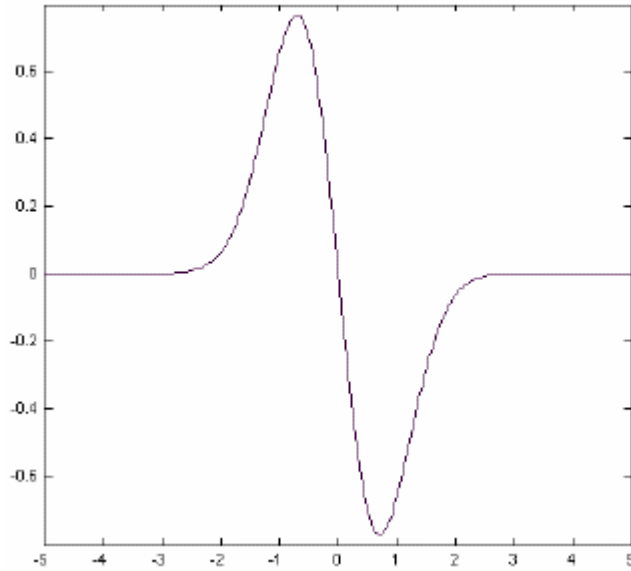


Fig. 2.19: Representación gráfica de Wavelet Gaussiana de orden 1

El objetivo principal de esta sección fue presentar algunas de las familias wavelets que existen. Sin embargo, existen algunas otras wavelets que aquí no se discuten pero que es importante mencionarlas debido a que son parte fundamental del desarrollo de la Teoría de Wavelets, entre estas tenemos: *Wavelet Shannon*, *Spline*, *Biortogonal*, *Meyer*, *Battle-Lemarié* y muchas otras que seguramente seguirán apareciendo.

Es importante hacer notar que la característica común entre todos estos tipos de funciones que se trataron a lo largo de esta sección, es la de ser *funciones finitas*; propiedad que se aprovecha y que es manipulada mediante las variables de *traslación* y *escala*.

Capítulo III

Reducción de ruido en señales utilizando Wavelets

3.1 Introducción

Luego de haber presentado los fundamentos teóricos que dan origen a la Teoría de Wavelets durante los primeros dos capítulos, en el presente se estudia con mayor detalle una de las principales aplicaciones de dicha teoría, como es el caso de la *reducción de ruido en señales*.

De esta forma, el objetivo principal del capítulo 3 será presentar diversos algoritmos, basados en la Teoría de Wavelets, destinados a reducir las perturbaciones causadas por una señal interferente (ruido eléctrico) sobre una señal de interés.

Se presentarán los fundamentos matemáticos y las características de dichos algoritmos en la sección 3.5; para finalmente llevar a cabo una evaluación del desempeño de al menos tres de estos algoritmos auxiliándonos de una muy potente herramienta de simulación matemática, como lo es MATLAB®. Y determinar, luego de dicha comparación, aquel algoritmo que presente los mejores resultados; el cual será posteriormente implementado en la plataforma de simulación que se diseñará y que es el producto final del presente trabajo de graduación.

3.2 Definición de ruido eléctrico

La principal tarea de un demodulador o detector dentro de un sistema de comunicación es recuperar la forma de onda emitida por un transmisor, en la medida de lo posible libre de errores, es decir sin distorsiones. Las principales causas de la distorsión de una señal se pueden dividir en dos: a) efectos de filtrado del transmisor, el canal y el receptor, b) ruido eléctrico producido por diversas fuentes.

Se definirá *ruido eléctrico* como una mezcla de señales aleatorias de diferente frecuencia a la señal de interés (mensaje), pudiendo ser de una frecuencia mayor o menor a la que se desea. Además el ruido eléctrico es de naturaleza indeseable, lo que provoca problemas en la comunicación. En la figura 3.1 se puede observar una señal sinusoidal pura (a) y una segunda señal a la cual se le ha añadido ruido eléctrico (b).

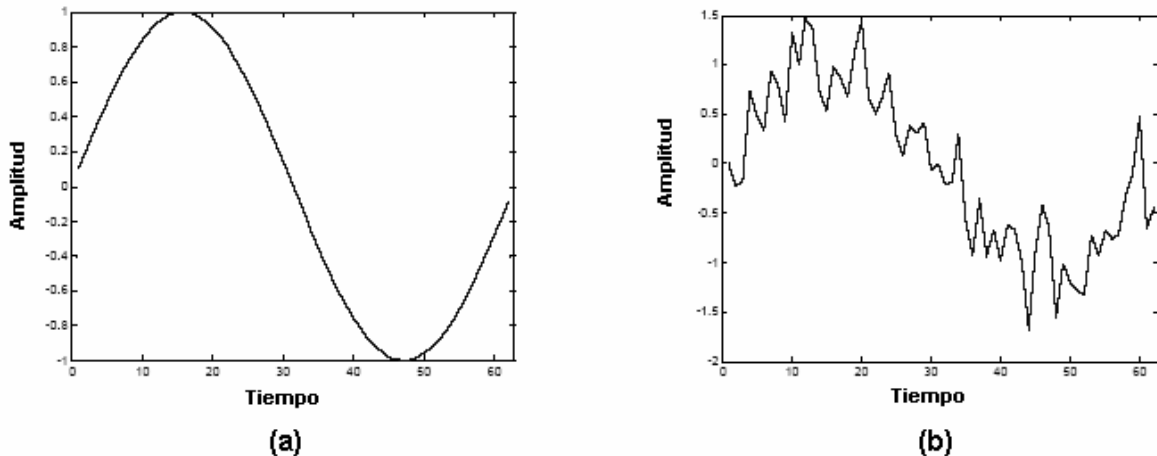


Fig. 3.1: (a) Sinusoidal pura, (b) sinusoidal con ruido aditivo

El ruido eléctrico dependiendo de la fuente que lo produce y de su naturaleza, puede presentarse de diversas formas, el cual en base a su comportamiento puede clasificarse en dos categorías principales: *Correlacionado* y *No correlacionado* [21].

El *ruido correlacionado* es aquel que existe únicamente cuando hay una señal presente. Mientras que por otra parte, el *ruido no correlacionado* existe independientemente de la presencia o no de una señal. En la figura 3.2 se presenta un resumen de la clasificación de los principales tipos de ruido eléctrico.

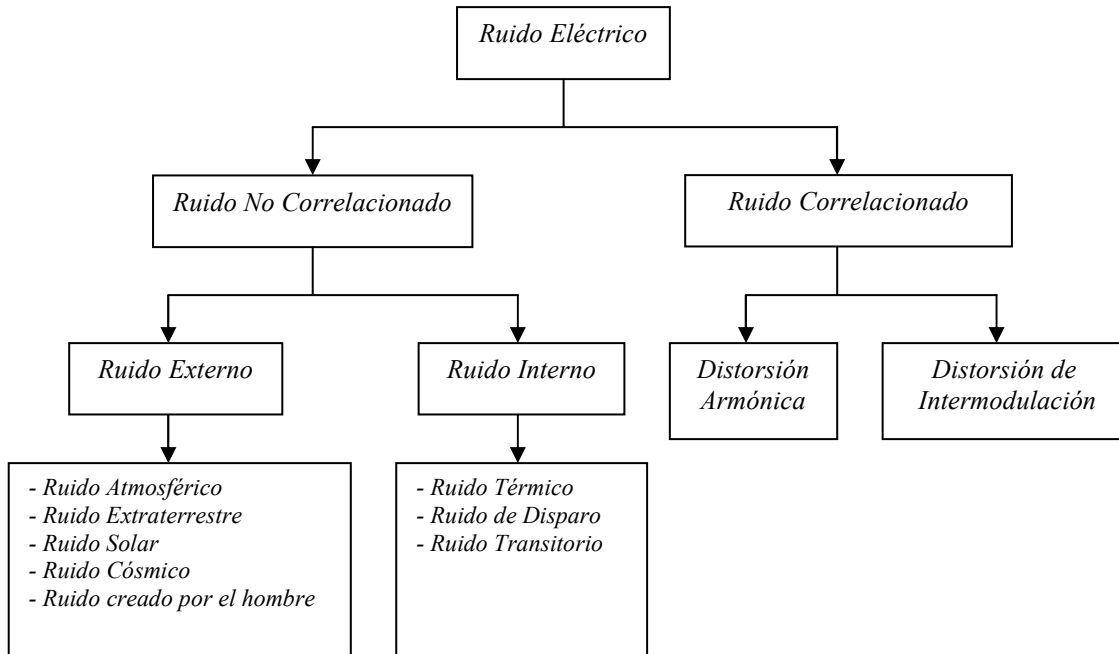


Fig. 3.2: Clasificación del ruido eléctrico [21]

De la clasificación anteriormente presentada en la figura 3.2, se pondrá mayor énfasis en el *ruido térmico*, ya que este, al ser un tipo de ruido *No Correlacionado* se encuentra presente en cualquier sistema de comunicación sin importar si existe o no una señal. Por lo que, siempre esta presente en cualquier diseño y es también el que será utilizado en la sección 3.6 para llevar a cabo las simulaciones y pruebas de los diferentes algoritmos basados en Wavelets que se analizarán. El resto de tipos de ruido eléctrico no serán definidos, ya que no es el objetivo del presente documento.

Una causa inevitable de ruido eléctrico se debe a la movilidad de los electrones en los medios conductores, ocasionando de esta forma el *ruido térmico*. Dicha movilidad de electrones produce en los circuitos electrónicos una degradación de la señal en forma aditiva, es decir, la señal recibida $s(t)$, es la suma de la señal transmitida $f(t)$, y del ruido térmico $r(t)$. Tal como se detalla en la ecuación 3.1.

$$s(t) = f(t) + r(t) \quad (3.1)$$

Capítulo 3. Reducción de ruido en señales utilizando Wavelets

El fenómeno del ruido térmico fue observado por primera vez en 1927 por *J.B. Johnson* de los *Bell Telephone Laboratorios* [21]. El cual se explica de la siguiente manera: un electrón en un conductor porta una unidad negativa de carga, y la velocidad media cuadrática del electrón es proporcional a la temperatura absoluta, por lo que el recorrido de cada electrón entre las colisiones con las moléculas representa un pequeño pulso de corriente que desarrolla un pequeño voltaje a través de los componentes resistivos del conductor. Debido a que los movimientos del electrón son totalmente aleatorios y en todas las direcciones, el voltaje promedio es de 0 voltios de corriente directa. Sin embargo, como se trata de un movimiento aleatorio produce componentes de corriente alterna. A este fenómeno también se le conoce como *ruido browniano*, *ruido Johnson* o *ruido blanco*, ya que el movimiento aleatorio es en todas las frecuencias.

Otra propiedad de interés de este tipo de ruido es que presenta una densidad espectral de potencia constante en todas las frecuencias tal como se puede observar en la figura 3.3 (c), por ello es que a una fuente de ruido térmico se le conoce también como una fuente de *ruido blanco*, en analogía con la luz blanca, la cual presenta todas las frecuencias del espectro visible.

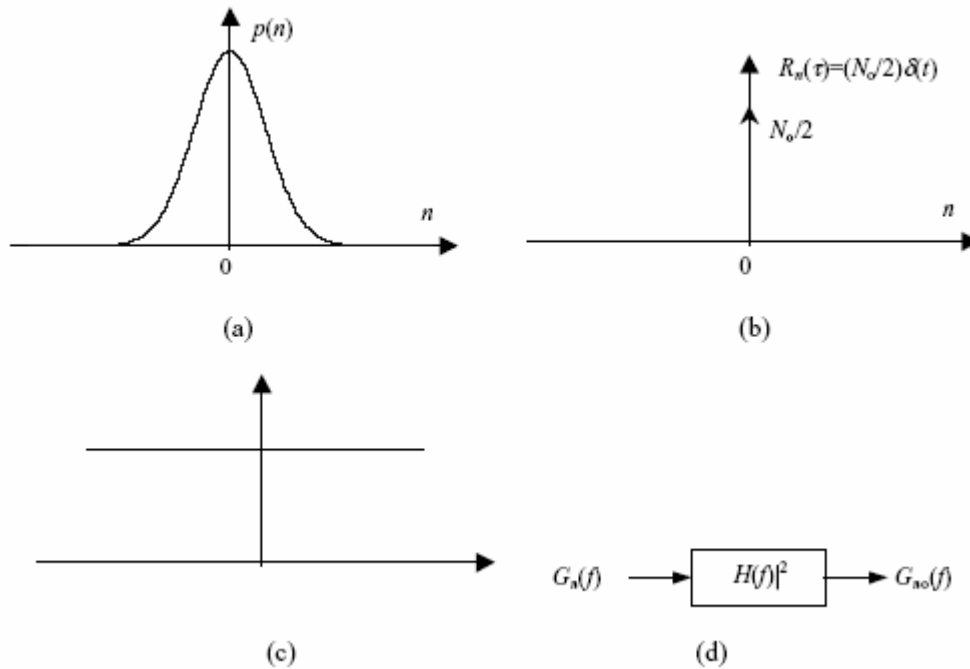


Fig. 3.3: Modelo de Ruido térmico, blanco o Gaussiano

Por tratarse el ruido blanco de una variable aleatoria, solamente puede ser modelado a través de su PDF, la cual denota una distribución normal o Gaussiana con media igual a cero y varianza igual a 1, lo que implica una desviación estándar igual a 1; tal como se observa en la figura 3.3(a) y es denotado por la expresión de la ecuación 3.2. También el ruido blanco o Gaussiano puede ser representado a través de su función de autocorrelación ilustrada en la figura 3.3(b) y su densidad espectral de potencia (figura 3.3(c)).

$$p(n) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{n}{\sigma}\right)^2} \quad (3.2)$$

La mezcla de la señal de interés y la señal de ruido expresada en la ecuación 3.1 puede ocurrir por varias razones, entre algunos de los fenómenos que causan este comportamiento en las señales se encuentran los diferentes tipos de procesamientos a los que son sometidas dichas señales (envío de datos por líneas de transmisión, radiofrecuencia, etc.). Otros factores que provocan el incremento del nivel de ruido en las comunicaciones son el clima y factores geográficos, ya que al hablar de señales que utilizan el aire como medio de transmisión, se presenta el ruido térmico y se producen los fenómenos de reflexión, difracción, dispersión y refracción; los cuales son fenómenos que permiten que las señales se mezclen aunque tengan diferentes componentes de frecuencia.

En la práctica, el nivel de ruido de una transmisión en general, queda definido por la *Relación Señal a Ruido (SNR)*, la cual expresa la relación que existe entre la potencia de la señal de interés (mensaje) P_s y la potencia de la señal de ruido P_n , relación que es definida por la ecuación 3.3. Dicha relación generalmente es expresada en decibeles (dB), esto con el fin e facilitar el manejo de la información.

$$SNR = 10 \log \frac{P_s}{P_n} \quad [\text{dB}] \quad (3.3)$$

Si las resistencias de entrada y salida del amplificador, receptor o red son iguales, la ecuación 3.3 puede expresarse como la relación entre el voltaje de la señal de interés V_s y el voltaje de la señal de ruido V_n tal como se ve expresado en la ecuación 3.4.

$$SNR = 20 \log \frac{V_s}{V_n} \quad [\text{dB}] \quad (3.4)$$

Cuando el valor del SNR es positivo ($P_s > P_n$ ó $V_s > V_n$) implica que existe una ganancia a la salida del receptor; mientras que por otra parte, cuando el valor del SNR de una señal obtenida a la salida del receptor es negativo ($P_s < P_n$ ó $V_s < V_n$) se dice que existen pérdidas. Aunque el decibel es la unidad más ampliamente utilizada, existen otras unidades que se derivan de ella y que tienen un uso específico, para mayor detalle obsérvese la tabla 3.1.

Unidades derivadas del decibel (dB)

| UNIDAD | UNIDADES | REFERENCIA | USO |
|--------|----------|------------------------|-------------------|
| dBa | Potencia | $10^{-11.5} \text{ W}$ | Ruido |
| dBm | Potencia | 1 mW | Comunicaciones |
| dBrn | Potencia | 10^{-12} W | Ruido |
| dBV | Voltaje | 1 V | Amplificadores |
| dBmV | Voltaje | 1 mV | Comunicaciones |
| dBµV | Voltaje | 1 µV | Cktos. Integrados |
| dBw | Potencia | 1 W | Comunicaciones |
| dBx | Potencia | -90 dB | Referencias |

Tabla 3.1: Variantes del decibel y sus usos

3.3 Procedimiento general de reducción de ruido

Entre algunas de las aplicaciones que tiene la Transformada Wavelet se encuentran:

- Detección de discontinuidades
- Evolución a largo plazo
- Identificación de frecuencias puras
- Eliminación de señales
- Compresión de imágenes
- Reducción de ruido en señales, etc.

Para cada una de las aplicaciones antes mencionadas, se pueden distinguir entre ellas, aquellas que utilizan únicamente la Transformada Wavelet (WT) y aquellas aplicaciones que hacen uso conjuntamente de la Transformada Wavelet (WT) y la Transformada Inversa Wavelet (IWT), como es el caso de la reducción de ruido. Aplicación sobre la cual (tal y como se expuso durante la introducción de este capítulo) centraremos este trabajo [15].

El problema de la limpieza de ruido, o en el mejor de los casos, la reducción del nivel de ruido; es un tema que avanza lentamente debido principalmente a la variedad de señales y ruido que existen. Para simplificar el presente estudio se considerarán únicamente señales determinísticas y ruido blanco Gaussiano de media nula y varianza igual a la unidad, tal y como ya se había especificado anteriormente en la sección 3.2, lo cual es una aproximación bastante cercana a la realidad.

Tal y como se expuso en la sección 2.4, el *Análisis Multiresolución* nos permite separar el comportamiento general y el detalle de una señal y luego reconstruir en forma correcta la señal original a partir de la descomposición Wavelet. Para nuestro caso, si se establece un umbral sobre los coeficientes wavelet con el fin de reducir su número eliminando los coeficientes de valores pequeños (considerados como ruido) y dejando solamente aquellos coeficientes considerados como significativos de acuerdo a un cierto criterio, entonces, al realizar la reconstrucción únicamente se obtendrá una aproximación de la señal original. La idea principal es que una versión más clara de la señal original resulte cuando solo las componentes más significativas son retenidas. El criterio de selección del umbral óptimo se basa principalmente en parámetros estadísticos como la desviación estándar, desviación media absoluta, etc.

El proceso genérico para reducir el ruido de una señal utilizando la WT puede describirse por los siguientes pasos [5]:

1. Escoger una Wavelet, un nivel y calcular los coeficientes de descomposición Wavelet de la señal $s(t)$ en el nivel J .
2. Aplicar un umbral a los coeficientes de detalle, desde el nivel 1 hasta el J . El cálculo de dicho umbral y la forma de aplicarlo, determinará el resultado del proceso.

3. Calcular la IWT y a partir de ella reconstruir la señal, basándose en los coeficientes de la aproximación y los coeficientes modificados del detalle.

Los pasos descritos anteriormente pueden ser representados como se detalla en la figura 3.4.

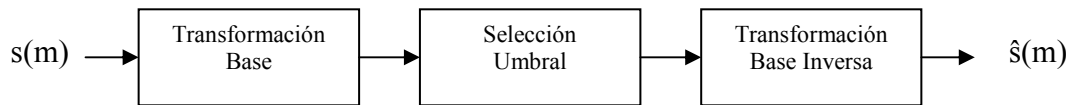


Fig. 3.4: Esquema de proceso de reducción de ruido con wavelets

Las condiciones que se le exigen a dicho proceso de reducción de ruido son que se minimice el *error cuadrático medio*, es decir que se maximice el *SNR* resultante, y que la señal obtenida sea al menos tan suave como la señal inicial, para evitar que, aunque se cumpla la primera condición, aparezcan picos en la señal resultante, los cuales se pueden confundir con información presente en la señal [10].

3.3.1 Criterios de umbral

Existen varios métodos para la reducción de ruido basados en la Transformada Wavelet, los cuales serán discutidos más adelante en la sección 3.2, y que cumplen los requisitos mencionados en el párrafo anterior. La principal diferencia entre dichos métodos es la forma de realizar el cálculo del umbral y la manera de aplicarlo a los coeficientes de detalle. El hecho de establecer un umbral significa que todos los coeficientes Wavelet, o pertenecientes al detalle de la señal, que resulten ser menores que el valor del umbral serán igualados a cero ya que ellos pueden ser omitidos sin afectar en forma substancial las características principales de los datos de entrada [5].

Lo primero que hay que saber es que dichos métodos se pueden clasificar en métodos de reducción de ruido *lineal* y *no lineal*. Los métodos lineales son aquellos que son independientes del tamaño empírico de los coeficientes de la señal, por lo que no son tomados en cuenta. Estos se fundamentan en que el ruido se encuentra principalmente en los coeficientes finos de escala, de este modo elimina todos los coeficientes con una escala

más fina que cierto umbral de escala λ . Si los coeficientes wavelet son $\{d_{j,k}\}$, entonces ellos se relacionan de la forma expresada en la ecuación 3.5.

$$d_{j,k} = \begin{cases} 0 & , j \geq \lambda \\ d_{j,k} & , j < \lambda \end{cases} \quad (3.5)$$

Por otra parte, los métodos no lineales se basan en la idea de que el ruido se encuentra en cada coeficiente y esta distribuido sobre todas las escalas.

Los métodos no lineales pueden ser aplicados en dos versiones el *umbral suave* y el *umbral duro* [5], [14], [15]. El umbral duro elimina los coeficientes que se encuentran por debajo de cierto umbral que se escogió con anterioridad, tal y como lo detalla la ecuación 3.6.

$$s(t) = \begin{cases} s(t) & , |t| > \lambda \\ 0 & , |t| \leq \lambda \end{cases} \quad (3.6)$$

En la ecuación 3.6, $s(t)$ representa a la señal que esta siendo analizada y λ es el umbral que ha sido elegido previamente. El umbral suave, por su parte, es una extensión del umbral duro, ya que este elimina los elementos de menor valor y envía a un valor determinado el resto de ellos. Este proceso es descrito por la ecuación 3.7 que se muestra a continuación:

$$s(t) = \begin{cases} \text{Sign}(t)(|t| - \lambda) & , |t| > \lambda \\ 0 & , |t| \leq \lambda \end{cases} \quad (3.7)$$

En la figura 3.5 que se muestra a continuación se muestra el ejemplo de una función que ha sido tratada con ambos métodos. Es importante hacer notar que el umbral duro genera discontinuidades en $s(t) = \pm \lambda$, mientras que por otra parte, el umbral suave no lo hace.

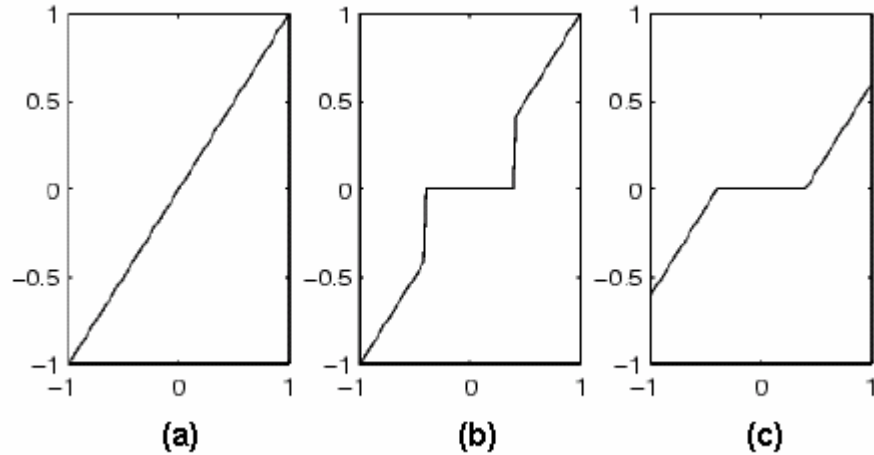


Fig. 3.5: Comportamiento de métodos no lineales de reducción de ruido. (a) Señal original, (b) Umbral duro, (c) Umbral suave

3.4 Algoritmos de reducción de ruido basados en la teoría de wavelets

Con la finalidad de introducir el estudio de diversos algoritmos cuyo objetivo es la reducción de ruido y que se fundamentan en la Transformada Wavelet, se parte de una leve modificación a la ecuación 3.1 y que se detalla en la ecuación 3.8 que se presenta a continuación:

$$s(n) = f(n) + \sigma r(n) \quad (3.8)$$

en donde $f(n)$ es la señal de interés que se desea transmitir, $r(n)$ es ruido blanco Gaussiano y σ es un estimador del nivel de ruido presente en la señal resultante $s(n)$. El objetivo de todos los algoritmos que se estudiarán en la presente sección será recuperar la señal $f(n)$ suprimiendo el ruido presente en la señal $s(n)$ o bien lograr una estimación lo más cercana posible a $f(n)$ [5].

Tal y como se describió en la sección 3.3, el procedimiento para reducir o eliminar ruido en una señal requiere de tres etapas, a continuación los volveremos a mencionar de una manera más específica y con el fin de definir la nomenclatura que se utilizará para el estudio de los diversos algoritmos. Para referirnos a la Transformada Wavelet Discreta hacemos uso de la ecuación 3.9:

$$X(a,b) = DWT\{x(n)\} \quad (3.9)$$

y su transformada inversa es definida por la ecuación 3.10:

$$x(n) = IDWT\{X(a,b)\} \quad (3.10)$$

Ahora se pueden definir las tres etapas necesarias para la reducción de ruido de la siguiente manera [14]:

1. *Descomposición*: Seleccionar una Wavelet ($h_{a,b}$) con la cual se va a trabajar y elegir un número J de niveles de descomposición (resolución) a calcular. Luego calcular la DWT de la señal $s(n)$, como lo detalla la ecuación 3.11:

$$S(a,b) = DWT\{s(n), h_{a,b}, J\} \quad (3.11)$$

2. *Umbral*: Esta es la etapa medular del proceso. Para cada coeficiente (nivel de descomposición) de 1 a J , se selecciona un umbral λ y luego se aplica umbral duro o suave a cada uno de dichos coeficientes. Esto con el fin de eliminar los coeficientes de más baja energía. Para esto definiremos un operador $u\{\}$ que determine el cálculo del umbral (ecuación 3.12) y un segundo operador $D\{\}$ que realice el proceso de reducción de ruido (ecuación 3.13) y de esta manera obtener los coeficientes limpios de ruido definido por $Z\{a,b\}$.

$$\lambda = u\{S(a,b)\} \quad (3.12)$$

$$Z(a,b) = D\{S(a,b), \lambda\} \quad (3.13)$$

3. *Reconstrucción*. Realizar el cálculo de la Transformada Inversa Wavelet Discreta haciendo uso de las modificaciones que se han hecho a los coeficientes, lo cual es descrito por la ecuación 3.14:

$$z(n) = IDWT\{Z(a,b)\} \quad (3.14)$$

Luego de describir de manera general el proceso de reducción de ruido y definir la nomenclatura a utilizar en los diferentes algoritmos que se estudiarán a continuación, es

necesario señalar la importancia que tiene la elección del tipo de umbral y la forma en que se calcula dicho umbral. Ya que, como se menciono anteriormente el método del cálculo del umbral es lo que hace realmente la diferencia en cada algoritmo.

Existen una gran variedad de métodos para elegir el valor del umbral λ y pueden ser agrupados en dos categorías: *umbrales globales* y *umbrales dependientes del nivel*. Los primeros son aquellos en los cuales se selecciona un valor fijo de λ para ser aplicado globalmente a todos los coeficientes wavelets $S(a, b)$ que corresponden a los valores $d_{j,k}$ definidos en la sección 3.3.1. Mientras que los *umbrales dependientes del nivel*, son aquellos que tienen la posibilidad de un valor diferente de umbral λ_a para cada nivel de resolución o escala dado.

Todos los umbrales requieren de una estimación de los niveles de ruido, para lo cual no se utiliza la desviación estándar usual de los valores de los datos. Sino que se hace uso de un algoritmo propuesto por *Donoho* y *Johnstone* en 1994 [15], en el cual, ellos realizan el cálculo de una variable que consideran un estimador en el dominio wavelet y de esta manera sugieren un estimador robusto que esta basado en la desviación media absoluta y toma en cuenta solamente los coeficientes empíricos de wavelets en los niveles de escala más finos, ya que estos son los que concentran la mayor cantidad de ruido. Este estimador de los niveles de ruido (σ) esta dado por, la ecuación 3.15:

$$\sigma = \frac{1}{n} \sum_k \frac{|d_{J-1,k}|}{0.6745} \quad (3.15)$$

En donde J denota el número de niveles de descomposición. Dicho estimador σ , se aplica en los diferentes algoritmos de reducción de ruido que se estudiarán en las siguientes secciones.

3.4.1 Algoritmo Minimax [1], [4], [15]

El primer algoritmo que será estudiado es el *Minimax*, el cual hace uso del principio minimax que se utiliza en estadística para el diseño de estimadores. En este algoritmo, la reducción de ruido de la señal puede ser interpretada como el estimador de una regresión de

una función desconocida, el umbral minimax λ_M es la variable que obtiene el mínimo de un conjunto de valores dados del máximo error cuadrático medio, también conocido como *riesgo*.

Este algoritmo fue el primero en conocerse y fue desarrollado por Donoho y Johnstone en 1994 y hace uso de un umbral que minimiza el término constante en el riesgo de la estimación de una función. El óptimo umbral *Minimax* propuesto, el cual depende del tamaño n de las muestras es descrito por la ecuación 3.16:

$$\lambda_M = \sigma \lambda_n^* \quad (3.16)$$

donde λ_n^* es definido como el valor de λ y es dado por la ecuación 3.17, como sigue:

$$\lambda_n^* = \mathbf{inf}_{\lambda} \mathbf{sup}_d \left\{ \frac{R_{\lambda}(d)}{n^{-1} + R_{oracle}(d)} \right\} \quad (3.17)$$

Donde $R_{\lambda}(d) = E(\delta_{\lambda}(d) - d)^2$ y $R_{oracle}(d)$ es el riesgo ideal calculado con ayuda de uno de los dos oráculos. Uno de ellos denominado *Diagonal Linear Projection (DLP)*, el cual indica cuando “mantener” o “anular” cada uno de los coeficientes wavelet. Y el otro es el *Diagonal Linear Shrinker (DLS)*, el cual nos dice cuando escoger cada coeficiente wavelet. El riesgo ideal calculado para cada uno de los oráculos esta definido por las ecuaciones 3.18 y 3.19, que se expresan a continuación:

$$R_{oracle}^{DLP}(d): = \mathbf{min}(d^2, 1) \quad (3.18)$$

$$R_{oracle}^{DLS}(d): = \frac{d^2}{d^2 + 1} \quad (3.19)$$

donde la expresión $\mathbf{min}(\cdot)$ denota el valor mínimo de d^2 .

3.4.2 Algoritmo Fixed form threshold [1], [4], [15]

Este segundo algoritmo también fue propuesto y desarrollado por Donoho y Johnstone en 1994 y recibe el nombre de *fixed form threshold*, *universal threshold* o *visushrink*. Y fue propuesto como una alternativa al uso de umbrales *Minimax*. En dicho algoritmo hacen uso de un umbral fijo, el cual está definido por la ecuación 3.20.

$$\lambda_{UNI} = \sigma \sqrt{2 \log(N)} \quad (3.20)$$

En donde el umbral λ_{UNI} es función de N , la cual representa la longitud de la señal a limpiar y σ es el estimador de los niveles de ruido que ya ha sido definido por la ecuación 3.15. Esto provoca que el umbral varíe de acuerdo con la longitud de la señal. Es decir, que para una señal de mayor tamaño el umbral será mayor que para una señal de menor tamaño.

Este umbral universal (ecuación 3.20) es sustancialmente más grande que el umbral minimax descrito por la ecuación 3.16 para cualquier valor particular de N . Como resultado de este fenómeno, la reconstrucción incluirá menos coeficientes, lo que representa una estimación, lo cual es el mejor balance para el proceso de limpieza, ya que se distribuye mejor la estandarización de los coeficientes que en la estimación minimax. Otra característica importante es que este umbral tiene una alta probabilidad que garantiza que todas las muestras de la DWT en las cuales la función analizada es exactamente cero, serán estimadas como cero. Este algoritmo al igual que el *Minimax*, son catalogados dentro de los métodos de umbral global.

Una cuestión importante que se debe aclarar es el por qué el umbral depende de la longitud de la señal. La explicación radica en que si agregamos más muestras aumenta la redundancia de la señal, es decir, hay menos información nueva en las nuevas muestras de las que hay en las primeras muestras. En el dominio wavelet, esto significa que el número de los coeficientes importantes está creciendo escasamente y toda la información está concentrada en un número limitado de coeficientes. De esta forma, el número total de coeficientes de ruido es proporcional a n al igual que el número de coeficientes que tienen

la información, solo que se guarda una relación que esta determinada por el umbral universal.

3.4.3 Algoritmo Rigorous SURE [1], [4], [15]

Los dos algoritmos presentados anteriormente coinciden en ser de umbral global, sin embargo el algoritmo *Rigorous SURE* propuesto también por Donoho y Johnstone en 1995 introduce un esquema que utiliza los coeficientes de las wavelets de tal forma que para cada nivel de escala a exista un umbral λ_a . A este algoritmo también se le conoce como *Sureshrink* porque se basa en la aplicación del *Estimador de Riesgo Imparcial de Stein* (*SURE*, por sus siglas en inglés), el cual es un estimador de umbral suave. También es conocido como *Función de pérdida cuadrática*.

Lo primero que se hace es obtener una estimación de riesgo para un valor de umbral en particular. Reducir el riesgo o pérdida en ese valor nos da una selección del valor del umbral. El valor final del umbral, después del procedimiento matemático esta dado por la ecuación 3.21.

$$\lambda_{SURE} = \mathbf{arg\ min}_{0 < \lambda < \lambda_{UNI}} SURE\left(\lambda, \frac{S(a, b)}{\sigma}\right) \quad (3.21)$$

Donde $a = a_0, \dots, J-1$; $b = 0, 1, \dots, 2^{J-1}$; λ_{UNI} fue definido por la ecuación 3.20; $S(a, b)$ son los coeficientes de la DWT y σ ya fue definido por la ecuación 3.15. El operador *arg min* denota que se utilizarán los elementos menores del estimador *SURE* cuya relación matemática esta definida por la ecuación 3.22.

$$SURE(\lambda; X) = n - 2 \otimes \{i : |X_i| \leq \lambda\} + [\min(|X_i|, \lambda)]^2 \quad (3.22)$$

En 3.22 el símbolo \otimes denota la cardinalidad del conjunto $\{i : |X_i| \leq \lambda\}$. Para el caso de los coeficientes DWT la variable X se cambia por el coeficiente $\frac{S(a, b)}{\sigma}$.

3.5 Simulación de algoritmos para reducción de ruido

El objetivo de la presente sección será llevar a la práctica los fundamentos de la teoría de Wavelets que fueron establecidos anteriormente a lo largo de este documento, y más específicamente en la aplicación de reducción de ruido en señales. Para conseguir esto nos auxiliaremos de una potente herramienta de simulación y cálculo matemático, como lo es MATLAB® 7 Release 14 y su Toolbox de Wavelets. Con dicha herramienta se llevarán a cabo una serie de simulaciones de los diversos algoritmos que se presentaron en la sección 3.4, para que finalmente se concluya este capítulo con una comparación entre dichos algoritmos y de aquí dar paso a la selección de aquel algoritmo que arroje los mejores resultados; y de esta manera desarrollar la plataforma de simulación la cual es el fin último del presente trabajo de graduación.

3.5.1 Definición de señales de prueba

Con el fin de llevar a cabo las diversas simulaciones se definirán un conjunto de señales de prueba que son una recopilación de ecuaciones definidas por diferentes autores, las cuales fueron desarrolladas para evaluar el comportamiento de los algoritmos ante las diferentes propiedades que cada una de ellas presentan.

Las señales que aquí se presentarán han sido utilizadas en diversos estudios de wavelets realizados por *Marron, Adak, Johnstone, Neumann y Patil en 1998* [4].

A continuación se definen las funciones que servirán para nuestro estudio [4]:

1. *Función Cuadrada (Step)*: Las características de saltos repentinos (discontinuidades) que presenta la señal cuadrada son una buena prueba para evaluar la velocidad de reacción de los diversos algoritmos. Por estas mismas características dicha función es muy difícil de estimar a través de métodos lineales no así a través de métodos no lineales. La función que aquí se evaluará está definida por la ecuación 3.23, en donde $I_{(1/3, 3/4)}(x)$ denota los valores de x que están entre $1/3$ y $3/4$ y que puede observarse en la figura 3.6 (a).

$$f_1(x) = 0.2 + 0.6I_{(1/3, 3/4)}(x) \quad (3.23)$$

2. *Función de Ondas:* Esta función es el resultado de la suma de dos señales sinusoidales, la principal característica que presenta esta señal es que es menos suave que las señales sinusoidales normales, lo que nos servirá para evaluar el comportamiento de los diferentes tipos de wavelets en base a los momentos de desvanecimiento que cada una de ellas presentan. Con este tipo de señal, los resultados obtenidos al aplicar métodos lineales son comparables a los obtenidos al aplicar métodos no lineales. En la ecuación 3.24 se puede observar la expresión matemática de dicha función. En la figura 3.6 (b) se muestra la representación grafica de dicha función.

$$f_2(x) = 0.5 + 0.2\text{Cos}(4\pi x) + 0.1\text{Cos}(24\pi x) \quad (3.24)$$

3. *Función Cresta:* Este tipo de señal se obtiene al sumar una función lineal y la función de distribución de Densidad Gaussiana. El salto repentino que dicha función describe, es de gran utilidad para evaluar velocidad de reacción y a la vez el comportamiento frente a una señal suave. La forma de onda y la ecuación que describe a este tipo de función se pueden observar en la ecuación 3.25 y la figura 3.6 (c) respectivamente.

$$f_3(x) = \left(0.32 + 0.6x + 0.3e^{-100(x-0.3)^2}\right)I_{(0, 0.8)}(x) + \left(-0.28 + 0.6x + 0.3e^{-100(x-1.3)^2}\right)I_{(0.8, 1)}(x) \quad (3.25)$$

4. *Función de bloques:* Esta señal es semejante a la función cuadrada presentada anteriormente, la diferencia radica en que esta presenta diferentes amplitudes entre en cada uno de los saltos. La gráfica de esta señal se presenta en la figura 3.6 (d) y queda perfectamente definida por la ecuación 3.26.

$$f_4(x) = \sum h_j K(x - t_j) \quad (3.26)$$

$$K(x) = \frac{1 + \text{sgn}(x)}{2}$$

En donde:

$$(t_j) = (0.1, 0.13, 0.15, 0.23, 0.25, 0.40, 0.44, 0.65, 0.76, 0.78, 0.81)$$

$$(h_j) = (4, -5, 3, -4, 5, -4.2, 2.1, 4.3, -3.1, 5.1, -4.2)$$

5. *Función Bumps*: Este tipo de señal se caracteriza por presentar saltos repentinos y picos muy pronunciados, por lo cual es un verdadero reto para cualquier método de reducción de ruido. Esta función es muy similar a una señal de ruido impulsivo que es muy común encontrar en los sistemas de distribución de energía eléctrica. La función *Bumps* y su expresión pueden ser observadas en la ecuación 3.27 y la figura 3.6 (e) respectivamente.

$$f_5(x) = \sum h_j K\left(\frac{x - t_j}{w_j}\right) \quad (3.27)$$

$$K(x) = \left(1 + |x|^4\right)$$

En donde:

$$(t_j) = (0.1, 0.13, 0.15, 0.23, 0.25, 0.40, 0.44, 0.65, 0.76, 0.78, 0.81)$$

$$(h_j) = (4, 5, 3, 4, 5, 4.2, 2.1, 4.3, 3.1, 5.1, 4.2)$$

$$(w_j) = (0.005, 0.005, 0.006, 0.01, 0.01, 0.03, 0.01, 0.01, 0.005, 0.008, 0.005)$$

6. *Función de Ángulos*: Esta función presenta cambios repentinos en su pendiente y permite evaluar la función triangular que se presenta en una parte de dicha onda. La forma de onda descrita por la ecuación 3.28 y que se ve ilustrada en la figura 3.6 (f) describen perfectamente este tipo de funciones.

$$f_6(x) = (2x + 0.5)I_{(0, 0.15)}(x) + (-12(x - 0.15) + 0.8)I_{(0.15, 0.2)}(x) + 0.2I_{(0.2, 0.5)}(x) \quad (3.28)$$

$$+ (6(x - 0.5) + 0.2)I_{(0.5, 0.6)}(x) + (-10(x - 0.6) + 0.8)I_{(0.6, 0.65)}(x)$$

$$+ (-5(x - 0.65) + 0.3)I_{(0.65, 0.85)}(x) + (2(x - 0.85) + 0.2)I_{(0.85, 1)}(x)$$

7. *Función Senoidal desplazada en el tiempo*: Tal y como su nombre lo indica, esta función no es más que una señal sinusoidal desplazada en el tiempo. Este tipo de función será de utilidad para evaluar el desempeño de los algoritmos para la

reducción de ruido frente a señales suaves. En la ecuación 3.29 se describe la forma matemática de dicha función, y en la figura 3.6 (g) describe su representación gráfica.

$$f_7(x) = 0.3 \text{Sen}(3\pi(g(g(g(g(x)))))) + x) + 0.5 \quad (3.29)$$

$$g(x) = \frac{1 - \text{Cos}(\pi x)}{2}$$

8. *Función Esquina*: Como última función a presentar y que posteriormente se evaluará a través de las diferentes simulaciones, tenemos la función *Corner (esquinas)*. La particularidad de este tipo de función es que presenta una combinación de crestas suaves y valles rígidos. La función matemática que describe este comportamiento se detalla en la ecuación 3.30 y en la figura 3.6 (h) se puede observar su forma de onda.

$$f_8(x) = \left(\frac{0.6}{\text{range}(g)} \right) g(x) + 0.6 \quad (3.30)$$

$$g(x) = 623.87x^3(1-4x)I_{(0,0.5)}(x) + 187.161(0.125+x^3)x^4I_{(0.5,0.8)}(x) + 3708.470441(x-1)^3I_{(0.8,1)}(x)$$

Luego de haber finalizado la descripción de las diferentes señales de prueba que se analizarán en nuestras simulaciones, con el fin de evaluar el desempeño de los algoritmos para reducción de ruido, se definirán a continuación en el apartado 3.5.2 los parámetros que serán tomados en cuenta a la hora de llevar a cabo dichas simulaciones. Así también, se detallara paso a paso el procedimiento que se lleva a cabo para “limpiar” a una señal de nuestro interés de la perturbación del ruido blanco, esto con el fin de presentar a manera de ejemplo el proceso que se ejecutará en las diversas combinaciones de simulaciones; ya que no sería práctico evaluar por separado cada una de estas combinaciones.

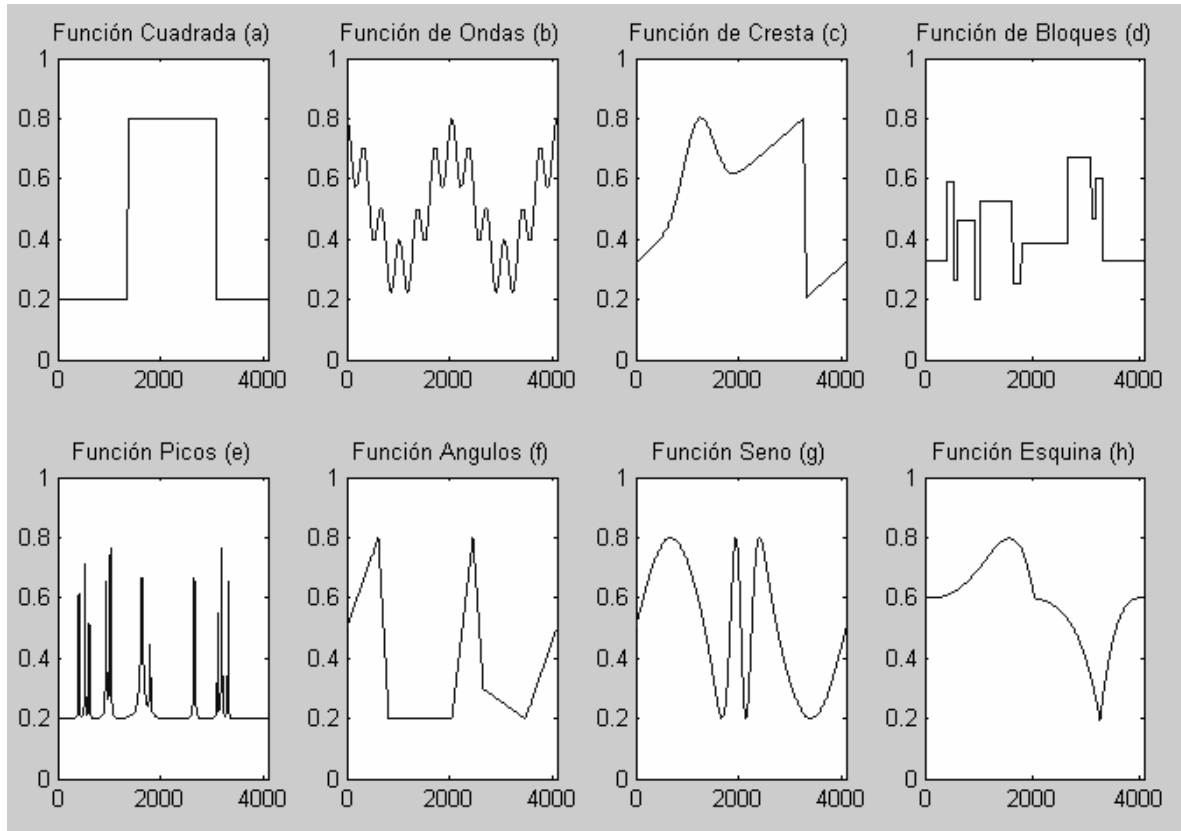


Fig. 3.6: Señales sintéticas de prueba con $n = 4096$ muestras

3.5.2 Definición de parámetros de simulaciones

Los resultados que se obtengan de las simulaciones realizadas pueden variar ampliamente de simulación en simulación, ya que los algoritmos de reducción de ruido dependen de muchas variables, tales como: *número de muestras*, *tipo de umbral*, *tipo de wavelet*, *señal de prueba*, *relación señal a ruido* y *nivel de descomposición*. Así pues, debido a la gran cantidad de parámetros involucrados en las simulaciones es evidente que no sería práctico presentar en este trabajo todos los resultados posibles a partir de la combinación de las diferentes variables; por lo que se definirán ciertos límites para el presente trabajo los cuales se detallan en la tabla 3.2 que se muestra a continuación:

| <i>Parámetro</i> | <i>Valor</i> |
|----------------------------------|---------------------------------------------------|
| Número de muestras (n) | 128, 512, 1024, 2048 y 4096 |
| Nivel de descomposición (J) | 7, 9, 10, 11, 12 |
| Tipo de umbral | Suave y Duro |
| Tipo de Wavelet | Haar, Daubechies, Coiflets y Symlets (de orden 5) |
| Relación Señal a Ruido (SNR) | 3dB |
| Señales de prueba | Todas las estudiadas en el apartado 5.3.1 |

Tabla 3.2: Delimitación de variables para simulaciones en MATLAB®

En la tabla 3.2 encontramos dos parámetros que se encuentran íntimamente ligados en este tipo de análisis, tal es el caso del *número de muestras* y del *nivel de descomposición*, esta relación se ve reflejada en la ecuación 3.31.

$$n = 2^J \quad (3.31)$$

De aquí los valores mostrados en la primera y segunda fila de dicha tabla. Posteriormente, a las señales de prueba que se definieron en la sección 3.5.1 se les agregará ruido blanco Gaussiano con $N(0, 1)$. Se tiene además una relación señal a ruido igual a 3dB, lo cual denota que existe una ganancia en la señal recibida con ruido. Mientras que la elección de las wavelets escogidas obedece a que todas ellas son ortogonales, con lo que se garantiza la estabilidad del análisis. Y finalmente, se evaluarán ambos umbrales (suave y duro) y de esta manera poder comparar el comportamiento de ambos análisis.

Con la finalidad de evaluar el desempeño de los algoritmos, se hará uso del Error Cuadrático Medio (*MSE*, por sus siglas en inglés), el cual compara las señales de salida, luego de haber sido procesada por el algoritmo, con las señales originales. Esta relación puede ser representada de la forma que se muestra en la ecuación 3.32 [17]:

$$MSE = \frac{1}{n} \sum_{i=1}^n (f - f_e)^2 \quad (3.32)$$

donde f representa los valores de la señal original, dentro de las n muestras existentes, mientras que f_e se refiere a la señal reconstruida a partir de los coeficientes wavelets sin ruido, esto luego de aplicar el umbral respectivo.

Es importante señalar que si las señales f y f_e son iguales, el valor del MSE , tal y como se puede apreciar en la ecuación 3.32, es igual a cero. Ello implica que una señal con un valor de MSE muy cercano a cero será una mejor estimación de la señal original que aquella que este más alejada del cero.

3.5.3 Simulación de algoritmos utilizando MATLAB®

En la sección 3.4 ya fueron definidos matemáticamente los algoritmos de reducción de ruido que serán simulados, en este punto se está listo para presentar la manera en la cual nuestra herramienta de simulación (MATLAB®) procesa dichos algoritmos, ya que ella en algunos casos difiere de la teoría presentada. Estas diferencias radican en que los algoritmos de los programas implementados en el Wavelet Toolbox de MATLAB® suponen que se trabajará con ruido blanco Gaussiano y por ende se tiene la forma $N(0, 1)$, donde se denota la distribución normal, la media igual a cero y la desviación estándar σ igual a uno, por lo que muchos cálculos se simplifican. Ahora bien, si se desea tomar en cuenta la variable σ , esta puede ser calculada haciendo uso de la ecuación 3.15.

A continuación se presenta la forma en que MATLAB® interpreta cada uno de los algoritmos:

1. *Algoritmo Minimax*. Este algoritmo es interpretado por MATLAB® como se detalla en la ecuación 3.33. En este caso lo que se hace es evaluar la longitud de la señal con ruido y en caso de que esta sea menor a 32 muestras el umbral es cero, en caso contrario se aplica la ecuación correspondiente para calcular el umbral. En la ecuación 3.33 se puede observar que este es un umbral global, ya que el valor de λ_M es el mismo para todas las muestras analizadas.

$$\lambda_M = \begin{cases} 0 & , n < 32 \\ 0.3936 + 0.1829 \left(\frac{\log(n)}{\log(2)} \right) & , n > 32 \end{cases} \quad (3.33)$$

2. *Algoritmo Fixed form Threshold.* Para la implementación práctica de este algoritmo MATLAB® hace uso de la misma expresión de la ecuación 3.20, sin embargo, la estimación que aquí tiene lugar es que el nivel de ruido $\sigma = 1$, lo cual simplifica aún más este algoritmo.
3. *Algoritmo Rigorous SURE.* Nuestra herramienta de simulación contiene también un programa que realiza el cálculo de umbral para este algoritmo, el cual está implementado según las ecuaciones 3.34 y 3.35.

$$SURE_M = n - 2\{1:n\} + \frac{\sum_{a=0}^n |S(a,b)|^2 + \{n-1:0\}|S(a,b)|^2}{n} \quad (3.34)$$

$$\lambda_{SURE_M} = \sqrt{|S(a,b)|_{\min(SURE_M)}^2} \quad (3.35)$$

Ahora que se ha definido la forma en que MATLAB® interpreta cada uno de los algoritmos, se procederá a presentar la forma en que las diversas señales sintéticas de prueba serán analizadas por cada uno de los algoritmos. El procedimiento que se describirá a continuación tendrá un fin ilustrativo, ya que los resultados completos de los análisis serán presentados en forma detallada en el apéndice B.

En el Apéndice A se pueden encontrar el conjunto de programas desarrollados y que se utilizaron para llevar a cabo las simulaciones de los algoritmos de reducción de ruido en sus diferentes combinaciones de parámetros, cuyos resultados se pueden consultar en el Apéndice B, como ya se estableció en el párrafo anterior.

En el presente apartado se describirá el funcionamiento de los programas, así como el proceso que el software sigue para obtener los resultados mostrados en el apéndice B.

El primer paso es construir las señales de prueba que se utilizarán en nuestro análisis, esto se logra a través la función *ConstruirSignal.m* y del script *GenerarSignals.m*. Los cuales están basados en las expresiones matemáticas de las funciones sintéticas de prueba ya antes definidas en el apartado 3.5.1 y que se pueden observar en la figura 3.6. Luego de esto, se agrega Ruido Blanco Gaussiano de la forma $N(0, 1)$ a cada una de las señales de prueba, lo cual se logra a través de las ecuaciones 3.36 y 3.37, definidas a continuación:

$$Stds = \frac{std(f_i(x))}{SNR} \quad (3.36)$$

$$e(n) = randn(size(f_i(x)))Stds \quad (3.37)$$

donde SNR es la relación Señal a Ruido, que para el presente estudio se ha utilizado un valor de $3dB$; $f_i(x)$ representa a cada una de las señales de prueba (*Cuadrada, Ondas, Cresta, Bloques, Picos, Ángulos, Seno Desplazado en el Tiempo y Esquinas*); $e(n)$ representa al ruido blanco Gaussiano que será sumado a la señal de prueba generado por la función *randn* y donde la expresión $size(f_i(x))$ es equivalente al número de muestras n utilizada para generar las señales de prueba. En la figura 3.7 se puede observar una muestra del conjunto de señales de prueba con el ruido blanco agregado, la cual es generada con el script *GenerarSignals.m*; en el cual también se lleva a cabo la mezcla de las señales de prueba con el ruido blanco Gaussiano.

Luego de obtener las señales de prueba con el ruido mezclado, se procederá a calcular algunos parámetros de interés para nuestro análisis, como son el *umbral* (λ) el cual establece el nivel de los coeficientes que serán suprimidos dependiendo del tipo de umbral que se utilice (*suave o duro*). El segundo parámetro es el *Error Cuadrático Medio (MSE)*, el cual será de utilidad al momento de evaluar el desempeño de cada uno de los algoritmos.

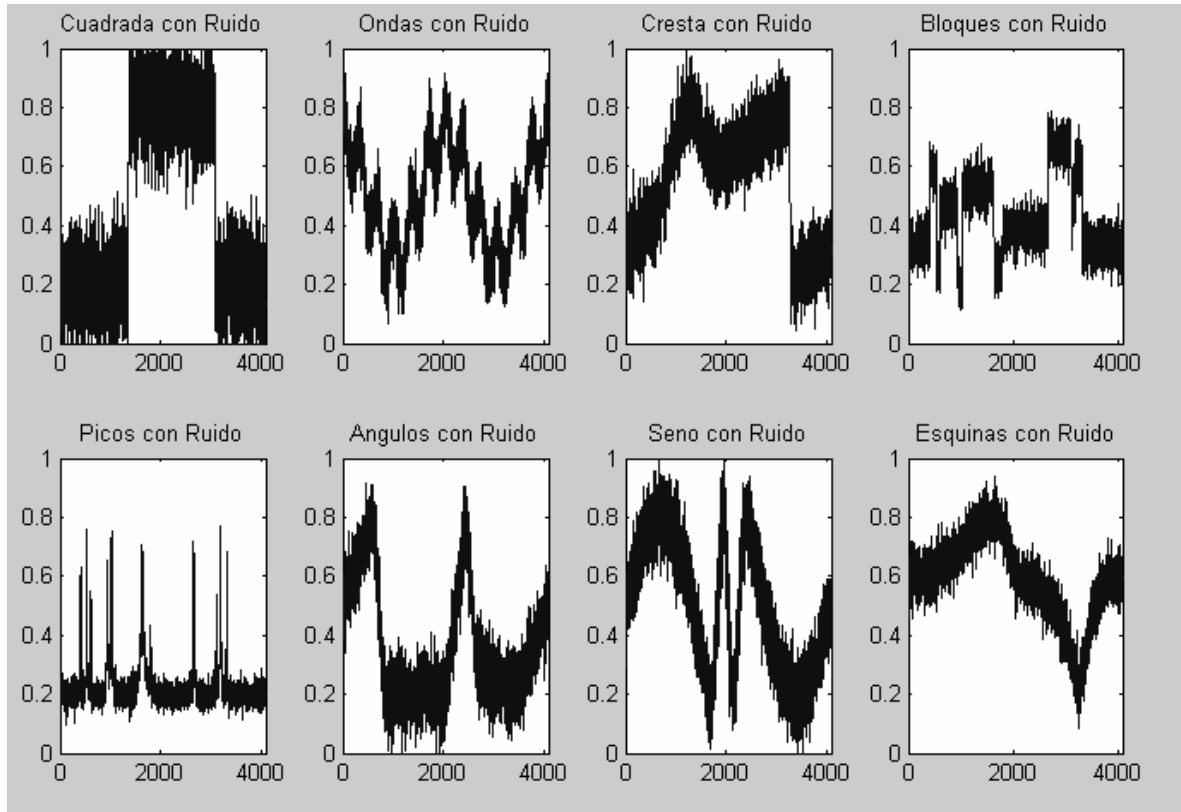


Fig. 3.7: Señales sintéticas de prueba con ruido, $n = 4096$ y $SNR = 3$

Para obtener el valor del umbral λ , Matlab® y el *Wavelet Toolbox*® hacen uso de la función $thr = thselect(y, tptr)$. Donde y representa a la señal de prueba con ruido y $tptr$ al algoritmo de reducción de ruido que se está evaluando (*Rigorous SURE*, '*rigrsure*'. *Fixed form Threshold*, '*sqtwolog*'. *Heuristic SURE*, '*heursure*'. *Mínimax*, '*minimaxi*'). El algoritmo *Heuristic SURE* no es evaluado en el presente estudio, ya que es una mezcla de *Rigorous SURE* y *Fixed form Threshold*, esta función se puede observar con más detenimiento en el script *Denoising.m* del Apéndice A.

Por otra parte, para llevar a cabo el cálculo del Error Cuadrático Medio (*MSE*) se hace uso de la función $mse(f, y)$, donde f representa a la señal original, mientras que y representa a la señal recuperada a partir de la muestra con ruido, este proceso es perfectamente descrito por la ecuación 3.32, y puede encontrarse en el script *Denoising.m*.

Capítulo 3. Reducción de ruido en señales utilizando Wavelets

En la tabla 3.3, se puede observar a manera de ejemplo, el conjunto de parámetros obtenidos a partir de los procedimientos anteriormente descritos para el cálculo del umbral y el MSE.

| Algoritmo: Minimax Umbral Suave | | | | | |
|------------------------------------|----------|----------|----------|----------|----------|
| Wavelet Daubechies 5 | | | | | |
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| Umbral (λ) | 1.6739 | 2.0397 | 2.2226 | 2.4055 | 2.5884 |
| | MSE | MSE | MSE | MSE | MSE |
| Señal Cuadrada | 0.01565 | 0.004349 | 0.003102 | 0.001204 | 0.001006 |
| Señal Ondas | 0.013156 | 0.004959 | 0.000601 | 0.000109 | 0.000097 |
| Señal Cresta | 0.011319 | 0.003313 | 0.001527 | 0.000512 | 0.000386 |
| Señal Bloques | 0.010563 | 0.004411 | 0.002608 | 0.001299 | 0.00062 |
| Señal Picos | 0.005736 | 0.003816 | 0.002664 | 0.002131 | 0.000917 |
| Señal Ángulos | 0.023731 | 0.001743 | 0.000231 | 0.00018 | 0.000131 |
| Señal Seno | 0.019663 | 0.004292 | 0.000262 | 0.000168 | 0.000159 |
| Señal Esquinas | 0.003587 | 0.000103 | 0.000061 | 0.000066 | 0.000051 |

Tabla 3.3: Ejemplo de Valores de MSE y umbral: Minimax, Umbral Suave, Daubechies 5

En la sección 3.5.4 se discutirán las implicaciones y significado de los valores presentados en la tabla 3.3, la cual ha sido tomada del apéndice B.

En el script *Denoising.m* que se detalla en el apéndice A, también encontramos la función *wden* la cual juega un papel determinante en la presente aplicación, ya que esta se encarga de desarrollar de manera automática el proceso de reducción/eliminación de ruido. Para llevar a cabo su tarea, esta función hace uso de una serie de parámetros y tiene la forma: $[XD, CXD, LXD] = wden(X, TPTR, SORH, SCAL, N, 'wname')$, la cual entrega como resultado una versión “libre de ruido” *XD* de la señal de entrada *X*. El parámetro *TPTR* al igual que en el caso anterior, representa la regla de selección del umbral, es decir, el algoritmo a aplicar en el proceso de reducción de ruido. El término *SORH* es el que determina el tipo de umbral que se evalúa, es decir, *suave* (*'s'*) o *duro* (*'h'*). *SCAL* representa un factor de escala o *estimador* del nivel de ruido, que para nuestro caso sea establecido en *'one'* lo cual implica que no se esta escalando el nivel de ruido Gaussiano presente. Además *N* representa el nivel de descomposición, que como ya se estableció, ha sido fijado en 5. Y finalmente *'wname'* se refiere al nombre de la Wavelet que se utilizará

para realizar la recuperación de la señal, los valores o tipos de wavelet que toma este parámetro son: *Haar*, *db5*, *coif5*, *sym5*. En la figura 3.8 se puede observar el resultado de la recuperación de las señales de prueba.

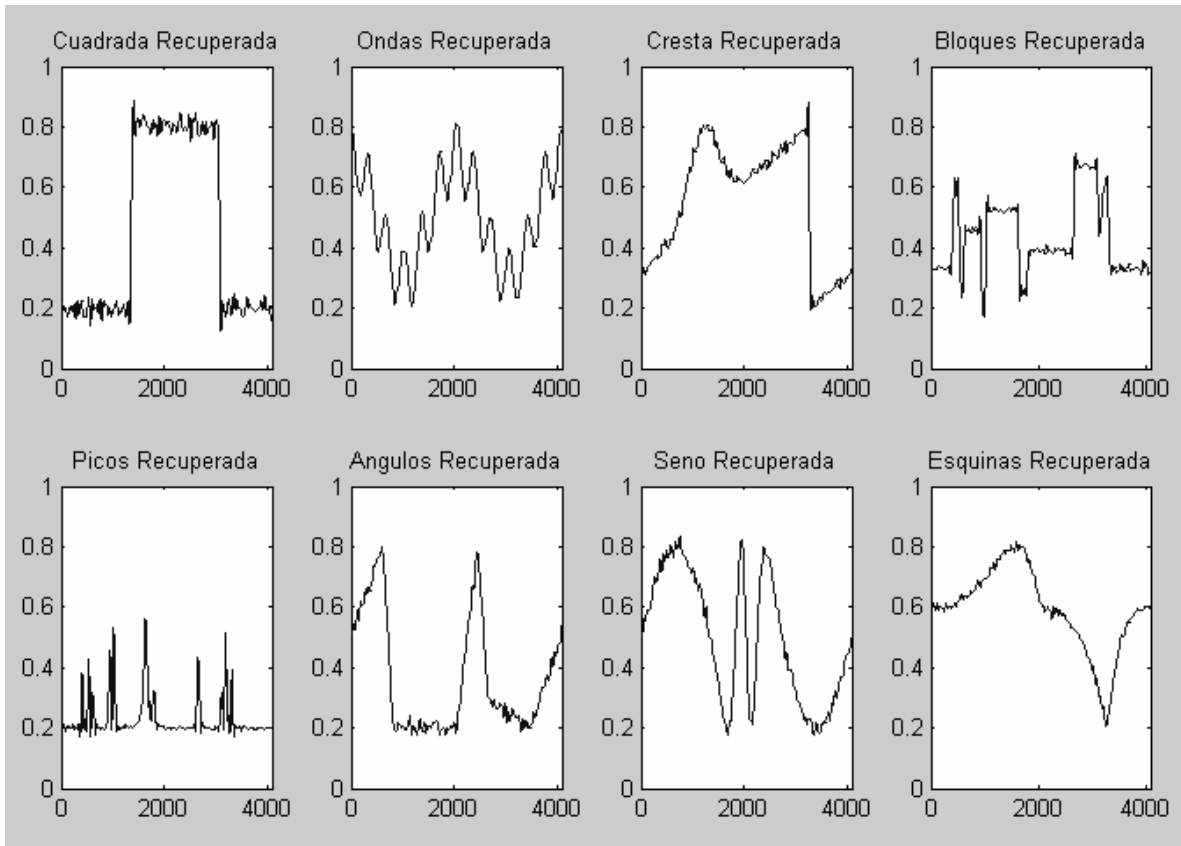


Fig. 3.8: Señales de prueba recuperadas: *Minimax*, *Umbral Suave*, *Daubechies 5*

Si comparamos la figura 3.8 con la figura 3.6, podemos observar en general que la recuperación de las señales utilizando Wavelet es bastante buena, pues se consigue el principal cometido, el cual es reducir el nivel de ruido. En el apartado 3.5.4 se analizarán con mayor detalle los fenómenos involucrados en este procedimiento y se explicarán las implicaciones que tienen en la reducción del ruido los diferentes parámetros involucrados.

3.5.4 Análisis de resultados y comparación entre algoritmos

En la presente sección se analizará el comportamiento de cada uno de los algoritmos implementados y su eficiencia para la reducción del ruido en las señales de prueba. Para ello se tomará la pareja *Wavelet – Método* que presente los valores de MSE más cercanos a

cero para cada una de las señales de prueba evaluadas. Cada una de estas combinaciones a su vez, serán clasificadas según el tipo de umbral que se aplicó (*suave o duro*), ello con el fin de diferenciar el comportamiento de los mismos. Para obtener dichos datos nos basaremos en las 960 simulaciones que fueron realizadas y que se encuentran en el Apéndice B.

Las tablas 3.4 y 3.5 han sido derivadas de dicho conjunto de simulaciones y los resultados han sido divididos según el tipo de umbral que se ha aplicado, umbral duro y umbral suave respectivamente. La nomenclatura que se ha utilizado en dichas tablas es la siguiente: *RS* (*Método Rigorous SURE*), *FT* (*Método Fixed form Threshold*) y *MM* (*Método Minimax*).

En primer lugar, es de hacer notar la relación que existe entre el número de muestras (n) y la calidad de la señal recuperada, expresada en nuestro caso por el valor del MSE, ya que tal y como se puede observar, al incrementar el número de muestras el valor del MSE de la señal disminuye y se acerca más a cero. Sin embargo, aunque matemáticamente un valor de MSE del orden de 1×10^{-3} se puede considerar un valor bastante aceptable, al observar las representaciones gráficas de algunas simulaciones, es de destacar que la calidad de la señal recuperada no es del todo buena, ya que se suprimen características de la señal que podrían contener información relevante acerca de la misma.

Lo anterior nos hace concluir que ha mayor número de muestras mejor será la estimación de la señal original que se obtendrá. Aunque es importante hacer notar que el decaimiento del valor del MSE conforme aumenta el número de muestras, no ocurren en la misma proporción, ya que en la mayoría de los casos si se compara el valor de MSE con que se inicia el análisis (128 muestras) hasta llegar a las 2048 muestras la ventana de variación del MSE es bastante amplia; no así en el rango de las 2048 a 4096 muestras, donde el decaimiento se hace mucho más lento.

Si ahora analizamos el comportamiento de los métodos basados en el tipo de Wavelet que se utilizó para llevar a cabo los análisis, se puede afirmar que existe una gran brecha entre la eficiencia de los algoritmos cuando se hace uso de la *Wavelet Coiflet 5* y cuando se hace

Capítulo 3. Reducción de ruido en señales utilizando Wavelets

uso de la *Wavelet Haar*. Este fenómeno obedece principalmente a la cantidad de momentos de desvanecimiento con que cuenta cada una de las Wavelets, así pues, la *Coiflet 5* presenta 10 momentos de desvanecimiento, la *Symlet 5* y *Daubechies 5* presenta 5 momentos de desvanecimiento y la *Haar* con un solo momento de desvanecimiento. Siempre en esta misma línea, se puede hacer notar que el comportamiento de las Wavelets no dicta por completo el valor del MSE de la señal recuperada.

| Umbral Duro | | | |
|-------------|--------|--------------|----------|
| Señal | Método | Wavelet | MSE |
| Cuadrada | MM | Coiflet 5 | 0.000874 |
| Ondas | FT | Daubechies 5 | 0.000070 |
| Cresta | RS | Symlet 5 | 0.000324 |
| Bloques | FT | Symlet 5 | 0.000605 |
| Picos | MM | Coiflet 5 | 0.000882 |
| Ángulos | RS | Coiflet 5 | 0.000096 |
| Seno | MM | Coiflet 5 | 0.000102 |
| Esquinas | RS | Daubechies 5 | 0.000054 |

Tabla 3.4: Mejores estimaciones Wavelet-Método aplicando el Umbral Duro

| Umbral Suave | | | |
|--------------|--------|------------------------|----------|
| Señal | Método | Wavelet | MSE |
| Cuadrada | MM/RS | Coiflet 5 | 0.000875 |
| Ondas | FT/MM | Daubechies 5/Coiflet 5 | 0.000071 |
| Cresta | FT | Symlet 5 | 0.000338 |
| Bloques | MM | Coiflet 5 | 0.000603 |
| Picos | MM | Coiflet 5 | 0.000887 |
| Ángulos | FT/MM | Daubechies 5/Coiflet 5 | 0.000103 |
| Seno | RS | Coiflet 5 | 0.000110 |
| Esquinas | MM | Daubechies 5 | 0.000051 |

Tabla 3.5: Mejores estimaciones Wavelet-Método aplicando el Umbral Suave

Ya que como se puede percibir de las tablas 3.4 y 3.5, la eficacia del trabajo reside en la combinación método – wavelet que se utilice. En ellas se observa que el algoritmo que se presenta con mayor frecuencia es el *Minimax*; mientras que la wavelet que aparece con mayor frecuencia es la *Coiflet 5* gracias a las características ya antes mencionadas. Otra pareja que arroja resultados interesantes es la combinación *Fixed form Threshold* – *Daubechies 5*, las cuales aparecen con cierta regularidad en las tablas 3.4 y 3.5.

Capítulo 3. Reducción de ruido en señales utilizando Wavelets

Al revisar los resultados del apéndice B y en lo que respecta a los valores de umbral obtenidos, podemos afirmar que en los métodos *Minimax* y *Fixed form Threshold*, el valor del umbral aumenta conforme se incrementa el número de muestras. Mientras que el método *Rigorous Sure* presenta un comportamiento diferente, en algunos casos disminuye con el aumento del número de muestras y en otros casos se mantiene oscilando alrededor de un valor específico. También es de hacer notar que el tipo de umbral que se utilice en *Minimax* y *Fixed form Threshold*, ya sea *suave* o *duro*, no interviene en el valor del umbral obtenido. Más bien, dicho valor depende del número de muestras y de la Wavelet que se haya elegido para realizar el análisis, de esta forma se tiene que para la misma cantidad de muestras pero con una diferente Wavelet el umbral es diferente. Caso contrario ocurre con el método *Rigorous Sure*, el cual utiliza un umbral diferente para cada nivel de descomposición.

En la figura 3.9 que se muestra a continuación, se pueden apreciar las gráficas de las mejores estimaciones equivalentes a la tabla 3.4 para el umbral duro.

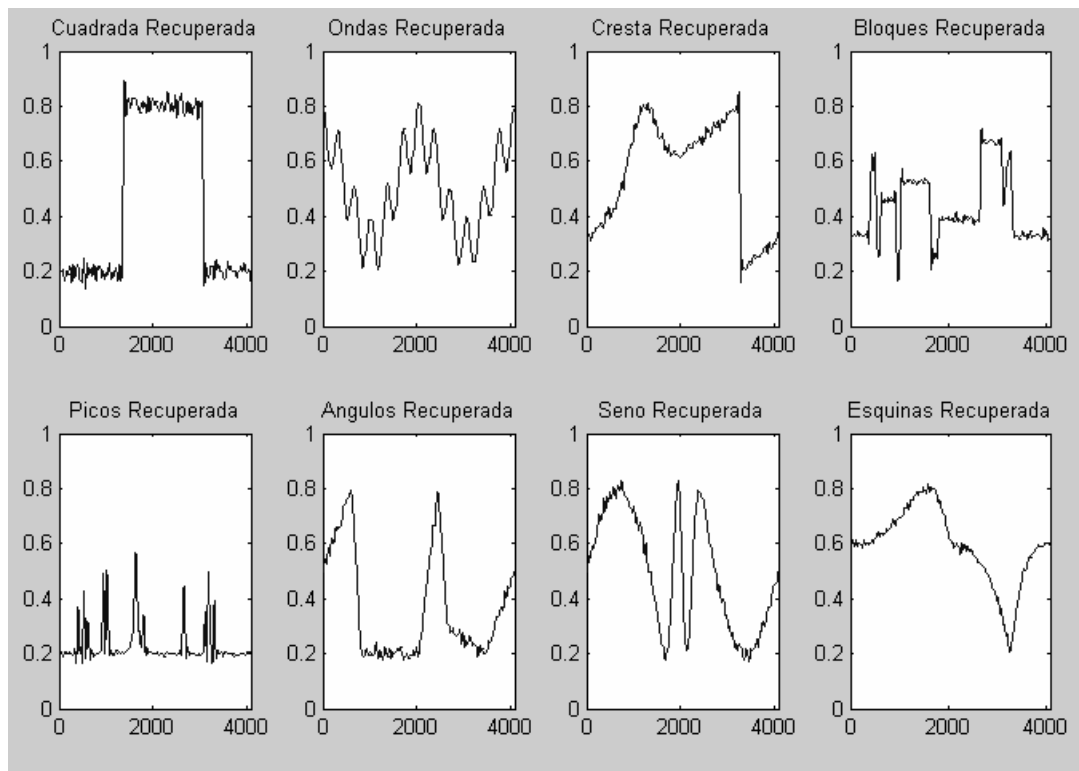


Fig. 3.9: Representación gráfica de mejores estimaciones – Umbral duro

Por otra parte, en la figura 3.10 se presentan las representaciones gráficas del análisis correspondiente a la tabla 3.5 para el umbral suave.

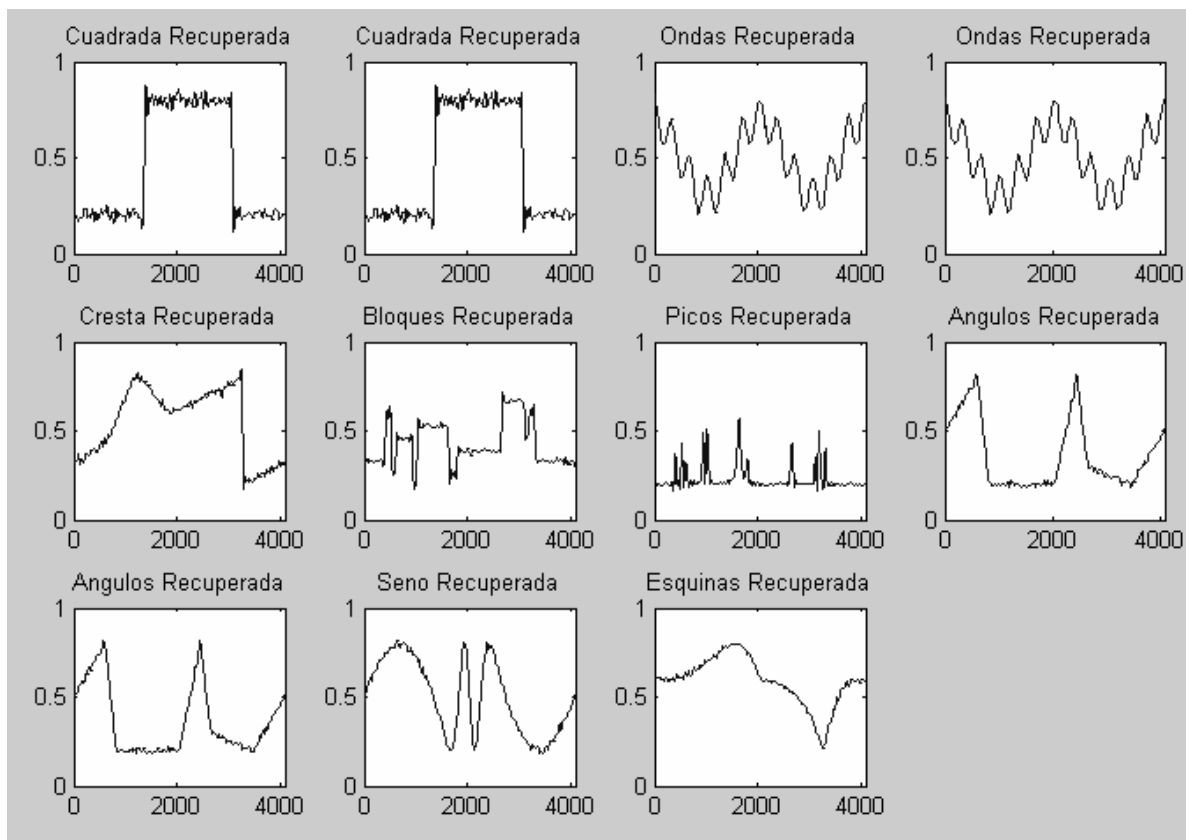


Fig. 3.10: Representación gráfica de mejores estimaciones – Umbral Suave

Basándose en los resultados numéricos obtenidos (tablas 3.4 y 3.5) y en las representaciones gráficas (figuras 3.9 y 3.10), podemos afirmar que en general el comportamiento de las simulaciones ha sido bastante bueno, principalmente como ya se menciono antes, en aquellas simulaciones en que el número de muestras era superior a las 512. Así pues, dichos resultados han demostrado que las estimaciones obtenidas se acercan por mucho a la señal original en la mayor parte de los casos, sin embargo estos resultados podrían verse mejorados si se toma en cuenta el factor de escala o estimador de nivel de ruido (σ) propuesto por la ecuación 3.15.

Capítulo IV

Desarrollo de aplicación de Wavelets basada en Visual Basic 6.0

4.1 Introducción

Ya en el Capítulo 3 se presentó la forma en que MATLAB®, auxiliándose del *Wavelet Toolbox*®, es capaz de introducirnos al estudio y posterior desarrollo de aplicaciones basadas en Wavelets para los interesados en el procesamiento de señales e imágenes, entre otras muchas áreas de las ciencias aplicadas. Como se pudo observar, MATLAB® cuenta con un gran número de librerías pre-diseñadas para implementar diversas funciones que son de mucha utilidad al trabajar con Wavelets. Algunas de las cuales se utilizaron en el presente trabajo (que pueden ser estudiadas con mayor detenimiento en el Apéndice A de este mismo documento) con el fin de desarrollar una aplicación que fuera capaz de demostrar la aplicación de los Wavelets en la limpieza de señales con ruido (ruido blanco) haciendo uso de diversos algoritmos.

Sin embargo, como el lector podrá observar, estas demostraciones solo son de utilidad si se cuenta con MATLAB® para poder ejecutarlas. Es por ello, que en el presente capítulo, se presentan los procedimientos que se llevaron a cabo para desarrollar una pequeña aplicación que sea capaz de demostrar a los usuarios interesados en este tema, el comportamiento de los Wavelets en la limpieza de señales con ruido. Desarrollo para el cual se utilizó Visual Basic 6.0 como plataforma de programación y que podrá ser ejecutada sin importar si el interesado cuenta o no con MATLAB® ejecutándose en su computadora. Dicha aplicación tiene la finalidad de interactuar con el usuario retándolo a que se modifiquen algunos parámetros y que el usuario pueda ver las modificaciones que sufre una señal en base a ellos.

Así pues, en las secciones siguientes se describirá en detalle las consideraciones que se tomaron en cuenta a la hora del diseño de la aplicación, así como también, su uso.

4.2 Transformada Rápida Wavelet y banco de filtros

En la sección 2.6.3 se definió de manera muy general la forma de realizar el cálculo de la Transformada Wavelet Discreta, un punto que es de vital importancia al implementar la limpieza de ruido en señales eléctricas.

En la presente sección, se estudiará con mayor detalle la forma en que la DWT es calculada, ello con el fin de brindar al lector las bases para la comprensión de los algoritmos implementados en la aplicación.

En la ecuación 2.34 se puede observar la expresión matemática correspondiente a la DWT, sin embargo, esta representación no es de mucha utilidad al momento de trabajar con señales discretas; ya que no se pueden observar de una forma adecuada las sumatorias involucradas en este proceso. Razón por la cual en la ecuación 4.1 se presenta la expresión correspondiente de la DWT pero haciendo uso de la notación correcta para j y k .

$$f(t) = \sum_{k=0}^{2^{j_0}-1} c_{j_0,k} \phi_{j_0,k}(t) + \sum_{j=j_0}^{N-1} \sum_{k=0}^{2^j-1} d_{j,k} \psi_{j,k}(t) \quad (4.1)$$

En donde $c_{j_0,k}$ son los coeficientes de escala o de aproximación, $d_{j,k}$ representa a los coeficientes wavelet o del detalle y j_0 representa el espacio de menor resolución. El factor 2^N expresa la longitud de la señal discreta $f(n)$, el cual limita el nivel de descomposición de una señal, ya que no tendría sentido representar una señal que se encuentra en un espacio V_{j_0} en ese mismo espacio.

Como puede intuirse de la ecuación 4.1, el gran tamaño de los cálculos matemáticos necesarios para obtener la DWT de una señal de interés, hace necesario buscar una manera más eficiente de llevar a cabo este procedimiento. Es por ello que se hace necesario conocer y comprender la teoría de banco de filtros, ya que esta nos guiará a la obtención de la Transformada Rápida Wavelet; la cual a su vez nos facilita en gran medida el diseño de algoritmos computacionales rápidos y eficientes.

Sin embargo, el desarrollo de la teoría de bancos de filtros y el diseño de rápidos algoritmos no está relacionado en forma directa con las funciones escala (aproximación) y wavelet (detalle), sino más bien con los coeficientes relacionados a estas funciones. Así pues, el primer paso en la descomposición de la señal será determinar los coeficientes $c_{n-1,k}$ y $d_{n-1,k}$ en términos de $c_{n,k}$. Es decir, obtener la representación de los coeficientes escala y wavelets en un nivel de resolución más bajo en términos de los coeficientes escala en un nivel más alto de resolución. Dichos coeficientes quedan completamente definidos por las ecuaciones 4.2 y 4.3 respectivamente que se detallan a continuación:

$$c_{n-1,k} = \sum_m h(m-2k)c_{n,m} \quad (4.2)$$

$$d_{n-1,k} = \sum_m h_1(m-2k)c_{n,m} \quad (4.3)$$

Las operaciones matemáticas expresadas en las ecuaciones 4.2 y 4.3 corresponden a una convolución discreta entre la secuencia de entrada descrita por $c_{n,k}$; y los factores h y h_1 ; ello con la finalidad de obtener por un lado una representación más “suave” de la señal original caracterizada por los coeficientes escala (ecuación 4.2), y por otra parte el detalle de la señal definido por los coeficientes wavelets (ecuación 4.3). El motivo por el cual esto ocurre, se debe a que los factores h y h_1 actúan como filtros digitales (pasa-bajo y pasa-alto respectivamente). Sin embargo, al aplicar dicha operación sobre una señal real, se presenta el caso que nuestros datos de salida estarán comprendidos por el doble de los datos de entrada. Es decir, si a la entrada de nuestro sistema se tiene una señal de 1024 muestras obtendremos una aproximación y un detalle de la señal original, cada uno con una longitud igual a 1024 muestras también. Con la finalidad de solventar dicho inconveniente, una vez realizada la convolución discreta sobre el conjunto de datos de entrada, se efectúa una operación denominada *submuestreo*, de la cual también se habla en la sección 2.6.3, y que consiste en tomar una señal de entrada x_n y producir una señal de salida $y_n = x_{2n}$, descargando luego únicamente aquellos valores de índice impar. El procedimiento anteriormente descrito se puede observar a la perfección en la figura 4.1, al cual se le conoce como: *Descomposición de una señal ó Análisis*.

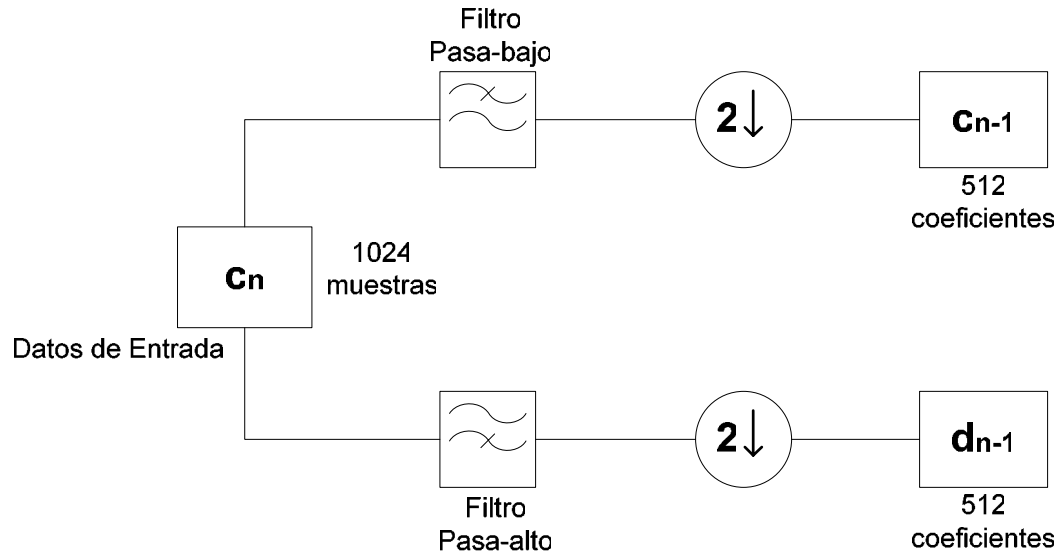


Fig. 4.1: Proceso de Descomposición Wavelet. Banco de Filtro de dos canales

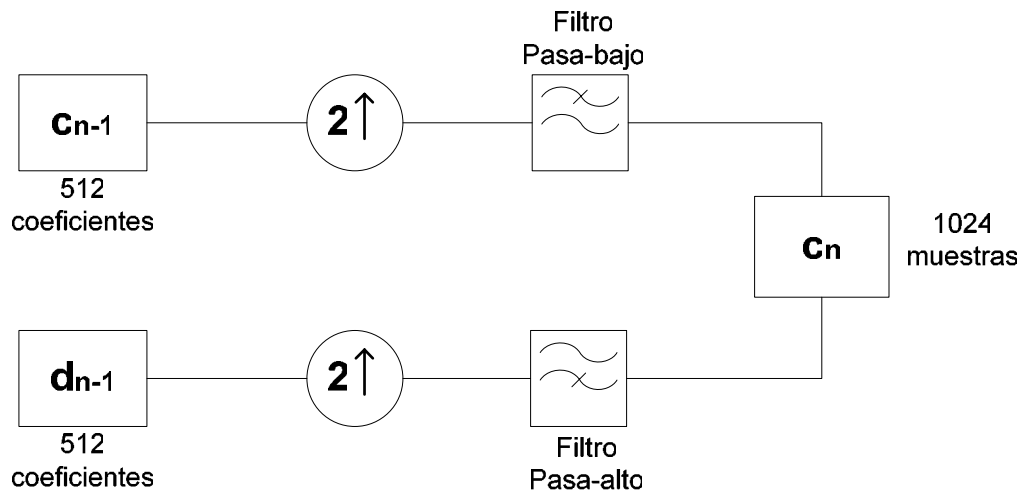


Fig. 4.2: Proceso de Reconstrucción Wavelet. Banco de Filtro de dos canales

El procedimiento que se describió en los párrafos anteriores es únicamente la mitad del camino, ya que lo que faltaría es la recuperación de la señal original sin pérdida de información a partir de las componentes obtenidas durante el análisis. A dicho procedimiento de recuperación se le llama también *síntesis* y corresponde a la inversa de la Transformada Discreta Wavelet (IDWT). Es decir, lo que se desea obtener es una representación de los coeficientes escala en un nivel de resolución más alto mediante la

combinación de los coeficientes escala y wavelets en un nivel de resolución más bajo. Procedimiento definido por la ecuación 4.4 que se muestra a continuación:

$$c_{n,k} = \sum_m c_{n-1,m} h(2m-k) + \sum_m d_{n-1,m} h_1(2m-k) \quad (4.4)$$

Ahora bien, así como durante el análisis se realiza una operación de filtrado y submuestreo, en la operación de síntesis ó reconstrucción se lleva a cabo un *sobremuestreo* (definido también en la sección 2.6.3) y posteriormente un filtrado. Tal y como se definió antes, el *sobremuestreo* inserta ceros entre cada muestra, con el fin de incrementar al doble la longitud de los componentes de entrada (coeficientes escala y wavelets provenientes de la sección de análisis); con ello se garantiza que la señal obtenida después del filtrado tenga la misma longitud que la señal original, este otro proceso se describe gráficamente en la figura 4.2

4.2.1 Múltiples niveles de descomposición – reconstrucción

Los procedimientos que se describieron en la sección anterior, conjuntamente se conocen como banco de filtros de dos canales. Sin embargo, debido a que dichos procesos son iterativos, ellos fácilmente pueden extenderse teóricamente de forma infinita, con la única restricción de que el proceso de síntesis depende del análisis.

Por razones obvias, en la práctica estos procesos no pueden repetirse infinitamente, siendo el nivel de resolución de la señal original la que fija el límite. Es decir: si se supone una señal original con una longitud $N = 2^n$, en el análisis dividimos la señal original en una aproximación y un detalle correspondientes al primer nivel de descomposición de longitud $N/2$, luego la aproximación de longitud 2^{n-1} es nuevamente dividida obteniendo una nueva aproximación y detalle correspondientes a un segundo nivel de descomposición. Este procedimiento se vuelve a repetir hasta que la aproximación y el detalle están representados por un solo coeficiente, es decir una longitud $2^0 = 1$; esto implica que el número de iteraciones posibles a realizar es de $n = \log_2 N$. De esta manera se obtiene un vector de longitud N que contiene un solo término encargado de representar la forma general de la señal (coeficiente escala) y todos los otros términos contienen información sobre el detalle

en los diferentes niveles de descomposición (coeficientes wavelets) tal y como se ilustra en la figura 4.3.

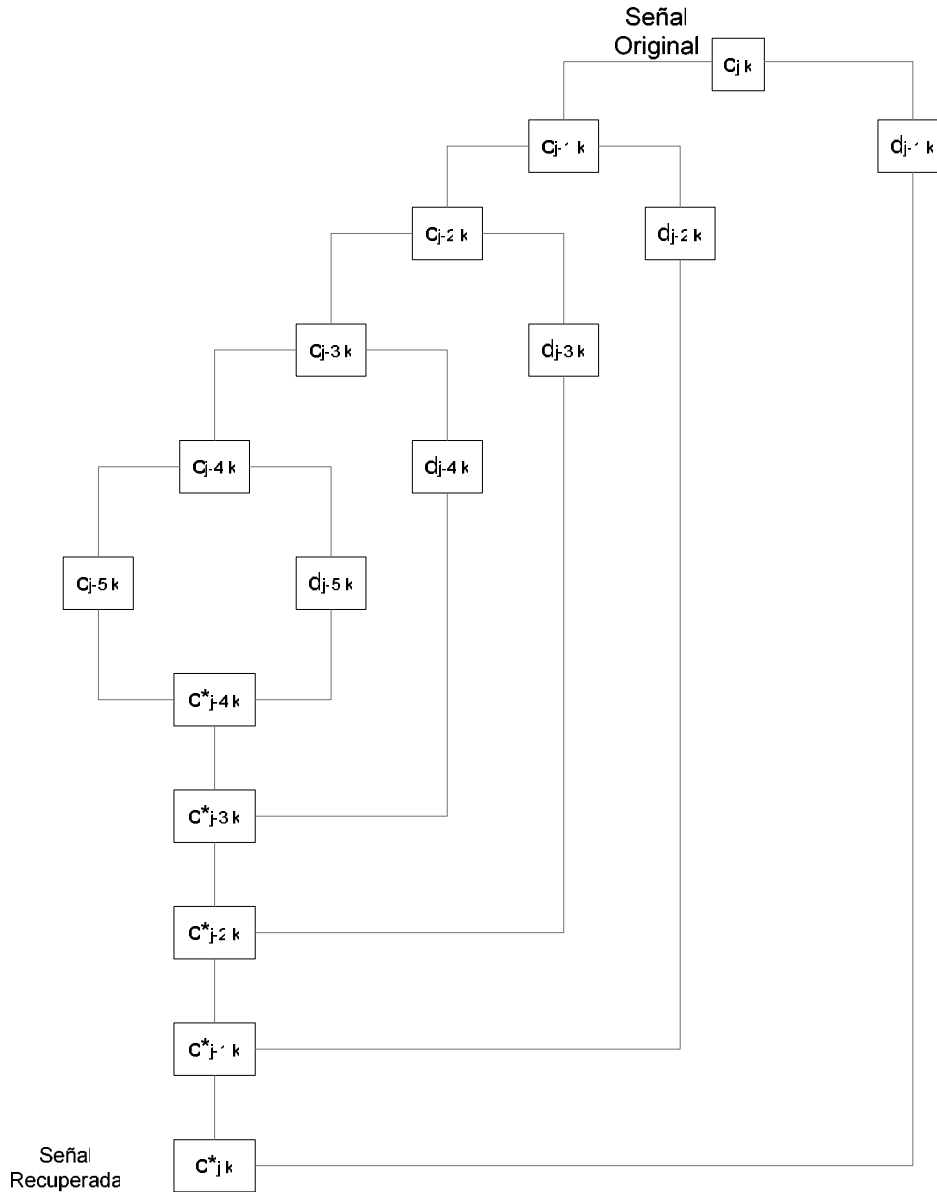


Fig. 4.3: Banco de filtros de 5 niveles de descomposición

Por su parte, el proceso de síntesis toma la aproximación y el detalle, aumenta su longitud mediante el sobremuestreo y realiza la convolución con los respectivos filtros, obteniéndose como resultado una mejor aproximación a la señal correspondiente al primer nivel de reconstrucción. Por supuesto, el número de veces que se realiza este proceso hasta llegar nuevamente a la señal original depende del grado de descomposición al que se llegó en el

análisis, tal y como se ilustra en la figura 4.3. Al conjunto de ambos procesos de análisis – síntesis se le llamó en la sección 2.6.3 *Codificación Sub-banda*. El algoritmo de codificación sub-banda que se ha implementado en la plataforma de simulación puede observarse en la figura 4.3, en ella se puede observar que el nivel de descomposición de nuestro sistema es igual a 5.

La razón por la cual este apartado se trata con mayor profundidad en el presente capítulo, obedece a la necesidad de que el lector tenga la información más cercana a la hora de consultar los algoritmos desarrollados en Visual Basic 6.0 para implementar la plataforma de simulación.

4.3 Descripción de aplicación desarrollada en Visual Basic 6.0

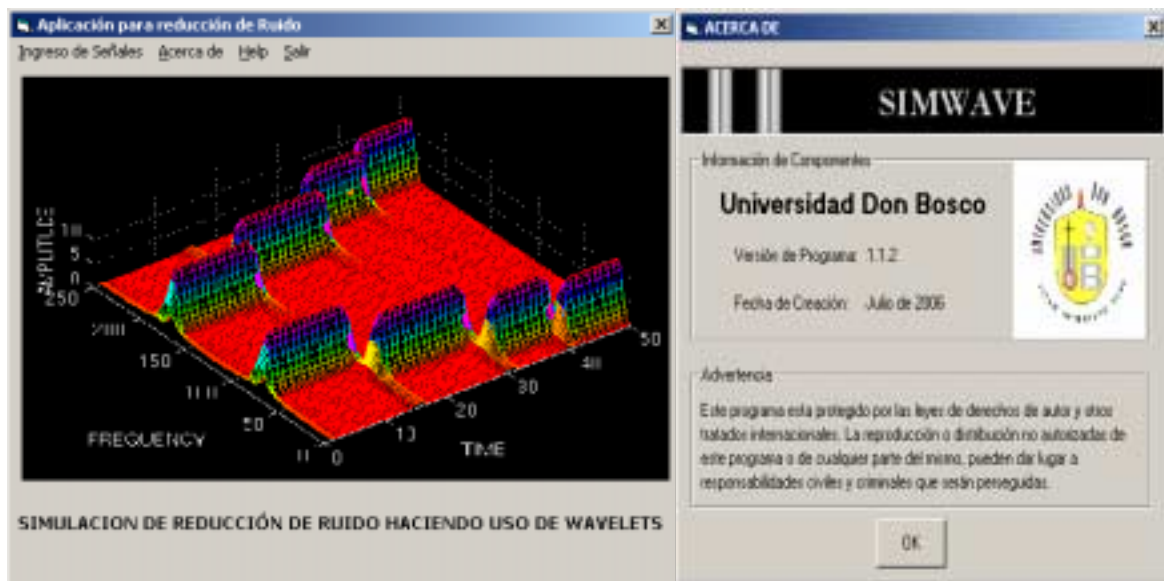


Fig. 4.4: Ventana de bienvenida e informativa de aplicación desarrollada

En la figura 4.4 que se mostró anteriormente se pueden observar la pantalla de bienvenida de la aplicación que fue desarrollada en Visual Basic 6.0. En ella se encuentran cuatro menús principales, los cuales nos permiten conocer la información del programa así como ingresar a la interfase de configuración principal de la aplicación la cual se ilustra en la figura 4.5. En la presente sección se describe de forma detallada cual es el funcionamiento de cada uno de los componentes de la plataforma de simulación. Ello será de utilidad, en

primer lugar, para guiar al usuario en la ejecución de dicha herramienta; y en segundo lugar, al momento de analizar los algoritmos desarrollados en Visual Basic 6.0 ya que en algunos casos hay mucha teoría matemática involucrada; la cual será necesario detallar aquí para fundamentar dichos algoritmos.

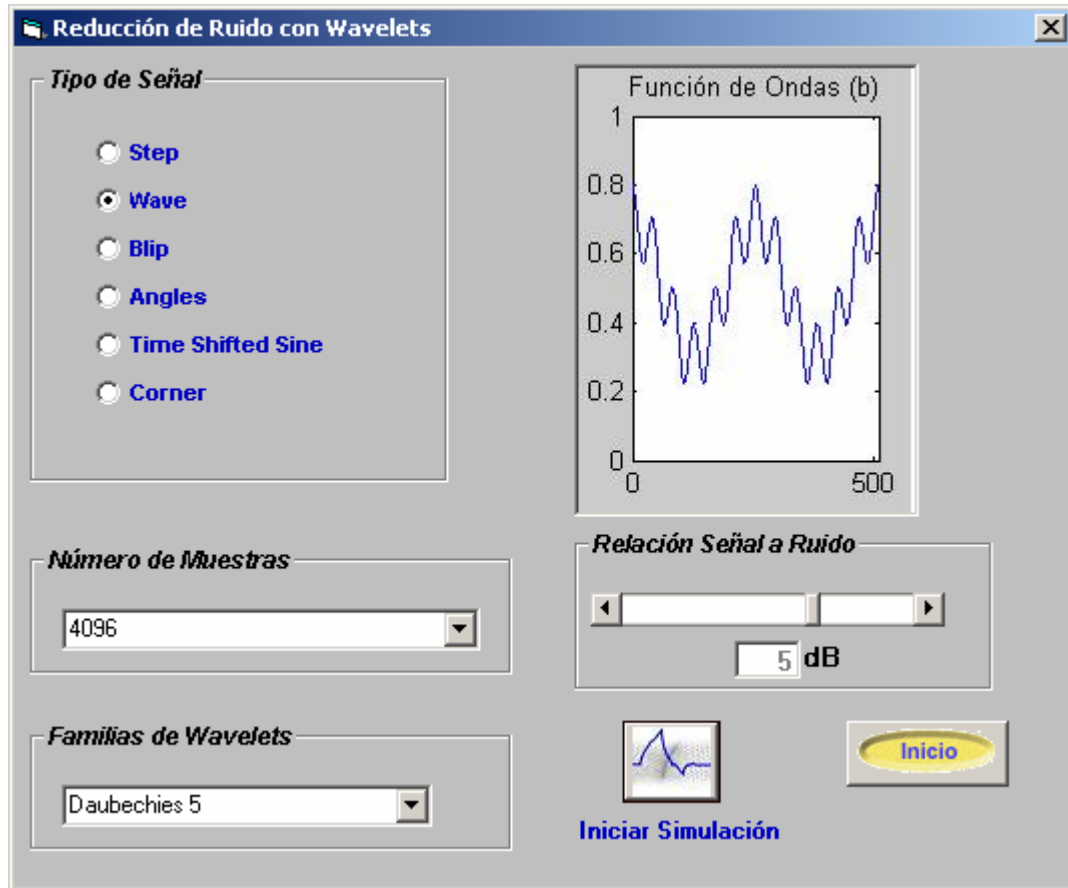


Fig. 4.5: Interfase de configuración principal

En los siguientes apartados de la presente sección, se describirán los aspectos relacionados con cada uno de los parámetros involucrados en la ejecución del programa y que son configurados en la interfase mostrada en la figura 4.5.

4.3.1 Generación de señales de prueba utilizando Visual Basic 6.0

En la interfase que se diseñó, tal y como se puede observar en la figura 4.5, se incluyen las formas de onda sobre las que se realizará el análisis de limpieza de ruido utilizando la teoría de Wavelets; ellas se encuentran completamente definidas en la Sección 3.5.1 y es en base

a estas definiciones que se han desarrollado los algoritmos necesarios para su generación y posterior visualización en la aplicación desarrollada en Visual Basic 6.0. Se ha elegido utilizar dichas formas de onda, debido a que las particularidades de cada una de ellas, examinan características específicas del proceso de Reducción de Ruido utilizando la Teoría de Wavelets y han sido utilizadas en diferentes estudios con este fin específico. Para implementar la visualización de dichas señales de prueba, el parámetro marcado como “Número de muestras” es muy importante; ya que este nos permite establecer cuantas muestras de la señal serán tomadas en cuenta en nuestro análisis; así pues, el usuario tendrá la posibilidad de seleccionar el número de muestras que sea necesario visualizar y puede escoger entre 256, 512, 1024, 2048 y 4096 muestras para ejecutar el análisis de las formas de onda respectivas.

A continuación se enumeran las formas de onda que pueden ser seleccionadas por el usuario en la sección “Tipo de Señal” y sus respectivas expresiones matemáticas; ello con el fin de dar un breve vistazo a las formas de onda que se implementarán, información adicional puede ser revisada en la Sección 3.5.1 del presente documento.

1. *Función Cuadrada.*

$$f_1(x) = 0.2 + 0.6I_{(1/3, 3/4)}(x) \quad (4.5)$$

2. *Función Ondas.*

$$f_2(x) = 0.5 + 0.2\text{Cos}(4\pi x) + 0.1\text{Cos}(24\pi x) \quad (4.6)$$

3. *Función Crestas.*

$$f_3(x) = \left(0.32 + 0.6x + 0.3e^{-100(x-0.3)^2}\right)I_{(0, 0.8)}(x) + \left(-0.28 + 0.6x + 0.3e^{-100(x-1.3)^2}\right)I_{(0.8, 1)}(x) \quad (4.7)$$

4. *Función Ángulos.*

$$\begin{aligned}
 f_6(x) = & (2x + 0.5)I_{(0, 0.15)}(x) + (-12(x - 0.15) + 0.8)I_{(0.15, 0.2)}(x) + 0.2I_{(0.2, 0.5)}(x) \\
 & + (6(x - 0.5) + 0.2)I_{(0.5, 0.6)}(x) + (-10(x - 0.6) + 0.8)I_{(0.6, 0.65)}(x) \\
 & + (-5(x - 0.65) + 0.3)I_{(0.65, 0.85)}(x) + (2(x - 0.85) + 0.2)I_{(0.85, 1)}(x)
 \end{aligned} \tag{4.8}$$

5. *Función Seno Desplazado en el Tiempo.*

$$\begin{aligned}
 f_7(x) = & 0.3\text{Sen}(3\pi(g(g(g(g(x)))))) + x) + 0.5 \\
 g(x) = & \frac{1 - \text{Cos}(\pi x)}{2}
 \end{aligned} \tag{4.9}$$

6. *Función Esquina.*

$$\begin{aligned}
 f_8(x) = & \left(\frac{0.6}{\text{range}(g)} \right) g(x) + 0.6 \\
 g(x) = & 623.87x^3(1 - 4x)I_{(0, 0.5)}(x) + 187.161(0.125 + x^3)x^4I_{(0.5, 0.8)}(x) \\
 & + 3708.470441(x - 1)^3I_{(0.8, 1)}(x)
 \end{aligned} \tag{4.10}$$

En la esquina superior izquierda de la figura 4.6, bajo el título *Señal Original*, se puede observar, a manera de ejemplo, la forma de onda correspondiente al tipo de señal seleccionado en la ventana de configuración ya antes mencionada. En esta figura la aplicación genera la visualización de la señal *ondas* con 4096 muestras de largo. Para generar esta visualización basta con pulsar el botón: “Graficar Señal Original”.

4.3.2 Generación de señal interferente

Como ya se definió en el capítulo 3, la señal interferente será del tipo aleatoria y distribución Gaussiana, la cual es de carácter aditivo, por lo que en la aplicación diseñada la ecuación 3.1 continua siendo valida en nuestro algoritmo.

El patrón aleatorio de la señal interferente se obtiene a través de la utilización de la función *randomize* de Visual Basic 6.0; la cual se encarga de generar un patrón de números aleatorios entre 0 y 1 basándose en el reloj del sistema de la PC que se este utilizando para

ejecutar la aplicación. Esto nos garantiza que los valores que la señal interferente pueda tomar realmente se asemejen a lo que es el ruido blanco.

La distribución de tipo Normal ó Gaussiana se logra a través de la implementación en el algoritmo de su correspondiente expresión matemática, la cual se muestra a continuación:

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (4.11)$$

Vale la pena mencionar que en cuanto a este parámetro, el usuario únicamente tiene control sobre la Relación Señal a Ruido que desea ajustar; y ello se logra al establecer el valor requerido en la sección “Relación Señal a Ruido” mediante la barra de desplazamiento de la ventana de configuración, la cual puede tomar tanto valores positivos como negativos. La visualización de la forma de onda de la señal original mezclada con la señal de ruido puede observarse nuevamente consultando la figura 4.6, bajo el título *Señal Original y Ruido Aditivo*; la cual se genera al presionar el botón “Graficar Señal Original y Ruido Aditivo” y puede observarse en la esquina superior derecha del formulario correspondiente.

4.3.3 Selección de la familia Wavelet a utilizar en el análisis

Tal y como se vio en secciones anteriores, la selección de la familia Wavelet es de vital importancia a la hora de llevar a cabo el cálculo de la Transformada Wavelet Discreta, la cual a su vez es utilizada para realizar la limpieza de las señales eléctricas cuando estas se ven influenciadas por interferencias, para nuestro caso ruido blanco.

Los algoritmos de las familias Wavelet que aquí se estudiarán, son las que se implementarán en la aplicación, por lo que es de interés definir con mayor precisión la forma en que se lleva a cabo el cálculo de la Transformada Rápida Wavelet que ya se estudió en la sección 4.2. Las familias Wavelet que el usuario puede seleccionar desde la sección “Familias de Wavelets” en la pantalla de configuración principal son: *Haar*, *Daubechies* y *Coiflets*. Cuyas características y representaciones gráficas ya fueron discutidas con anterioridad en la sección 2.7. En este apartado se demostrará como es que

cada una de dichas familias se puede utilizar para realizar el cálculo de la DWT y la IDWT conjuntamente con la teoría de banco de filtros.

En la teoría de banco de filtros se ha establecido que se hace necesario la existencia de un filtro pasa-bajos, representado por h ; y un filtro pasa-altos, representado por h_1 en cada nivel de descomposición durante el proceso de análisis – síntesis. Estos factores o coeficientes no son más que los valores discretos de la función escala (h) y la función wavelet (h_1) de cada una de las Familias de Funciones Wavelets. Es de recordar que la cantidad de dichos coeficientes rige la cantidad de momentos de desvanecimiento de cada función wavelet y el grado de la misma.

A continuación se describe la forma en que cada una de las familias wavelets que serán implementadas en la presente aplicación realiza el proceso de cálculo de la DWT e IDWT. La principal diferencia entre cada una de las familias radica en el valor de los coeficientes escala y wavelet (h y h_1 respectivamente) que se utilizan en algoritmo de codificación sub-banda que se ilustra en la figura 4.3.

a) Wavelet Haar

En primer lugar, es necesario conocer el valor de los coeficientes escala (h), llamados también coeficientes de aproximación; y de los coeficientes Wavelet (h_1) conocidos también como los coeficientes del detalle de la Familia de funciones Haar. Dichos coeficientes pueden observarse en la ecuación 4.12 y han sido obtenidos de [12].

$$\begin{aligned} h(0) = h(1) &= \frac{1}{\sqrt{2}} \\ h_1(0) &= \frac{1}{\sqrt{2}}; \quad h_1(1) = -\frac{1}{\sqrt{2}} \end{aligned} \tag{4.12}$$

Ahora bien, ya conociendo los valores de h y h_1 , estos son convolucionados con la secuencia de entrada discreta $c_{n,k}$, y de esta manera obtener en primer lugar una representación más “suave” de la señal original, la cual es descrita por los coeficientes escala $c_{n-1,k}$. Y a la vez, el detalle de la señal, el cual es descrito por los coeficientes

Wavelet $d_{n-1,k}$. Así pues, estos dos tipos de coeficientes quedan definidos como se observan en las ecuaciones 4.13 y 4.14 respectivamente.

$$c_{J-1,k} = \frac{1}{\sqrt{2}}(c_{J,2k} + c_{J,2k+1}) \quad (4.13)$$

$$d_{J-1,k} = \frac{1}{\sqrt{2}}(c_{J,2k} - c_{J,2k+1}) \quad (4.14)$$

Como se puede observar en las dos ecuaciones anteriores, la aproximación esta relacionada con el promedio, mientras que el detalle esta relacionado con las diferencias. En ambos casos estos coeficientes son obtenidos a partir de las N muestras de la señal original. Otro punto de interés en las ecuaciones 4.13 y 4.14 es que luego de llevar a cabo la función de convolución con los filtros h y h_I , la secuencia de entrada es submuestreada de tal forma que se obtengan dos conjuntos de coeficientes con una longitud correspondiente a la mitad de las muestras de la señal original.

Luego, se mantienen los coeficientes Wavelet (detalle) y se realiza el mismo procedimiento descrito anteriormente con los coeficientes escala (aproximación). Este procedimiento se repite hasta cierto nivel de descomposición, para el diseño de la presente aplicación se ha seleccionado el nivel de descomposición en 5. Con el procedimiento descrito anteriormente, se ha obtenido la Transformada Wavelet Discreta de nuestra señal de entrada.

Para realizar la reconstrucción de la señal a partir de los coeficientes escala y wavelet pertenecientes al primer nivel de descomposición, basta con sumar y restar las ecuaciones 4.13 y 4.14, de donde se obtienen las expresiones 4.15 y 4.16, que describen los coeficientes pertenecientes a la reconstrucción.

$$c_{J,2k} = \frac{1}{\sqrt{2}}(c_{J-1,k} + d_{J-1k}) \quad (4.15)$$

$$c_{J,2k+1} = \frac{1}{\sqrt{2}}(c_{J-1,k} - d_{J-1,k}) \quad (4.16)$$

Como se puede observar en las dos ecuaciones anteriores y en la figura 4.3, a partir de los coeficientes escala y wavelet correspondientes a un mismo nivel de descomposición se obtienen los coeficientes de aproximación recuperados en un nivel de descomposición superior, lo cual permite una reconstrucción perfecta de la señal.

b) Wavelet Daubechies

Esta familia Wavelet fue nombrada así en honor a su inventora, la matemática Ingrid Daubechies. Tanto en esta familia como en la *Coiflets*, que se estudiará a posteriormente, los filtros h y h_1 representan la cantidad de momentos de desvanecimiento de su función escala y wavelet respectivamente; por lo que para el caso que aquí se estudia, se ha seleccionado a la Wavelet Daubechies 5 para llevar a cabo el algoritmo de descomposición y reconstrucción. Por lo que, esta familia tendrá 10 coeficientes escala (h) y 10 coeficientes wavelet (h_1), lo que arroja un total de 20 coeficientes para dicha familia. La expresión que relaciona estos dos tipos de coeficientes, para una señal finita como es nuestro caso, se puede observar en la ecuación 4.17.

$$h_1(n) = (-1)^n h(N-1-n) \quad (4.17)$$

Así pues, en la ecuación anterior se puede observar como obtener los coeficientes wavelet dados los coeficientes escala. Los coeficientes Daubechies escala y wavelets normalizados han sido obtenidos de [12] y se presentan en la Tabla 4.1.

Cada uno de los coeficientes escala y wavelet mostrados en la Tabla 4.1 son convolucionados con los datos de la señal de entrada que se esta analizando. Es decir, si el conjunto de datos de entrada es N , las funciones escala y wavelet calculadas tendrán una longitud de $N/2$ cada una de ellas. Representando los coeficientes de escala nuevamente la suavidad de la señal analizada. Mientras que los coeficientes wavelet representan las diferencias o cambios que ocurren en los datos de entrada. En el primer nivel de descomposición estos datos de entrada es la señal original; mientras que en los siguientes

Capítulo 4. Desarrollo de aplicación de Wavelets basada en Visual Basic 6.0

niveles de descomposición representan a los coeficientes escala del nivel de descomposición anterior.

| <i>Daubechies 5</i> | |
|----------------------------------|--------------------------------------------|
| Coefficientes escala h | Coefficientes Wavelet h_1 |
| $h(0) = 1.60102 \times 10^{-1}$ | $h_1(0) = h(9) = 3.33572 \times 10^{-3}$ |
| $h(1) = 6.03829 \times 10^{-1}$ | $h_1(1) = -h(8) = 1.25807 \times 10^{-2}$ |
| $h(2) = 7.24308 \times 10^{-1}$ | $h_1(2) = h(7) = -6.24149 \times 10^{-3}$ |
| $h(3) = 1.38428 \times 10^{-1}$ | $h_1(3) = -h(6) = -7.75714 \times 10^{-2}$ |
| $h(4) = -2.42294 \times 10^{-1}$ | $h_1(4) = h(5) = -3.22448 \times 10^{-2}$ |
| $h(5) = -3.22448 \times 10^{-2}$ | $h_1(5) = -h(4) = 2.42294 \times 10^{-1}$ |
| $h(6) = 7.75714 \times 10^{-2}$ | $h_1(6) = h(3) = 1.38428 \times 10^{-1}$ |
| $h(7) = -6.24149 \times 10^{-3}$ | $h_1(7) = -h(2) = -7.24308 \times 10^{-1}$ |
| $h(8) = -1.25807 \times 10^{-2}$ | $h_1(8) = h(1) = 6.03829 \times 10^{-1}$ |
| $h(9) = 3.33572 \times 10^{-3}$ | $h_1(9) = -h(0) = -1.60102 \times 10^{-1}$ |

Tabla 4.1: Coeficientes escala y wavelet para Familia Daubechies 5

Las ecuaciones que describen este comportamiento de forma general, para el primer nivel de descomposición, son las que se describen a continuación:

$$c_{J-1,k} = h(0)c_{J,2k-1} + h(1)c_{J,2k} + h(2)c_{J,2k+1} \dots + h(5)c_{J,2k+4} \quad (4.18)$$

$$d_{J-1,k} = h_1(0)c_{J,2k-1} + h_1(1)c_{J,2k} + h_1(2)c_{J,2k+1} \dots + h_1(9)c_{J,2k+8} \quad (4.19)$$

Las ecuaciones 4.18 y 4.19 describen el procedimiento para realizar el cálculo de la DWT haciendo uso de la Familia Wavelet Daubechies 5. Como se puede observar en ambas ecuaciones, con cada iteración el índice k se incrementa en dos con el fin de calcular el nuevo coeficiente escala y wavelet.

La Transformada Inversa Wavelet es calculada en base a los N elementos obtenidos de la DWT; en donde los primeros $N/2$ elementos del arreglo son obtenidos a partir de los

coeficientes escala, mientras que los siguientes $N/2$ elementos se calculan a partir de los coeficientes wavelets.

Sin importar que en este caso se trate de la Familia Daubechies, el algoritmo ilustrado en la figura 4.3 se sigue cumpliendo tanto para el cálculo de la DWT e IDWT; la principal diferencia es que en este caso, al igual que ocurre en la familia Coiflet, el procedimiento del cálculo de los coeficientes se trata de una matriz que consta de $N/2$ filas; y su cálculo se vuelve más complejo que en el sistema Haar. Esto debido a que en el sistema Haar únicamente se contaba con dos coeficientes.

c) Wavelet Coiflets

Esta familia Wavelet fue diseñada con la finalidad de mantener una muy buena semejanza entre los valores estimados y los valores de la señal original. Esta familia es muy similar a la familia Daubechies que se estudió anteriormente. La única diferencia entre ellas es el valor de los coeficientes escala y wavelet para la familia Coiflets, los cuales pueden ser observados en la tabla 4.2 que se muestra a continuación. Un dato que hay que resaltar es que la relación descrita por la ecuación 4.17 continúa teniendo validez para este caso.

| <i>Coiflets 3</i> | |
|----------------------------------|--------------------------------------------|
| Coeficientes escala h | Coeficientes Wavelet h_1 |
| $h(0) = -7.27326 \times 10^{-2}$ | $h_1(0) = h(5) = -1.56557 \times 10^{-2}$ |
| $h(1) = 3.37898 \times 10^{-1}$ | $h_1(1) = -h(4) = 7.27326 \times 10^{-2}$ |
| $h(2) = 8.52572 \times 10^{-1}$ | $h_1(2) = h(3) = 3.84865 \times 10^{-1}$ |
| $h(3) = 3.84865 \times 10^{-1}$ | $h_1(3) = -h(2) = -8.52572 \times 10^{-1}$ |
| $h(4) = -7.27326 \times 10^{-2}$ | $h_1(4) = h(1) = 3.37898 \times 10^{-1}$ |
| $h(5) = -1.56557 \times 10^{-2}$ | $h_1(5) = -h(0) = 7.27326 \times 10^{-2}$ |

Tabla 4.2: Coeficientes escala y wavelet para Familia Coiflets 3

Así pues, la familia Coiflet que se implementará en nuestra plataforma será la Coiflet 3 cuyo conjunto de filtros están definidos en la tabla 4.2.

4.3.4 Algoritmo de reducción de ruido implementado

El algoritmo sobre el que se basa nuestra plataforma de simulación y el cual *no* es un parámetro que el usuario pueda seleccionar; es el que en la sección 3.4.2 fue denominado como Fixed form Threshold y cuya expresión matemática es:

$$\lambda_{UNI} = \sigma \sqrt{2 \log(N)} \quad (4.20)$$

Las características de este algoritmo pueden encontrarse en este mismo documento en la sección anteriormente mencionada. Aquí es importante señalar que este cálculo del umbral es un paso intermedio entre el cálculo de la DWT, correspondiente a la señal original más señal interferente, y el cálculo de la IDWT. Ya que este valor de umbral tiene la tarea de eliminar aquellos coeficientes que en esencia son puro ruido. En este método de reducción de ruido se propone que el nivel de ruido de los datos de entrada se calcule como la media absoluta de los coeficientes wavelet obtenidos en el primer nivel de descomposición dividida por 0.6745; ello debido a que los coeficientes wavelet en dicho nivel de descomposición son principalmente ruido. Este estimador de nivel de ruido ya se presentó en la ecuación 3.15, sin embargo se trae aquí nuevamente para refrescar la memoria del lector.

$$\sigma = \frac{1}{n} \sum_k \frac{|d_{J-1,k}|}{0.6745} \quad (4.21)$$

4.3.5 Visualización de resultados del análisis

En la figura 4.6, en la esquina inferior izquierda, se pueden observar los resultados de aplicar los conceptos antes definidos en las secciones 4.3.3 y 4.3.4. En esta gráfica se visualiza la forma de onda de la Señal Recuperada la cual es obtenida a partir de la forma de onda de la señal con el ruido blanco mezclado. Para el ejemplo que aquí se presenta, la familia seleccionada para realizar el análisis es la *Daubechies 5* y haciendo uso del algoritmo *Fixed form Threshold*.

Capítulo 4. Desarrollo de aplicación de Wavelets basada en Visual Basic 6.0

Además de los resultados del tipo gráfico, que son de mucha utilidad al momento de evaluar el nivel de eficiencia del algoritmo de reducción de ruido que aquí se presenta, ya que se tiene una apreciación de la pérdida o no de las particularidades de la forma de onda que se está estudiando, en la aplicación se hace uso del cálculo del Error Cuadrático Medio, el cual describe el nivel de semejanza que existe entre la señal original de entrada y la señal recuperada, ello con la finalidad de describir de manera estadística la eficacia del algoritmo en realizar su trabajo de limpieza/reducción del nivel de ruido en señales unidimensionales.

Al presionar el botón “Graficar Señal Recuperada”, que se muestra en la figura 4.6, se genera la visualización correspondiente a la *Señal Recuperada* haciendo uso de todos los parámetros ya definidos en la ventana de configuración.

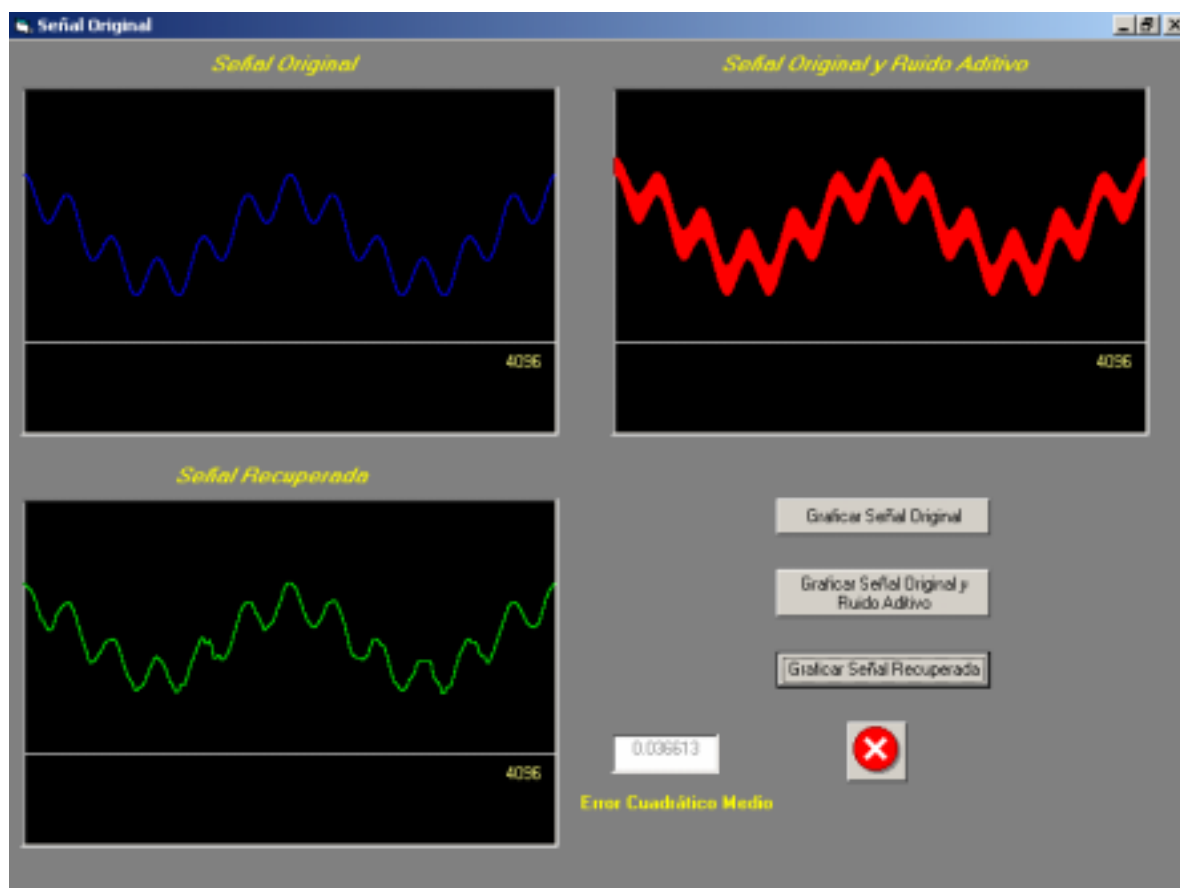


Fig. 4.6: Resultados obtenidos con aplicación diseñada en VB 6.0 para la limpieza/reducción de ruido con Wavelets

Capítulo 4. Desarrollo de aplicación de Wavelets basada en Visual Basic 6.0

Como puede observarse, la interfase gráfica que fue diseñada cumple con el objetivo de ser una aplicación amigable para el usuario. Ya que únicamente tendrá que introducir los parámetros necesarios del análisis en la ventana de configuración principal y luego ejecutar, según su conveniencia, cada una de las simulaciones para poder observar la forma en que funcionan los Wavelets en el proceso de Reducción de Ruido.

Conclusiones

- La Teoría de Wavelets es una muy interesante herramienta alternativa al clásico análisis de Fourier en el procesamiento digital de señales e imágenes; principalmente en aquellos casos en los cuales el análisis de Fourier presenta ciertas limitantes debido a su formulación.
- Debido a la gran variedad de Familias Wavelet existentes, de las cuales algunas de ellas fueron estudiadas en el presente trabajo de graduación, se puede observar que ellas presentan una gran versatilidad al momento de realizar diversos análisis del comportamiento y particularidades de las señales; ello gracias a las propiedades de ortogonalidad, ortonormalidad, momentos de desvanecimiento, etc. Que cada una de las familias presenta.
- En el análisis de señales no estacionarias la Transformada de Fourier no es óptima; ya que aunque entrega una información completa del contenido espectral de la señal, no es capaz de localizar en el tiempo las componentes de frecuencia.
- En el análisis Wavelet, la función escala es la encargada de analizar el comportamiento general de la señal, mientras que la función wavelet se encarga de analizar el comportamiento del detalle de la señal.
- La Transformada Wavelet Discreta permite generar un análisis multiresolución sobre una señal discreta. El análisis multiresolución se refiere al proceso iterativo de convoluciones entre la señal discreta de interés y coeficientes que actúan como filtros pasa-bajos y pasa-altos, provenientes de las familias wavelets, de forma tal que la señal discreta es descompuesta obteniendo información sobre las características generales de la señal y sobre las características del detalle de la señal en forma separada.

Conclusiones

- Con cada nivel de descomposición implementado en el cálculo de la Transformada Wavelet Discreta, el número de coeficientes correspondientes a cada uno de los niveles de descomposición superiores, se ven truncados a la mitad de los coeficientes correspondientes al nivel de descomposición anterior.
- La utilización de la Transformada Wavelet Discreta en el análisis de señales, permite la fácil implementación computacional, ello gracias al desarrollo de algoritmos muy eficientes y rápidos para el cálculo de los coeficientes. Ya que los cálculos se reducen únicamente a operaciones de suma y multiplicación entre la señal objeto de estudio y los filtros que se utilizan.
- Luego de finalizadas un conjunto de 960 simulaciones, correspondientes a la limpieza de ruido en señales, utilizando MATLAB® y su *Wavelet Toolbox*® se ha llegado a la conclusión de que en realidad no existe un único método que presente los mejores resultados para analizar todos los tipos de señales que aquí se presentaron. Más bien se encontró que el desempeño de cada uno de los métodos de reducción/limpieza de ruido depende en gran medida del tipo de señal que se analice, la wavelet que se escoja, el nivel de descomposición utilizado y el número de muestras.
- El tipo de Wavelet que mejores resultados arrojó fue la Wavelet *Coiflet 5* gracias a sus diez momentos de desvanecimiento, lo cual puede demostrarse gracias a que en las tablas resumen de las mejores estimaciones aparece un total de diez veces combinada con diversos algoritmos.
- Si se compara el valor del umbral del algoritmo *Fixed form Threshold* y del *Minimax*, se notará en general que el umbral del primero siempre es mayor para un mismo número de muestras. Por lo que el primero ofrece mejores resultados, ya que elimina un mayor número de coeficientes que pueden ser en esencia puro ruido.

Conclusiones

- El valor del umbral en los algoritmos *Fixed form Threshold* y *Minimax* dependen únicamente de la wavelet utilizada en el análisis y del número de muestras, acá no es de importancia si se trata de umbral suave o umbral duro o del tipo de señal que se este analizando.
- El algoritmo Rigorous SURE es un método de mayor complejidad que el resto de los estudiados. Esto se refleja en que el cálculo del umbral depende del tipo de umbral, la señal de estudio, la wavelet y del número de muestras.
- Del estudio realizado con MATLAB® se puede observar que en el umbral suave los cambios entre un valor de umbral para un número de muestras determinado es más lento y el valor del umbral es menor; si es que se le compara con el caso del umbral duro en las mismas condiciones de análisis.
- Con la implementación práctica, a través del uso de MATLAB®, se respaldó la teoría de Wavelets y su aplicación en la reducción de ruido, la cual fue presentada de forma teórica a lo largo del presente documento.
- Al momento de implementar la plataforma de simulación, utilizando Visual Basic 6.0, se hizo necesario desarrollar los algoritmos para la DWT e IDWT hasta el quinto nivel de descomposición; en donde se pudo observar que en el caso de la Wavelet Haar dicho algoritmo presenta menor complejidad si se le compara con los algoritmos de las Wavelet Daubechies y Coiflets implementados también en la aplicación.
- La plataforma de simulación que se ha desarrollado en Visual Basic 6.0 tiene un fin principalmente didáctico para la demostración de una alternativa que se tiene con los Wavelets en el análisis de señales y específicamente en la reducción/limpieza de ruido en señales

Recomendaciones

- Con el fin de implementar de una manera práctica los conceptos y demostraciones planteadas en el presente documento, se podría diseñar un sistema electrónico que sea capaz de captar desde un transductor alguna magnitud física y transmitirla remotamente y posteriormente procesarla y recuperarla limpia en el extremo receptor validando alguno de los algoritmos presentados.
- En las tecnologías inalámbricas que tanta pujanza han tenido en nuestro medio en los últimos tiempos, la teoría de wavelets podría ser utilizada a fin de comprobar su eficacia en la recuperación de señales y compararlos con los métodos tradicionalmente utilizados.
- Tal y como se ha mencionado incontables ocasiones a lo largo del presente documento, la Teoría de Wavelets es una alternativa válida a la utilización tradicional de la Teoría de Fourier en el ámbito del procesamiento de señales. Por lo que sería de interés realizar un estudio comparativo de índole práctico entre ambas Teorías.

Apéndice A

Programas implementados en MATLAB®

En el presente apéndice se presentan todos los algoritmos desarrollados en MATLAB® que fueron de utilidad para realizar todos los cálculos y resultados gráficos que en este trabajo de graduación se muestran.

Cada uno de los programas esta basado fundamentalmente en funciones preestablecidas del *Wavelet Toolbox*®, por lo que en dichos programas únicamente se hace el llamado a las funciones que automáticamente llevan a cabo una tarea bien definida, como es el caso de *wden*, *thselect*, *wavedec*, etc. Funciones que en este apéndice no se da mayor detalle ya que ellas pueden ser consultadas en la sección de *ayuda (help)* de Matlab®.

El programa principal de la interfase desarrollada es el script *Denoising.m*, el cual se encarga de calcular de manera automática la señal recuperada y presentarla tanto gráfica como matemáticamente, a través del cálculo del Error Cuadrático Medio (MSE) y del cálculo del Umbral para cada uno de los algoritmos de reducción de ruido estudiados. Este script se encarga de hacer el llamado a cada una de las funciones necesarias de manera sucesiva para llevar a cabo el conjunto de simulaciones.

La función *ConstruirSignal.m* y el script *GenerarSignal.m*, trabajan en conjunto para dar forma a las funciones sintéticas de prueba que se utilizan en nuestras diversas simulaciones. Aquí también se le solicita al usuario que introduzca los parámetros necesarios (Relación Señal a Ruido, Número de muestras, Familia Wavelet, Algoritmo de reducción de ruido y el tipo de umbral) para cada uno de los análisis que se realizarán.

a) Función ConstruirSignal.m

```
function sig = ConstruirSignal(nombre_signal,n)
% ConstruirSignal: Genera una señal artificial
% Entradas:
%     nombre_signal string: 'Step','Wave','Blip','Blocks','Bumps'
%     n     Longitud de la señal deseada. Número de muestras
% Salidas:
%     sig   Señal artificial

%A continuación se generan cada una de las funciones sintéticas a utilizar:

t = (1:n)/n;
if strcmp(nombre_signal,'Step'),
    sig = 0.2 + 0.6*(t>1/3&t<=0.75);

elseif strcmp(nombre_signal,'Wave'),
    sig = 0.5 + (0.2.*cos(4*pi*t)) + (0.1.*cos(24*pi*t));

elseif strcmp(nombre_signal,'Blip'),
    sig = (0.32 + (0.6.*t) + 0.3*exp(-100*((t-0.3).^2))).*(t<=0.8) + ...
        (-0.28 + (0.6.*t) + 0.3*exp(-100*((t-1.3).^2))).*(t>0.8);

elseif strcmp(nombre_signal,'Blocks'),
    pos = [.1 .13 .15 .23 .25 .40 .44 .65 .76 .78 .81];
    hgt = [4 (-5) 3 (-4) 5 (-4.2) 2.1 4.3 (-3.1) 2.1 (-4.2)];
    sig = 2*ones(size(t));
    for j=1:length(pos)
        sig = sig + (1 + sign(t-pos(j))).*(hgt(j)/2);
    end
    sig = (0.6/9.2)*sig + 0.2;

elseif strcmp(nombre_signal,'Bumps'),
    pos = [.1 .13 .15 .23 .25 .40 .44 .65 .76 .78 .81];
    hgt = [4 5 3 4 5 4.2 2.1 4.3 3.1 5.1 4.2];
    wth = [.005 .005 .006 .01 .01 .03 .01 .01 .005 .008 .005];
    sig = zeros(size(t));
    for j=1:length(pos)
        sig = sig + hgt(j)./(1 + (abs(t-pos(j))./wth(j)).^4);
    end
    sig = ((0.6/5.3437952)*sig) + 0.2;

elseif strcmp(nombre_signal,'Angles'),
    sig = ((2*t + 0.5).*(t<=0.15)) + ...
        ((-12*(t-0.15) + 0.8).*(t>0.15 & t <= 0.2)) + ...
        0.2*(t>0.2 & t<=0.5) + ...
        ((6*(t-0.5) + 0.2).*(t>0.5 & t<=0.6)) + ...
```

```
((-10*(t-0.6) + 0.8).*(t>0.6 & t<=0.65)) + ...
((-0.5*(t-0.65) + 0.3).*(t>0.65 & t<=0.85)) + ...
((2*(t-0.85) + 0.2).*(t>0.85));

elseif strcmp(nombre_signal,'Time Shifted Sine'),
    u=t;
    for j=1:4,
        u = 0.5*(1-cos(pi*u));
    end
    sig = 0.3*sin(3*pi*(u+t)) + 0.5;

elseif strcmp(nombre_signal,'Corner'),
    sig = t;
    sig(t<=0.5) = 62.387.*10.*t(t<=0.5).^3.*(1-4.*t(t<=0.5).^2);
    sig(0.5<t & t<=0.8) = 62.387.*3.*(0.125-t(0.5<t & t<=0.8).^3).*t(0.5<t & t<=0.8).^4;
    sig(t>0.8) = 62.387.*59.443.*(t(t>0.8)-1).^3;
    sig = (0.6/range(sig)).*sig + 0.6;

else
    disp('ConstruirSeñal: I don*t recognize');nombre_signal
    disp('Las funciones permitidas son:')
    disp('Step'),
    disp('Wave'),
    disp('Blip'),
    disp('Blocks'),
    disp('Bumps'),
    disp('Angles'),
    disp('Time Shifted Sine'),
    disp('Corner'),
end
```

b) GenerarSignal.m

```
clear
close all

% Se solicita al usuario que introduzca los siguientes valores:
%Número de muestras
%Relación Señal a Ruido
%Familia Wavelet (WaveletName): 'Haar', 'db5', 'coif5', 'sym5'
%Método de reducción de ruido (Algoritmo): 'sqtwolog', 'rigrsure', 'minimaxi'
%Tipo de Umbral a emplear (TipoUmbral): 's', 'h'

n = input('Introduzca el número de muestras de la señal');
SNR = input('Introduzca la Relación Señal a Ruido');
WaveletName = input('Introduzca la Wavelet');
Algoritmo = input('Introduzca el algoritmo');
TipoUmbral = input('Introduzca el tipo de umbral');

%Se generan las señales de prueba
Signal_Cuadrada = ConstruirSignal('Step',n);
Signal_Ondas = ConstruirSignal('Wave',n);
Signal_Cresta = ConstruirSignal('Blip',n);
Signal_Bloques = ConstruirSignal('Blocks',n);
Signal_Picos = ConstruirSignal('Bumps',n);
Signal_Angulos = ConstruirSignal('Angles',n);
Signal_Seno = ConstruirSignal('Time Shifted Sine',n);
Signal_Esquinas = ConstruirSignal('Corner',n);

%Se grafican las señales de prueba
subplot(2,4,1)
plot(Signal_Cuadrada,axis([0 n 0 1]),title('Función Cuadrada (a)');
subplot(2,4,2)
plot(Signal_Ondas,axis([0 n 0 1]),title('Función de Ondas (b)');
subplot(2,4,3)
plot(Signal_Cresta,axis([0 n 0 1]),title('Función de Cresta (c)');
subplot(2,4,4)
plot(Signal_Bloques,axis([0 n 0 1]),title('Función de Bloques (d)');
subplot(2,4,5)
plot(Signal_Picos,axis([0 n 0 1]),title('Función Picos (e)');
subplot(2,4,6)
plot(Signal_Angulos,axis([0 n 0 1]),title('Función Angulos (f)');
subplot(2,4,7)
plot(Signal_Seno,axis([0 n 0 1]),title('Función Seno (g)');
subplot(2,4,8)
plot(Signal_Esquinas,axis([0 n 0 1]),title('Función Esquina (h)');
```

Apéndice A

%Se calcula la señal de ruido blanco Gaussiano para cada una de las señales de prueba

```
stdruido1 = std(Signal_Cuadrada)/SNR;  
epsilon1 = randn(size(Signal_Cuadrada))*stdruido1;  
stdruido2 = std(Signal_Ondas)/SNR;  
epsilon2 = randn(size(Signal_Ondas))*stdruido2;  
stdruido3 = std(Signal_Cresta)/SNR;  
epsilon3 = randn(size(Signal_Cresta))*stdruido3;  
stdruido4 = std(Signal_Bloques)/SNR;  
epsilon4 = randn(size(Signal_Bloques))*stdruido4;  
stdruido5 = std(Signal_Picos)/SNR;  
epsilon5 = randn(size(Signal_Picos))*stdruido5;  
stdruido6 = std(Signal_Angulos)/SNR;  
epsilon6 = randn(size(Signal_Angulos))*stdruido6;  
stdruido7 = std(Signal_Seno)/SNR;  
epsilon7 = randn(size(Signal_Seno))*stdruido7;  
stdruido8 = std(Signal_Esquinas)/SNR;  
epsilon8 = randn(size(Signal_Esquinas))*stdruido8;
```

%Se mezcla la señal de prueba con la señal de Ruido Blanco Gaussiano

```
Cuadrada_Ruido = Signal_Cuadrada + epsilon1;  
Ondas_Ruido = Signal_Ondas + epsilon2;  
Cresta_Ruido = Signal_Cresta + epsilon3;  
Bloques_Ruido = Signal_Bloques + epsilon4;  
Picos_Ruido = Signal_Picos + epsilon5;  
Angulos_Ruido = Signal_Angulos + epsilon6;  
Seno_Ruido = Signal_Seno + epsilon7;  
Esquinas_Ruido = Signal_Esquinas + epsilon8;
```

%Se grafican cada una de las señales de prueba con el ruido blanco mezclado

```
figure(2)  
subplot(2,4,1)  
plot(Cuadrada_Ruido,axis([0 n 0 1]),title('Cuadrada con Ruido'));  
subplot(2,4,2)  
plot(Ondas_Ruido,axis([0 n 0 1]),title('Ondas con Ruido'));  
subplot(2,4,3)  
plot(Cresta_Ruido,axis([0 n 0 1]),title('Cresta con Ruido'));  
subplot(2,4,4)  
plot(Bloques_Ruido,axis([0 n 0 1]),title('Bloques con Ruido'));  
subplot(2,4,5)  
plot(Picos_Ruido,axis([0 n 0 1]),title('Picos con Ruido'));  
subplot(2,4,6)  
plot(Angulos_Ruido,axis([0 n 0 1]),title('Angulos con Ruido'));  
subplot(2,4,7)  
plot(Seno_Ruido,axis([0 n 0 1]),title('Seno con Ruido'));  
subplot(2,4,8)  
plot(Esquinas_Ruido,axis([0 n 0 1]),title('Esquinas con Ruido'));
```

c) Denoising.m

GenerarSignals *%Se hace el llamado al script Generar Signals*

*%Se realiza el cálculo de la Transformada Wavelet Discreta para cada una de las señales
%de prueba*

```
dwtCuadrada = wavedec(Cuadrada_Ruido,n,WaveletName)
dwtOndas = wavedec(Ondas_Ruido,n,WaveletName)
dwtCresta = wavedec(Cresta_Ruido,n,WaveletName)
dwtBloques = wavedec(Bloques_Ruido,n,WaveletName)
dwtPicos = wavedec(Picos_Ruido,n,WaveletName)
dwtAngulos = wavedec(Angulos_Ruido,n,WaveletName)
dwtSeno = wavedec(Seno_Ruido,n,WaveletName)
dwtEsquinas = wavedec(Esquinas_Ruido,n,WaveletName)
```

%Se calcula y visualiza el valor del umbral y la señal recuperada

```
disp('Umbrales de Señal Cuadrada')
thr = thselect(dwtCuadrada,Algoritmo)
Cuadrada_Rec = wden(Cuadrada_Ruido,Algoritmo,TipoUmbral,'one',5,WaveletName);
disp('Umbrales de Señal Ondas')
thr = thselect(dwtOndas,Algoritmo)
Ondas_Rec = wden(Ondas_Ruido,Algoritmo,TipoUmbral,'one',5,WaveletName);
disp('Umbrales de Señal Cresta')
thr = thselect(dwtCresta,Algoritmo)
Cresta_Rec = wden(Cresta_Ruido,Algoritmo,TipoUmbral,'one',5,WaveletName);
disp('Umbrales de Señal Bloques')
thr = thselect(dwtBloques,Algoritmo)
Bloques_Rec = wden(Bloques_Ruido,Algoritmo,TipoUmbral,'one',5,WaveletName);
disp('Umbrales de Señal Picos')
thr = thselect(dwtPicos,Algoritmo)
Picos_Rec = wden(Picos_Ruido,Algoritmo,TipoUmbral,'one',5,WaveletName);
disp('Umbrales de Señal Angulos')
thr = thselect(dwtAngulos,Algoritmo)
Angulos_Rec = wden(Angulos_Ruido,Algoritmo,TipoUmbral,'one',5,WaveletName);
disp('Umbrales de Señal Seno')
thr = thselect(dwtSeno,Algoritmo)
Seno_Rec = wden(Seno_Ruido,Algoritmo,TipoUmbral,'one',5,WaveletName);
disp('Umbrales de Señal Esquinas')
thr = thselect(dwtEsquinas,Algoritmo)
Esquinas_Rec = wden(Esquinas_Ruido,Algoritmo,TipoUmbral,'one',5,WaveletName);
```

%Se grafica la señal recuperada de cada una de las señales de prueba

```
figure(3)
subplot(2,4,1)
plot(Cuadrada_Rec,axis([0 n 0 1]),title('Cuadrada Recuperada'));
subplot(2,4,2)
plot(Ondas_Rec,axis([0 n 0 1]),title('Ondas Recuperada'));
```

Apéndice A

```
subplot(2,4,3)
plot(Cresta_Rec),axis([0 n 0 1]),title('Cresta Recuperada');
subplot(2,4,4)
plot(Bloques_Rec),axis([0 n 0 1]),title('Bloques Recuperada');
subplot(2,4,5)
plot(Picos_Rec),axis([0 n 0 1]),title('Picos Recuperada');
subplot(2,4,6)
plot(Angulos_Rec),axis([0 n 0 1]),title('Angulos Recuperada');
subplot(2,4,7)
plot(Seno_Rec),axis([0 n 0 1]),title('Seno Recuperada');
subplot(2,4,8)
plot(Esquinas_Rec),axis([0 n 0 1]),title('Esquinas Recuperada');

%Se calcula y despliega el valor del Error Cuadrático Medio
disp('Error Cuadrático Medio')
error(1,:) = mse(Signal_Cuadrada,Cuadrada_Rec);
error(2,:) = mse(Signal_Ondas,Ondas_Rec);
error(3,:) = mse(Signal_Cresta,Cresta_Rec);
error(4,:) = mse(Signal_Bloques,Bloques_Rec);
error(5,:) = mse(Signal_Picos,Picos_Rec);
error(6,:) = mse(Signal_Angulos,Angulos_Rec);
error(7,:) = mse(Signal_Seno,Seno_Rec);
error(8,:) = mse(Signal_Esquinas,Esquinas_Rec);

fprintf('%f\n',error)

function ECM = mse(f,y);
k = f-y;
l = k.^2;
ECM = sum(l)/length(f);
```


Apéndice B

Resultados completos de simulaciones utilizando

MATLAB®

En el apéndice B se condensa toda la información obtenida luego de haber realizado el conjunto de simulaciones con todas las combinaciones de variables posibles con ayuda de los algoritmos presentados en el apéndice A. Las 960 simulaciones que aquí se resumen, están agrupadas en primera instancia por el método de reducción de ruido empleado y el tipo de umbral y luego según la familia Wavelet empleada en el análisis.

En cada una de las tablas se pueden consultar el valor del MSE según el tipo de señal de prueba y la cantidad de muestras de la señal. Conjuntamente con las tablas se presentan gráficos comparativos del desempeño de las diversas simulaciones realizadas, ello con el fin de tener una percepción gráfica y matemática del desempeño de dichos métodos de reducción de ruido.

**Algoritmo: Fixed form threshold
Umbral Suave**

| Wavelet Haar | | | | | |
|------------------------|----------|----------|----------|----------|----------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| Umbral (λ) | 3.3219 | 3.7209 | 3.9038 | 4.0780 | 4.2449 |
| | MSE | MSE | MSE | MSE | MSE |
| Señal Cuadrada | 0.019973 | 0.005136 | 0.002731 | 0.001495 | 0.000911 |
| Señal Ondas | 0.02508 | 0.005638 | 0.002231 | 0.000717 | 0.000253 |
| Señal Cresta | 0.015598 | 0.004227 | 0.002842 | 0.001022 | 0.000797 |
| Señal Bloques | 0.013162 | 0.005491 | 0.002668 | 0.002008 | 0.000948 |
| Señal Picos | 0.005761 | 0.003926 | 0.003084 | 0.002361 | 0.001392 |
| Señal Ángulos | 0.020541 | 0.003261 | 0.001306 | 0.00047 | 0.000197 |
| Señal Seno | 0.020497 | 0.00399 | 0.002197 | 0.000734 | 0.000279 |
| Señal Esquinas | 0.006949 | 0.000852 | 0.000262 | 0.00014 | 0.000075 |

Tabla B.1: Resultados de las simulaciones con algoritmo Fixed form Threshold, Umbral Suave, Haar

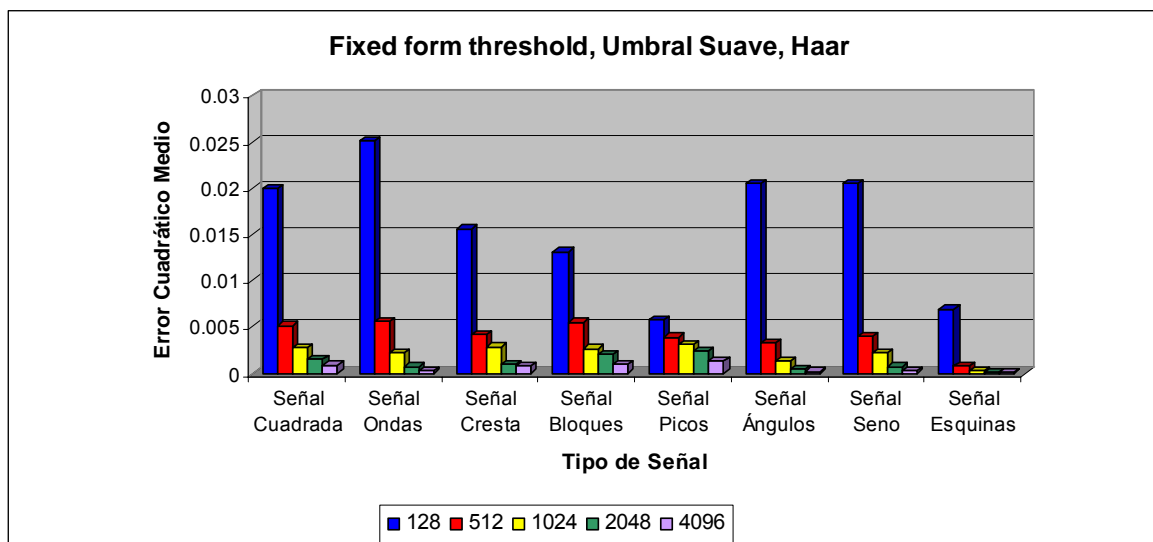


Fig. B.1: Gráfico comparativo de desempeño de las simulaciones con algoritmo Fixed form Threshold, Umbral Suave, Haar

**Algoritmo: Fixed form threshold
Umbral Suave**

| Wavelet Daubechies 5 | | | | | |
|-----------------------------|----------|----------|----------|----------|----------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| Umbral (λ) | 3.7815 | 4.1326 | 4.2972 | 4.4557 | 4.6087 |
| | MSE | MSE | MSE | MSE | MSE |
| Señal Cuadrada | 0.016269 | 0.004346 | 0.003015 | 0.001291 | 0.001036 |
| Señal Ondas | 0.012948 | 0.004956 | 0.00062 | 0.000071 | 0.000091 |
| Señal Cresta | 0.011711 | 0.0034 | 0.00154 | 0.000538 | 0.000344 |
| Señal Bloques | 0.01056 | 0.004401 | 0.002618 | 0.001274 | 0.000627 |
| Señal Picos | 0.005663 | 0.003803 | 0.002667 | 0.002122 | 0.000918 |
| Señal Ángulos | 0.022928 | 0.001766 | 0.000229 | 0.00013 | 0.000103 |
| Señal Seno | 0.018997 | 0.004344 | 0.000346 | 0.000159 | 0.000151 |
| Señal Esquinas | 0.003557 | 0.000119 | 0.000064 | 0.000106 | 0.000067 |

Tabla B.2: Resultados de las simulaciones con algoritmo Fixed form Threshold, Umbral Suave, Daubechies 5

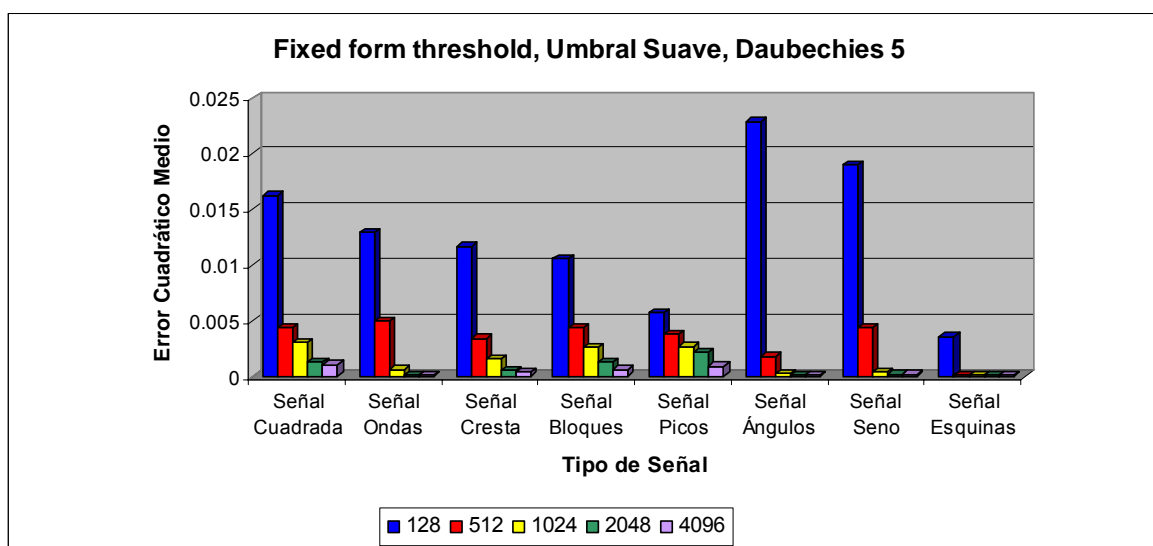


Fig. B.2: Gráfico comparativo de desempeño de las simulaciones con algoritmo Fixed form Threshold, Umbral Suave, Daubechies 5

**Algoritmo: Fixed form threshold
Umbral Suave**

| Wavelet Coiflet 5 | | | | | |
|--------------------------|------------|------------|-------------|-------------|-------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| Umbral (λ) | 4.0626 | 4.3907 | 4.5459 | 4.6959 | 4.8413 |
| | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> |
| Señal Cuadrada | 0.015757 | 0.004524 | 0.002616 | 0.001348 | 0.000884 |
| Señal Ondas | 0.017506 | 0.004931 | 0.0003 | 0.000107 | 0.000095 |
| Señal Cresta | 0.01356 | 0.002697 | 0.001557 | 0.000613 | 0.000407 |
| Señal Bloques | 0.012322 | 0.004771 | 0.002321 | 0.001183 | 0.000612 |
| Señal Picos | 0.005712 | 0.003738 | 0.002817 | 0.002151 | 0.000889 |
| Señal Ángulos | 0.01885 | 0.001104 | 0.000199 | 0.000107 | 0.000131 |
| Señal Seno | 0.018252 | 0.002363 | 0.000242 | 0.000119 | 0.000196 |
| Señal Esquinas | 0.003695 | 0.000155 | 0.000081 | 0.000059 | 0.000058 |

Tabla B.3: Resultados de las simulaciones con algoritmo Fixed form Threshold, Umbral Suave, Coiflet 5

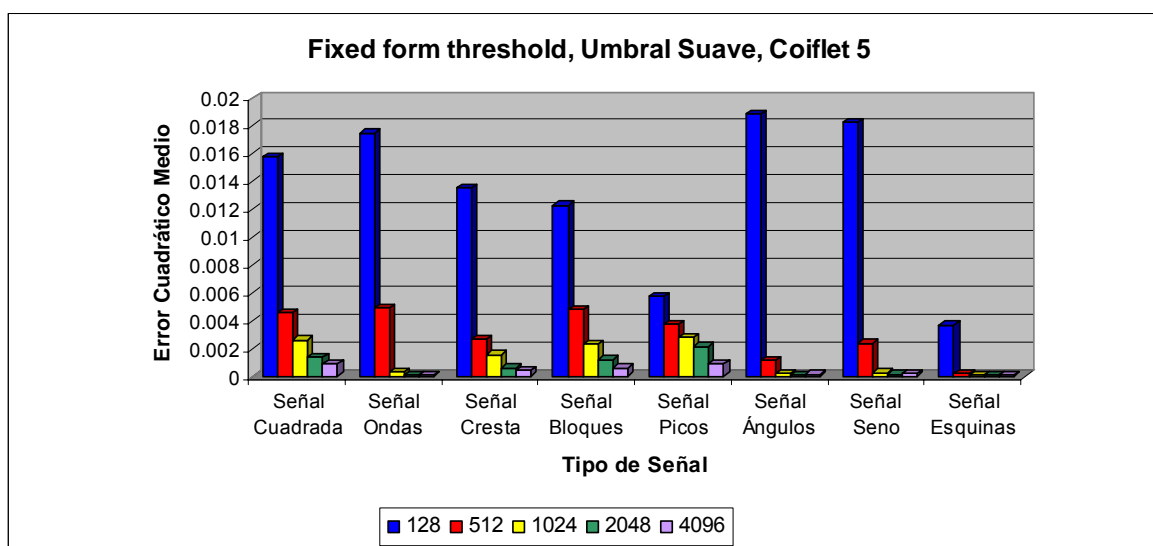


Fig. B.3: Gráfico comparativo de desempeño de las simulaciones con algoritmo Fixed form Threshold, Umbral Suave, Coiflet 5

**Algoritmo: Fixed form threshold
Umbral Suave**

| Wavelet Symlet 5 | | | | | |
|-------------------------|------------|------------|-------------|-------------|-------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| Umbral (λ) | 3.7815 | 4.1326 | 4.2972 | 4.4557 | 4.6087 |
| | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> |
| Señal Cuadrada | 0.016784 | 0.004658 | 0.003044 | 0.001438 | 0.00097 |
| Señal Ondas | 0.007953 | 0.005076 | 0.000709 | 0.000082 | 0.000081 |
| Señal Cresta | 0.010382 | 0.003346 | 0.001335 | 0.000583 | 0.000338 |
| Señal Bloques | 0.009035 | 0.004405 | 0.003105 | 0.001249 | 0.000609 |
| Señal Picos | 0.0056 | 0.003829 | 0.002695 | 0.00212 | 0.000912 |
| Señal Ángulos | 0.025471 | 0.002013 | 0.000198 | 0.000155 | 0.000137 |
| Señal Seno | 0.024996 | 0.006553 | 0.000343 | 0.00014 | 0.000157 |
| Señal Esquinas | 0.003337 | 0.000078 | 0.000088 | 0.00007 | 0.00006 |

Tabla B.4: Resultados de las simulaciones con algoritmo Fixed form Threshold, Umbral Suave, Symlet 5

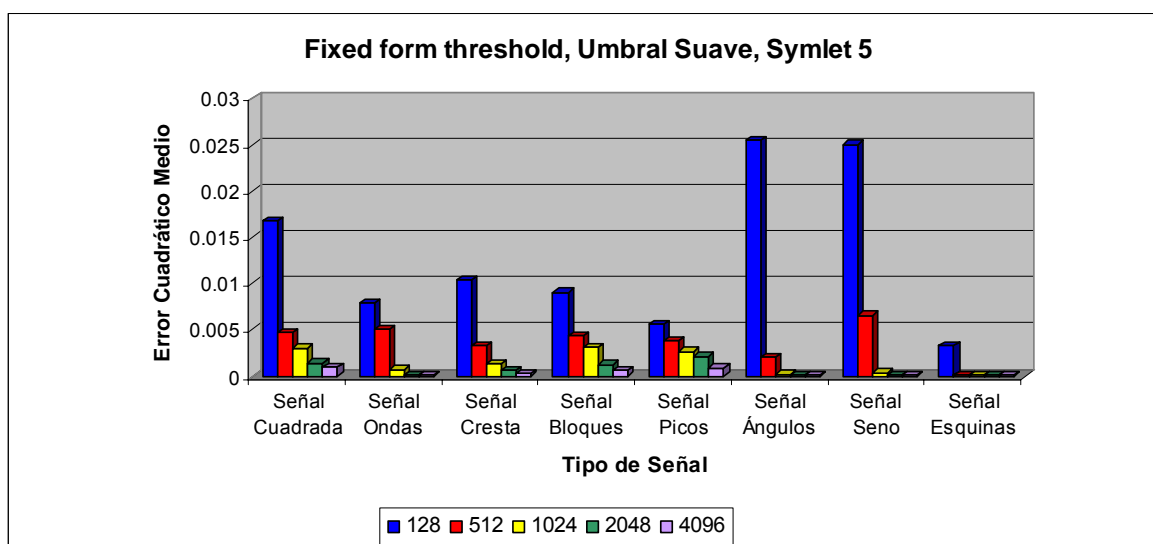


Fig. B.4: Gráfico comparativo de desempeño de las simulaciones con algoritmo Fixed form Threshold, Umbral Suave, Symlet 5

**Algoritmo: Fixed form threshold
Umbral Duro**

| Wavelet Haar | | | | | |
|------------------------|------------|------------|-------------|-------------|-------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| Umbral (λ) | 3.3219 | 3.7209 | 3.9038 | 4.0780 | 4.2449 |
| | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> |
| Señal Cuadrada | 0.019695 | 0.005135 | 0.002761 | 0.001569 | 0.000959 |
| Señal Ondas | 0.025089 | 0.005634 | 0.002278 | 0.000701 | 0.00025 |
| Señal Cresta | 0.015649 | 0.004239 | 0.002901 | 0.000984 | 0.000801 |
| Señal Bloques | 0.013137 | 0.00547 | 0.002696 | 0.001994 | 0.00095 |
| Señal Picos | 0.005799 | 0.003944 | 0.003078 | 0.002363 | 0.001389 |
| Señal Ángulos | 0.020579 | 0.003225 | 0.001317 | 0.000469 | 0.00021 |
| Señal Seno | 0.020459 | 0.003934 | 0.002273 | 0.000801 | 0.000301 |
| Señal Esquinas | 0.006987 | 0.000824 | 0.000267 | 0.000129 | 0.000071 |

Tabla B.5: Resultados de las simulaciones con algoritmo Fixed form Threshold, Umbral Duro, Haar

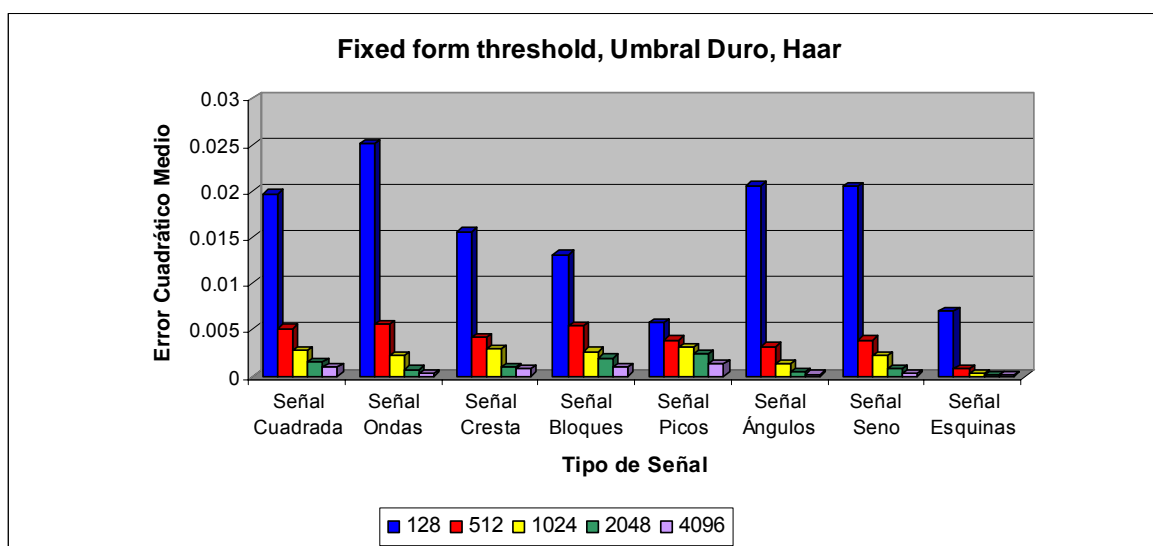


Fig. B.5: Gráfico comparativo de desempeño de las simulaciones con algoritmo Fixed form Threshold, Umbral Duro, Haar

**Algoritmo: Fixed form threshold
Umbral Duro**

| Wavelet Daubechies 5 | | | | | |
|-----------------------------|------------|------------|-------------|-------------|-------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| Umbral (λ) | 3.7815 | 4.1326 | 4.2972 | 4.4557 | 4.6087 |
| | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> |
| Señal Cuadrada | 0.015937 | 0.004317 | 0.003161 | 0.001378 | 0.001001 |
| Señal Ondas | 0.013386 | 0.004899 | 0.00062 | 0.000085 | 0.00007 |
| Señal Cresta | 0.011214 | 0.003206 | 0.001512 | 0.000548 | 0.000361 |
| Señal Bloques | 0.010511 | 0.004388 | 0.002597 | 0.001299 | 0.000611 |
| Señal Picos | 0.005693 | 0.003816 | 0.002663 | 0.002116 | 0.000918 |
| Señal Ángulos | 0.023197 | 0.001814 | 0.000193 | 0.000152 | 0.000137 |
| Señal Seno | 0.018917 | 0.004334 | 0.000307 | 0.000191 | 0.000181 |
| Señal Esquinas | 0.003453 | 0.000145 | 0.000074 | 0.000085 | 0.000071 |

Tabla B.6: Resultados de las simulaciones con algoritmo Fixed form Threshold, Umbral Duro, Daubechies 5

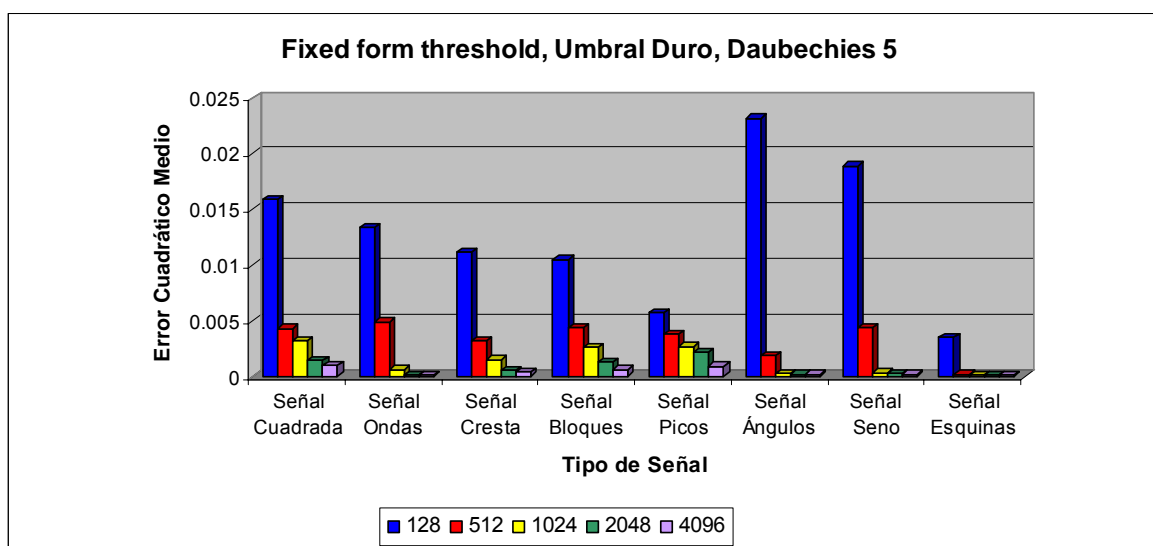


Fig. B.6: Gráfico comparativo de desempeño de las simulaciones con algoritmo Fixed form Threshold, Umbral Duro, Daubechies 5

**Algoritmo: Fixed form threshold
Umbral Duro**

| Wavelet Coiflet 5 | | | | | |
|--------------------------|------------|------------|-------------|-------------|-------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| Umbral (λ) | 4.0626 | 4.3907 | 4.5459 | 4.6959 | 4.8413 |
| | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> |
| Señal Cuadrada | 0.015986 | 0.004343 | 0.00252 | 0.001305 | 0.000889 |
| Señal Ondas | 0.017729 | 0.004986 | 0.000247 | 0.000084 | 0.000074 |
| Señal Cresta | 0.013724 | 0.00268 | 0.001564 | 0.000594 | 0.000438 |
| Señal Bloques | 0.012303 | 0.004793 | 0.002304 | 0.001208 | 0.00061 |
| Señal Picos | 0.005685 | 0.003738 | 0.002824 | 0.002151 | 0.000884 |
| Señal Ángulos | 0.018304 | 0.000987 | 0.000228 | 0.000129 | 0.000128 |
| Señal Seno | 0.017907 | 0.002431 | 0.000169 | 0.000152 | 0.000139 |
| Señal Esquinas | 0.003443 | 0.000167 | 0.000071 | 0.000064 | 0.000066 |

Tabla B.7: Resultados de las simulaciones con algoritmo Fixed form Threshold, Umbral Duro, Coiflet 5

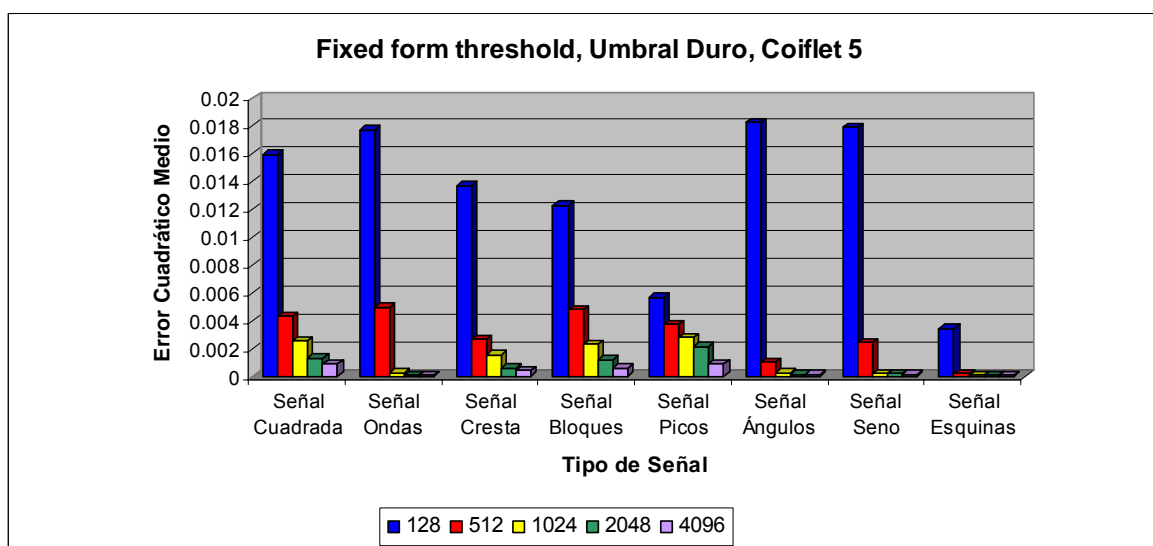


Fig. B.7: Gráfico comparativo de desempeño de las simulaciones con algoritmo Fixed form Threshold, Umbral Duro, Coiflet 5

**Algoritmo: Fixed form threshold
Umbral Duro**

| Wavelet Symlet 5 | | | | | |
|-------------------------|------------|------------|-------------|-------------|-------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| Umbral (λ) | 3.7815 | 4.1326 | 4.2972 | 4.4557 | 4.6087 |
| | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> |
| Señal Cuadrada | 0.016189 | 0.004933 | 0.003004 | 0.00148 | 0.00098 |
| Señal Ondas | 0.007518 | 0.005112 | 0.000653 | 0.000085 | 0.000108 |
| Señal Cresta | 0.010396 | 0.003225 | 0.001349 | 0.000579 | 0.000341 |
| Señal Bloques | 0.009156 | 0.004358 | 0.003129 | 0.00125 | 0.000605 |
| Señal Picos | 0.00561 | 0.003849 | 0.002692 | 0.002122 | 0.000907 |
| Señal Ángulos | 0.025252 | 0.002029 | 0.000195 | 0.000153 | 0.000109 |
| Señal Seno | 0.024328 | 0.006598 | 0.00035 | 0.000155 | 0.000162 |
| Señal Esquinas | 0.003456 | 0.000151 | 0.000082 | 0.000071 | 0.000065 |

Tabla B.8: Resultados de las simulaciones con algoritmo Fixed form Threshold, Umbral Duro, Symlet 5

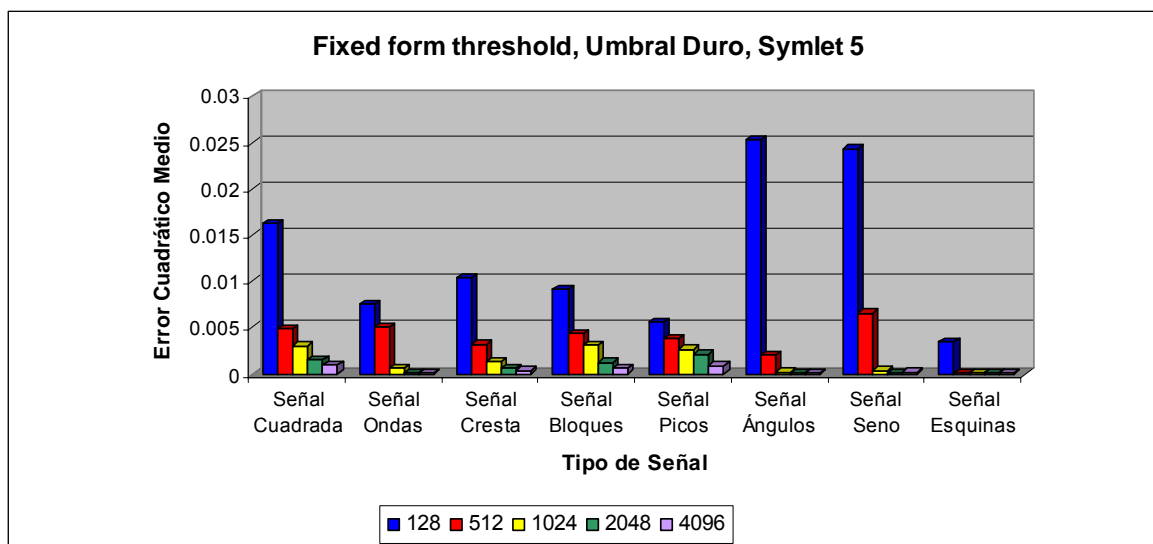


Fig. B.8: Gráfico comparativo de desempeño de las simulaciones con algoritmo Fixed form Threshold, Umbral Duro, Symlet 5

**Algoritmo: Minimax
Umbral Suave**

| Wavelet Haar | | | | | |
|------------------------|----------|----------|----------|----------|----------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| Umbral (λ) | 1.8495 | 2.2203 | 2.4042 | 2.5877 | 2.7709 |
| | MSE | MSE | MSE | MSE | MSE |
| Señal Cuadrada | 0.019658 | 0.005105 | 0.002943 | 0.001406 | 0.000982 |
| Señal Ondas | 0.025016 | 0.005652 | 0.00225 | 0.000704 | 0.00025 |
| Señal Cresta | 0.015649 | 0.004209 | 0.002879 | 0.000979 | 0.000796 |
| Señal Bloques | 0.013194 | 0.005485 | 0.002713 | 0.002 | 0.000955 |
| Señal Picos | 0.005811 | 0.003935 | 0.003076 | 0.002366 | 0.00139 |
| Señal Ángulos | 0.020536 | 0.003192 | 0.001333 | 0.000454 | 0.000215 |
| Señal Seno | 0.020455 | 0.003929 | 0.002195 | 0.00074 | 0.000308 |
| Señal Esquinas | 0.006966 | 0.000813 | 0.000241 | 0.000117 | 0.000066 |

Tabla B.9: Resultados de las simulaciones con algoritmo Minimax, Umbral Suave, Haar

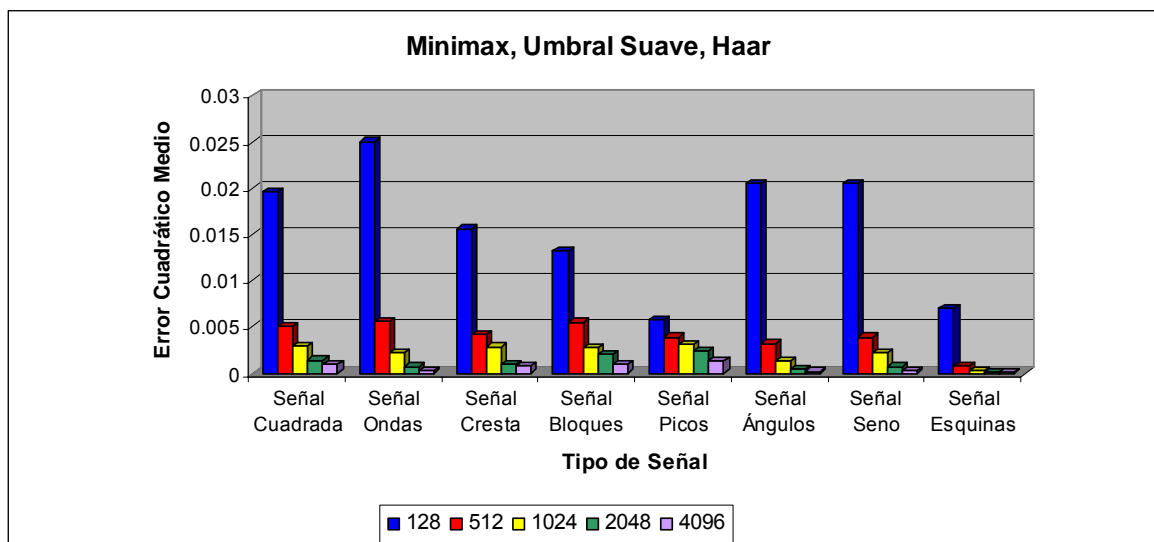


Fig. B.9: Gráfico comparativo de desempeño de las simulaciones con algoritmo Minimax, Umbral Suave, Haar

**Algoritmo: Minimax
Umbral Suave**

| Wavelet Daubechies 5 | | | | | |
|-----------------------------|----------|----------|----------|----------|----------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| Umbral (λ) | 2.2802 | 2.6469 | 2.8299 | 3.0130 | 3.1959 |
| | MSE | MSE | MSE | MSE | MSE |
| Señal Cuadrada | 0.01565 | 0.004349 | 0.003102 | 0.001204 | 0.001006 |
| Señal Ondas | 0.013156 | 0.004959 | 0.000601 | 0.000109 | 0.000097 |
| Señal Cresta | 0.011319 | 0.003313 | 0.001527 | 0.000512 | 0.000386 |
| Señal Bloques | 0.010563 | 0.004411 | 0.002608 | 0.001299 | 0.00062 |
| Señal Picos | 0.005736 | 0.003816 | 0.002664 | 0.002131 | 0.000917 |
| Señal Ángulos | 0.023731 | 0.001743 | 0.000231 | 0.00018 | 0.000131 |
| Señal Seno | 0.019663 | 0.004292 | 0.000262 | 0.000168 | 0.000159 |
| Señal Esquinas | 0.003587 | 0.000103 | 0.000061 | 0.000066 | 0.000051 |

Tabla B.10: Resultados de las simulaciones con algoritmo Minimax, Umbral Suave, Daubechies 5

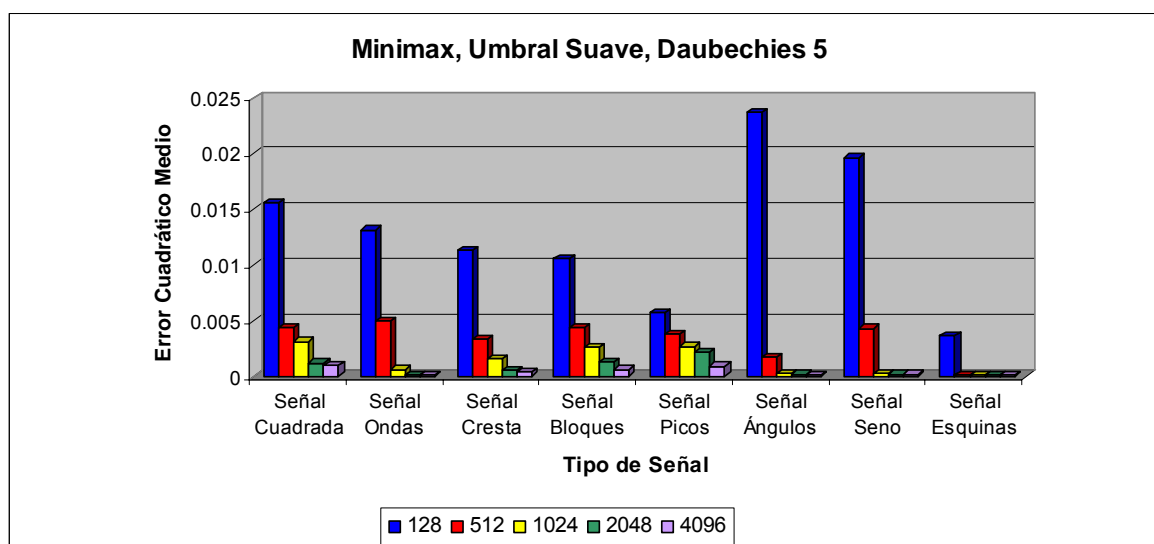


Fig. B.10: Gráfico comparativo de desempeño de las simulaciones con algoritmo Minimax, Umbral Suave, Daubechies 5

**Algoritmo: Minimax
Umbral Suave**

| Wavelet Coiflet 5 | | | | | |
|--------------------------|------------|------------|-------------|-------------|-------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| Umbral (λ) | 2.5711 | 2.9371 | 3.1200 | 3.3029 | 3.4859 |
| | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> |
| Señal Cuadrada | 0.01593 | 0.00437 | 0.002691 | 0.001309 | 0.000875 |
| Señal Ondas | 0.017749 | 0.005003 | 0.000238 | 0.000071 | 0.000101 |
| Señal Cresta | 0.013656 | 0.002757 | 0.001524 | 0.000648 | 0.000439 |
| Señal Bloques | 0.012425 | 0.00484 | 0.002297 | 0.001205 | 0.000603 |
| Señal Picos | 0.005689 | 0.003753 | 0.002826 | 0.002148 | 0.000887 |
| Señal Ángulos | 0.018018 | 0.001078 | 0.000202 | 0.000103 | 0.000128 |
| Señal Seno | 0.017501 | 0.002395 | 0.000224 | 0.000148 | 0.000174 |
| Señal Esquinas | 0.003481 | 0.000218 | 0.000073 | 0.000053 | 0.00006 |

Tabla B.11: Resultados de las simulaciones con algoritmo Minimax, Umbral Suave, Coiflet 5

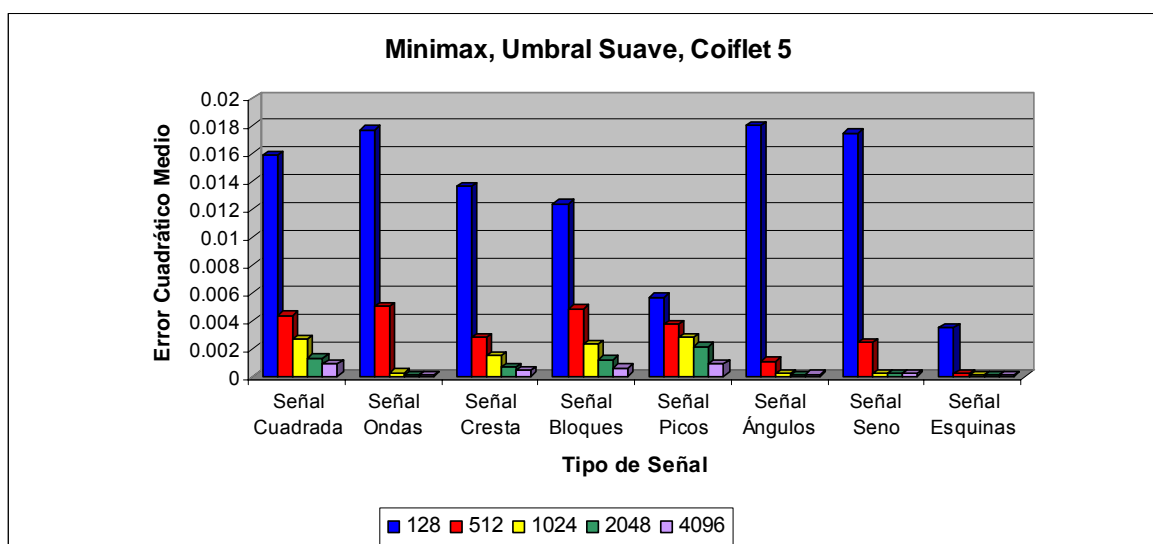


Fig. B.11: Gráfico comparativo de desempeño de las simulaciones con algoritmo Minimax, Umbral Suave, Coiflet 5

**Algoritmo: Minimax
Umbral Suave**

| Wavelet Symlet 5 | | | | | |
|-------------------------|------------|------------|-------------|-------------|-------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| Umbral (λ) | 2.2802 | 2.6469 | 2.8299 | 3.0130 | 3.1959 |
| | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> |
| Señal Cuadrada | 0.016157 | 0.004819 | 0.003048 | 0.001341 | 0.001026 |
| Señal Ondas | 0.008243 | 0.005029 | 0.000668 | 0.000104 | 0.000088 |
| Señal Cresta | 0.010055 | 0.003219 | 0.001371 | 0.000583 | 0.000351 |
| Señal Bloques | 0.009138 | 0.004355 | 0.003097 | 0.001223 | 0.000605 |
| Señal Picos | 0.005598 | 0.003829 | 0.002687 | 0.002124 | 0.000909 |
| Señal Ángulos | 0.025889 | 0.002014 | 0.000225 | 0.000121 | 0.000151 |
| Señal Seno | 0.024471 | 0.006583 | 0.000427 | 0.000165 | 0.000164 |
| Señal Esquinas | 0.003587 | 0.000122 | 0.000088 | 0.00006 | 0.000078 |

Tabla B.12: Resultados de las simulaciones con algoritmo Minimax, Umbral Suave, Symlet 5

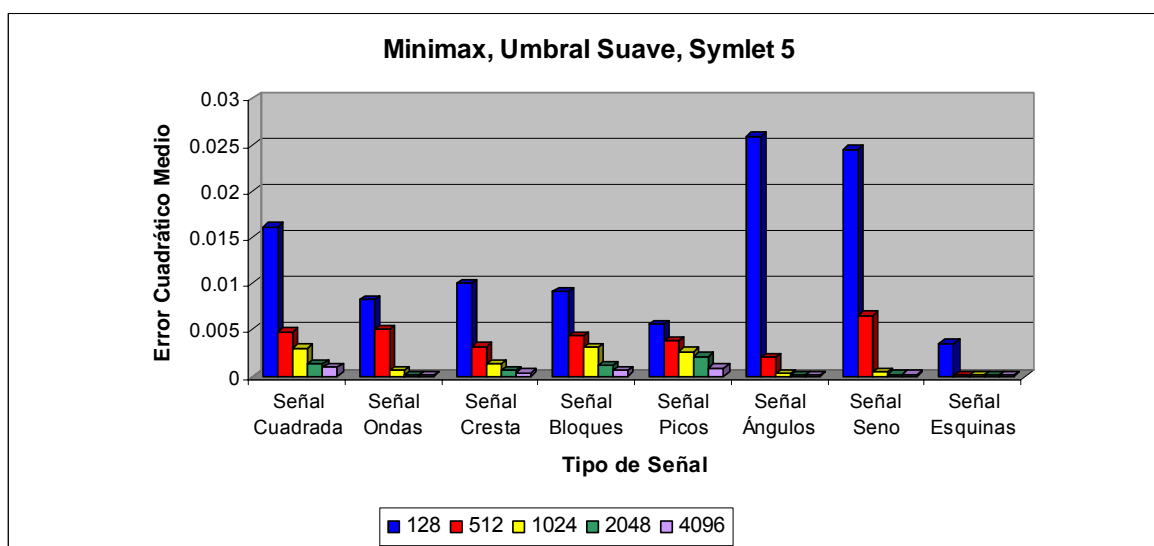


Fig. B.12: Gráfico comparativo de desempeño de las simulaciones con algoritmo Minimax, Umbral Suave, Symlet 5

**Algoritmo: Minimax
Umbral Duro**

| Wavelet Haar | | | | | |
|------------------------|----------|----------|----------|----------|----------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| Umbral (λ) | 1.8495 | 2.2203 | 2.4042 | 2.5877 | 2.7709 |
| | MSE | MSE | MSE | MSE | MSE |
| Señal Cuadrada | 0.019555 | 0.005191 | 0.002822 | 0.00147 | 0.000975 |
| Señal Ondas | 0.025037 | 0.005647 | 0.002281 | 0.000688 | 0.000241 |
| Señal Cresta | 0.015657 | 0.004117 | 0.002923 | 0.000996 | 0.000807 |
| Señal Bloques | 0.013149 | 0.005439 | 0.002699 | 0.002004 | 0.000946 |
| Señal Picos | 0.005815 | 0.003936 | 0.003081 | 0.00236 | 0.001391 |
| Señal Ángulos | 0.020649 | 0.003198 | 0.001305 | 0.000426 | 0.000212 |
| Señal Seno | 0.020633 | 0.004046 | 0.00218 | 0.000753 | 0.000288 |
| Señal Esquinas | 0.006936 | 0.000845 | 0.000254 | 0.000123 | 0.000087 |

Tabla B.13: Resultados de las simulaciones con algoritmo Minimax, Umbral Duro, Haar

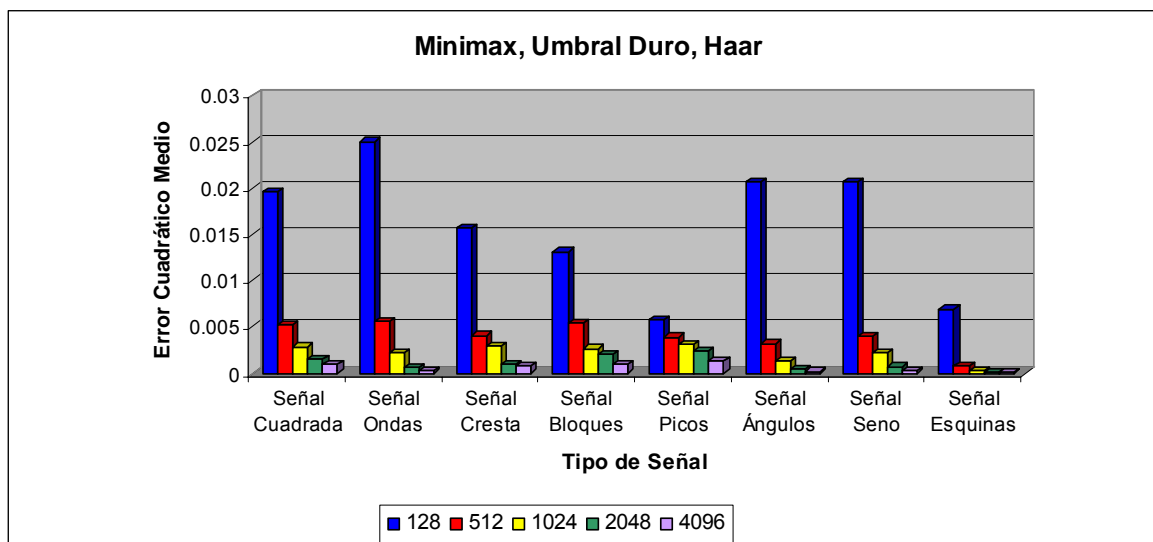


Fig. B.13: Gráfico comparativo de desempeño de las simulaciones con algoritmo Minimax, Umbral Duro, Haar

**Algoritmo: Minimax
Umbral Duro**

| Wavelet Daubechies 5 | | | | | |
|-----------------------------|------------|------------|-------------|-------------|-------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| Umbral (λ) | 2.2802 | 2.6469 | 2.8299 | 3.0130 | 3.1959 |
| | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> |
| Señal Cuadrada | 0.016622 | 0.004438 | 0.003126 | 0.001296 | 0.000996 |
| Señal Ondas | 0.014501 | 0.004905 | 0.000618 | 0.000092 | 0.000087 |
| Señal Cresta | 0.011709 | 0.003238 | 0.001553 | 0.0006 | 0.00036 |
| Señal Bloques | 0.010461 | 0.00439 | 0.002601 | 0.001282 | 0.000613 |
| Señal Picos | 0.005673 | 0.003815 | 0.002663 | 0.002123 | 0.00092 |
| Señal Ángulos | 0.023778 | 0.001788 | 0.000266 | 0.000123 | 0.000127 |
| Señal Seno | 0.019416 | 0.004316 | 0.00036 | 0.000134 | 0.000144 |
| Señal Esquinas | 0.003602 | 0.000135 | 0.000087 | 0.000063 | 0.000059 |

Tabla B.14: Resultados de las simulaciones con algoritmo Minimax, Umbral Duro, Daubechies 5

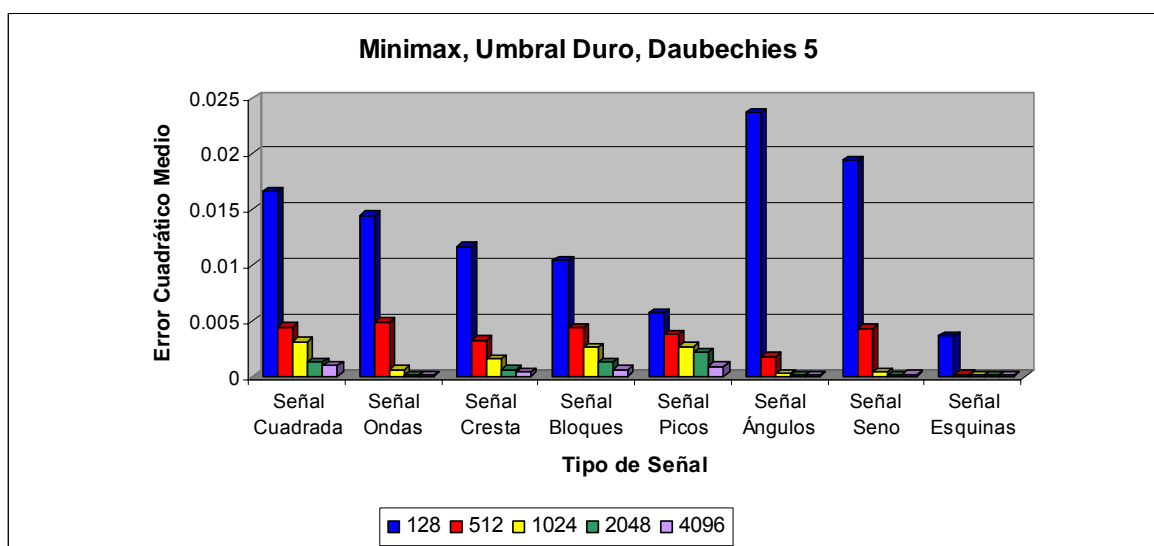


Fig. B.14: Gráfico comparativo de desempeño de las simulaciones con algoritmo Minimax, Umbral Duro, Daubechies 5

**Algoritmo: Minimax
Umbral Duro**

| Wavelet Coiflet 5 | | | | | |
|--------------------------|------------|------------|-------------|-------------|-------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| Umbral (λ) | 2.5711 | 2.9371 | 3.1200 | 3.3029 | 3.4859 |
| | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> |
| Señal Cuadrada | 0.016674 | 0.004452 | 0.002571 | 0.001419 | 0.000874 |
| Señal Ondas | 0.018306 | 0.004977 | 0.00025 | 0.000129 | 0.000096 |
| Señal Cresta | 0.013426 | 0.002719 | 0.00157 | 0.000613 | 0.000401 |
| Señal Bloques | 0.012404 | 0.00478 | 0.002294 | 0.001196 | 0.00061 |
| Señal Picos | 0.005687 | 0.003758 | 0.002822 | 0.002153 | 0.000882 |
| Señal Ángulos | 0.018296 | 0.001027 | 0.000185 | 0.000167 | 0.000111 |
| Señal Seno | 0.01775 | 0.002278 | 0.000222 | 0.000102 | 0.00017 |
| Señal Esquinas | 0.003522 | 0.000176 | 0.000073 | 0.000065 | 0.000057 |

Tabla B.15: Resultados de las simulaciones con algoritmo Minimax, Umbral Duro, Coiflet 5

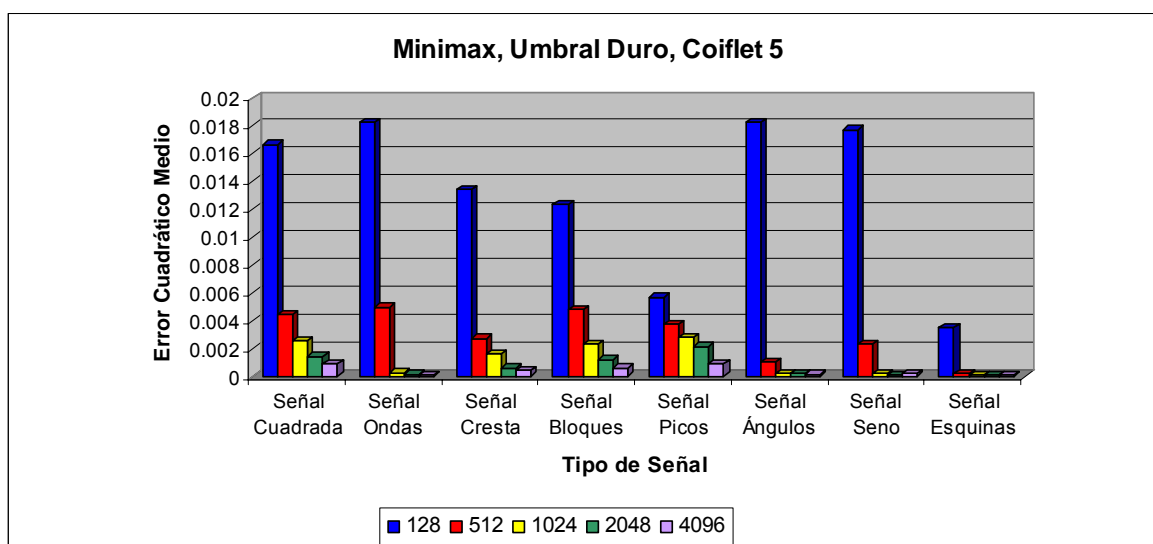


Fig. B.15: Gráfico comparativo de desempeño de las simulaciones con algoritmo Minimax, Umbral Duro, Coiflet 5

**Algoritmo: Minimax
Umbral Duro**

| Wavelet Symlet 5 | | | | | |
|-------------------------|------------|------------|-------------|-------------|-------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| Umbral (λ) | 2.2802 | 2.6469 | 2.8299 | 3.0130 | 3.1959 |
| | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> | <i>MSE</i> |
| Señal Cuadrada | 0.016161 | 0.004642 | 0.003005 | 0.00144 | 0.000955 |
| Señal Ondas | 0.008063 | 0.005046 | 0.00066 | 0.000118 | 0.000082 |
| Señal Cresta | 0.009798 | 0.003245 | 0.001345 | 0.000668 | 0.000349 |
| Señal Bloques | 0.009141 | 0.00439 | 0.003092 | 0.001237 | 0.000622 |
| Señal Picos | 0.005605 | 0.003847 | 0.0027 | 0.002119 | 0.000912 |
| Señal Ángulos | 0.024943 | 0.002143 | 0.000184 | 0.000145 | 0.000141 |
| Señal Seno | 0.024615 | 0.00654 | 0.00029 | 0.000113 | 0.000113 |
| Señal Esquinas | 0.003398 | 0.000153 | 0.000081 | 0.000072 | 0.000071 |

Tabla B.16: Resultados de las simulaciones con algoritmo Minimax, Umbral Duro, Symlet 5

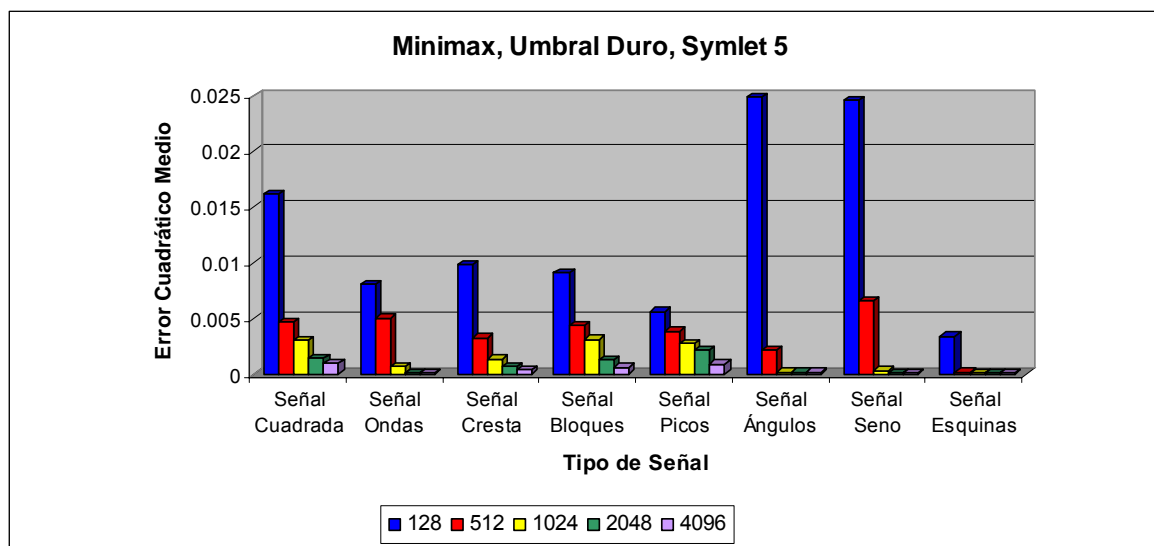


Fig. B.16: Gráfico comparativo de desempeño de las simulaciones con algoritmo Minimax, Umbral Duro, Symlet 5

**Algoritmo: Rigorous SURE
Umbral Suave**

| <i>Wavelet Haar</i> | | | | | |
|------------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| | MSE | MSE | MSE | MSE | MSE |
| Señal Cuadrada | 0.019512 | 0.004991 | 0.002883 | 0.001439 | 0.001025 |
| Señal Ondas | 0.025004 | 0.005646 | 0.002258 | 0.000678 | 0.000234 |
| Señal Cresta | 0.015665 | 0.004233 | 0.002884 | 0.001028 | 0.000800 |
| Señal Bloques | 0.013143 | 0.005466 | 0.002674 | 0.002028 | 0.000953 |
| Señal Picos | 0.005774 | 0.003934 | 0.003083 | 0.002360 | 0.001392 |
| Señal Ángulos | 0.020547 | 0.003187 | 0.001268 | 0.000469 | 0.000208 |
| Señal Seno | 0.020490 | 0.004006 | 0.002319 | 0.000759 | 0.000295 |
| Señal Esquinas | 0.006993 | 0.000844 | 0.000267 | 0.000132 | 0.000066 |
| | Umbral (λ) | Umbral (λ) | Umbral (λ) | Umbral (λ) | Umbral (λ) |
| Señal Cuadrada | 0.4928 | 0.5277 | 0.6574 | 0.3652 | 0.2859 |
| Señal Ondas | 0.7088 | 0.6829 | 0.9187 | 1.2048 | 0.1989 |
| Señal Cresta | 1.2720 | 0.9980 | 0.9948 | 0.5636 | 0.2068 |
| Señal Bloques | 0.6031 | 1.3184 | 1.3214 | 1.2765 | 0.1700 |
| Señal Picos | 0.3410 | 0.4781 | 0.6836 | 0.9077 | 0.1686 |
| Señal Ángulos | 1.2773 | 1.2796 | 0.9203 | 0.6600 | 0.1973 |
| Señal Seno | 1.1393 | 0.7192 | 1.2296 | 1.3543 | 0.2162 |
| Señal Esquinas | 1.0658 | 0.6147 | 0.8593 | 1.1639 | 0.1514 |

Tabla B.17: Resultados de las simulaciones con algoritmo Rigorous SURE, Umbral Suave, Haar

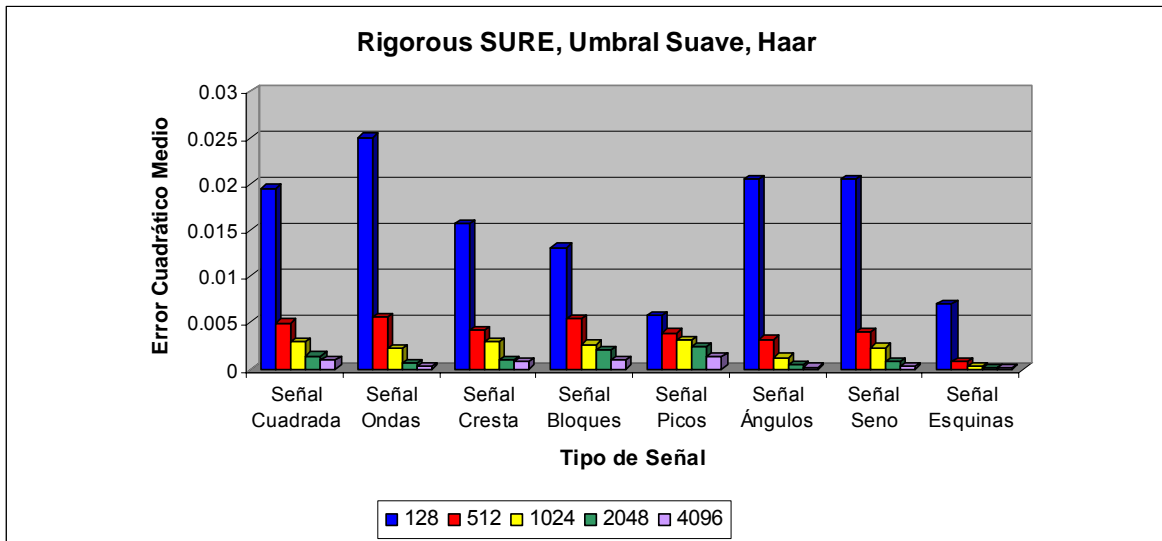


Fig. B.17: Gráfico comparativo de desempeño de las simulaciones con algoritmo Rigorous SURE, Umbral Suave, Haar

**Algoritmo: Rigorous SURE
Umbral Suave**

| Wavelet Daubechies 5 | | | | | |
|-----------------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| | MSE | MSE | MSE | MSE | MSE |
| Señal Cuadrada | 0.016092 | 0.004328 | 0.003103 | 0.001305 | 0.000993 |
| Señal Ondas | 0.013576 | 0.004943 | 0.000621 | 0.000089 | 0.000088 |
| Señal Cresta | 0.011831 | 0.003292 | 0.001564 | 0.000560 | 0.000370 |
| Señal Bloques | 0.010622 | 0.004395 | 0.002605 | 0.001278 | 0.000622 |
| Señal Picos | 0.005703 | 0.003814 | 0.002659 | 0.002123 | 0.000919 |
| Señal Ángulos | 0.023968 | 0.001745 | 0.000256 | 0.000177 | 0.000134 |
| Señal Seno | 0.018902 | 0.004342 | 0.000335 | 0.000172 | 0.000154 |
| Señal Esquinas | 0.003389 | 0.000128 | 0.000093 | 0.000058 | 0.000060 |
| | Umbral (λ) | Umbral (λ) | Umbral (λ) | Umbral (λ) | Umbral (λ) |
| Señal Cuadrada | 0.2091 | 0.1745 | 0.1693 | 0.1880 | 0.1790 |
| Señal Ondas | 0.1593 | 0.1488 | 0.1151 | 0.1273 | 0.1127 |
| Señal Cresta | 0.2591 | 0.1645 | 0.1556 | 0.1426 | 0.1429 |
| Señal Bloques | 0.2187 | 0.1408 | 0.1241 | 0.1104 | 0.1070 |
| Señal Picos | 0.2116 | 0.1252 | 0.1135 | 0.0714 | 0.0764 |
| Señal Ángulos | 0.2102 | 0.1389 | 0.1282 | 0.1459 | 0.1319 |
| Señal Seno | 0.1953 | 0.1476 | 0.1443 | 0.1423 | 0.1447 |
| Señal Esquinas | 0.1930 | 0.1182 | 0.1126 | 0.1123 | 0.1096 |

Tabla B.18: Resultados de las simulaciones con algoritmo Rigorous SURE, Umbral Suave, Daubechies 5

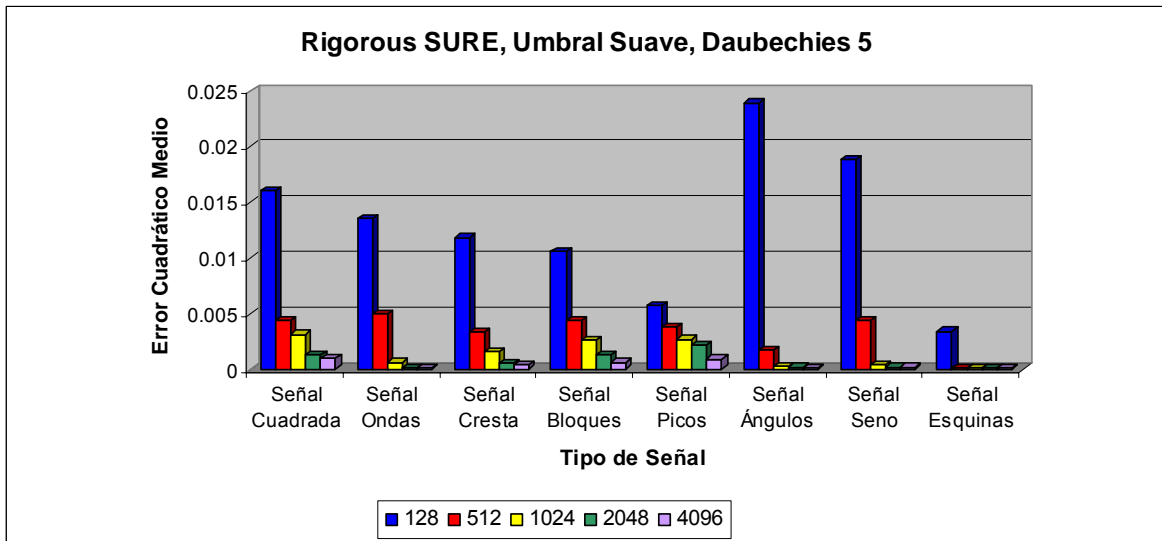


Fig. B.18: Gráfico comparativo de desempeño de las simulaciones con algoritmo Rigorous SURE, Umbral Suave, Daubechies 5

**Algoritmo: Rigorous SURE
Umbral Suave**

| Wavelet Coiflet 5 | | | | | |
|--------------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| | MSE | MSE | MSE | MSE | MSE |
| Señal Cuadrada | 0.016513 | 0.004378 | 0.002798 | 0.001359 | 0.000875 |
| Señal Ondas | 0.018365 | 0.004941 | 0.000233 | 0.000073 | 0.000081 |
| Señal Cresta | 0.013842 | 0.002674 | 0.001567 | 0.000603 | 0.000420 |
| Señal Bloques | 0.012320 | 0.004826 | 0.002301 | 0.001194 | 0.000609 |
| Señal Picos | 0.005686 | 0.003751 | 0.002823 | 0.002151 | 0.000890 |
| Señal Ángulos | 0.018318 | 0.001032 | 0.000144 | 0.000166 | 0.000113 |
| Señal Seno | 0.018094 | 0.002301 | 0.000153 | 0.000110 | 0.000165 |
| Señal Esquinas | 0.003485 | 0.000152 | 0.000056 | 0.000086 | 0.000061 |
| | Umbral (λ) | Umbral (λ) | Umbral (λ) | Umbral (λ) | Umbral (λ) |
| Señal Cuadrada | 0.2061 | 0.1667 | 0.1389 | 0.1268 | 0.1254 |
| Señal Ondas | 0.2207 | 0.1146 | 0.1132 | 0.1020 | 0.0997 |
| Señal Cresta | 0.1876 | 0.1384 | 0.1131 | 0.1112 | 0.1056 |
| Señal Bloques | 0.2004 | 0.1210 | 0.1021 | 0.0882 | 0.0854 |
| Señal Picos | 0.1937 | 0.1035 | 0.0770 | 0.0728 | 0.0661 |
| Señal Ángulos | 0.1942 | 0.1284 | 0.1114 | 0.1137 | 0.1090 |
| Señal Seno | 0.2056 | 0.1318 | 0.1146 | 0.1208 | 0.1100 |
| Señal Esquinas | 0.1927 | 0.1164 | 0.0975 | 0.0941 | 0.0974 |

Tabla B.19: Resultados de las simulaciones con algoritmo Rigorous SURE, Umbral Suave, Coiflet 5

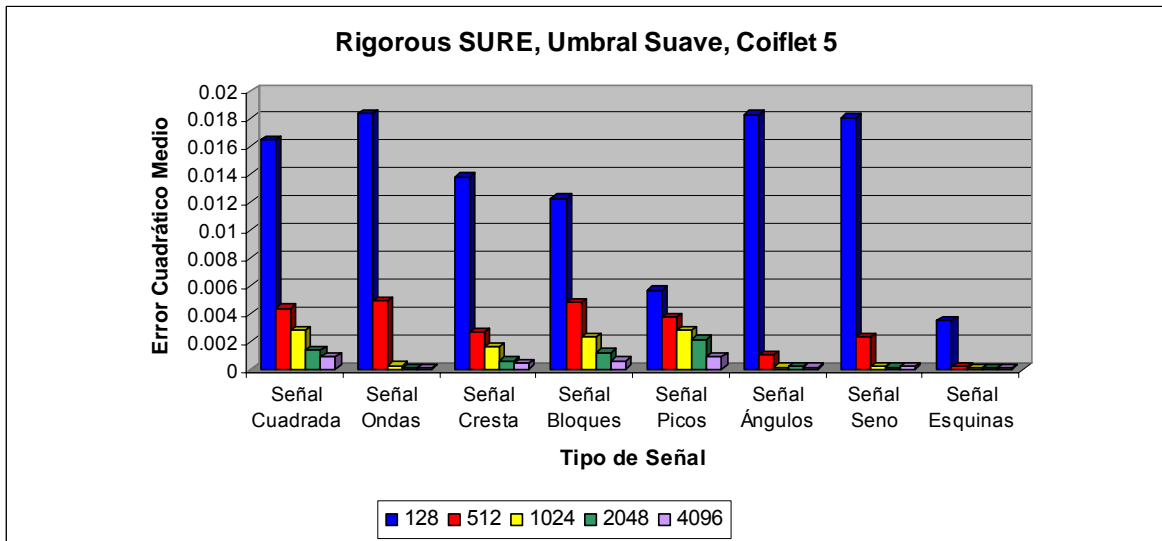


Fig. B.19: Gráfico comparativo de desempeño de las simulaciones con algoritmo Rigorous SURE, Umbral Suave, Coiflet 5

**Algoritmo: Rigorous SURE
Umbral Suave**

| Wavelet Symlet 5 | | | | | |
|-------------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| | MSE | MSE | MSE | MSE | MSE |
| Señal Cuadrada | 0.016197 | 0.004725 | 0.002944 | 0.001445 | 0.001059 |
| Señal Ondas | 0.008112 | 0.005025 | 0.000631 | 0.000076 | 0.000095 |
| Señal Cresta | 0.010159 | 0.003276 | 0.001349 | 0.000563 | 0.000362 |
| Señal Bloques | 0.009207 | 0.004381 | 0.003082 | 0.001251 | 0.000622 |
| Señal Picos | 0.005614 | 0.003830 | 0.002694 | 0.002124 | 0.000914 |
| Señal Ángulos | 0.026252 | 0.002123 | 0.000228 | 0.000113 | 0.000116 |
| Señal Seno | 0.024267 | 0.006595 | 0.000332 | 0.000119 | 0.000150 |
| Señal Esquinas | 0.003457 | 0.000094 | 0.000088 | 0.000061 | 0.000056 |
| | Umbral (λ) | Umbral (λ) | Umbral (λ) | Umbral (λ) | Umbral (λ) |
| Señal Cuadrada | 0.1968 | 0.1965 | 0.1763 | 0.1771 | 0.1737 |
| Señal Ondas | 0.2295 | 0.1295 | 0.1245 | 0.1239 | 0.1187 |
| Señal Cresta | 0.2129 | 0.1464 | 0.1359 | 0.1401 | 0.1499 |
| Señal Bloques | 0.2384 | 0.1414 | 0.1288 | 0.1079 | 0.1040 |
| Señal Picos | 0.1810 | 0.1255 | 0.0953 | 0.0819 | 0.0731 |
| Señal Ángulos | 0.1872 | 0.1523 | 0.1441 | 0.1437 | 0.1381 |
| Señal Seno | 0.2225 | 0.1378 | 0.1354 | 0.1412 | 0.1418 |
| Señal Esquinas | 0.2082 | 0.1317 | 0.1062 | 0.1132 | 0.1155 |

Tabla B.20: Resultados de las simulaciones con algoritmo Rigorous SURE, Umbral Suave, Symlet 5

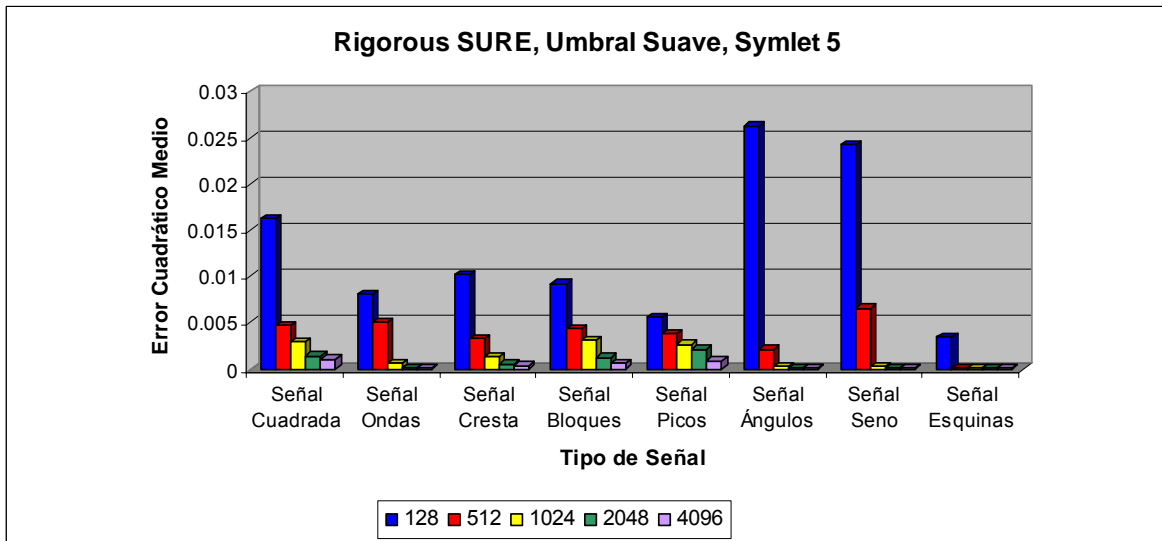


Fig. B.20: Gráfico comparativo de desempeño de las simulaciones con algoritmo Rigorous SURE, Umbral Suave, Symlet 5

**Algoritmo: Rigorous SURE
Umbral Duro**

| <i>Wavelet Haar</i> | | | | | |
|------------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| | MSE | MSE | MSE | MSE | MSE |
| Señal Cuadrada | 0.019722 | 0.005085 | 0.002855 | 0.001489 | 0.000943 |
| Señal Ondas | 0.024978 | 0.005688 | 0.002266 | 0.000726 | 0.000240 |
| Señal Cresta | 0.015616 | 0.004193 | 0.002863 | 0.001006 | 0.000794 |
| Señal Bloques | 0.013142 | 0.005521 | 0.002678 | 0.002011 | 0.000945 |
| Señal Picos | 0.005798 | 0.003941 | 0.003081 | 0.002362 | 0.001390 |
| Señal Ángulos | 0.020523 | 0.003208 | 0.001316 | 0.000449 | 0.000232 |
| Señal Seno | 0.020497 | 0.003957 | 0.002271 | 0.000723 | 0.000311 |
| Señal Esquinas | 0.006949 | 0.000824 | 0.000277 | 0.000114 | 0.000080 |
| | Umbral (λ) | Umbral (λ) | Umbral (λ) | Umbral (λ) | Umbral (λ) |
| Señal Cuadrada | 1.0160 | 0.5718 | 0.4698 | 0.3494 | 0.2740 |
| Señal Ondas | 0.7893 | 0.6172 | 0.9260 | 1.1869 | 0.1711 |
| Señal Cresta | 1.2678 | 1.1018 | 1.3844 | 0.8860 | 0.2011 |
| Señal Bloques | 0.6270 | 0.9224 | 1.3374 | 1.2759 | 0.1685 |
| Señal Picos | 0.3556 | 0.4455 | 0.6357 | 0.9451 | 0.1579 |
| Señal Ángulos | 1.2693 | 1.3047 | 0.7861 | 0.5829 | 0.2010 |
| Señal Seno | 1.2202 | 0.8221 | 1.1779 | 1.0983 | 0.2167 |
| Señal Esquinas | 1.1143 | 0.5280 | 0.8353 | 1.1892 | 0.1438 |

Tabla B.21: Resultados de las simulaciones con algoritmo Rigorous SURE, Umbral Duro, Haar

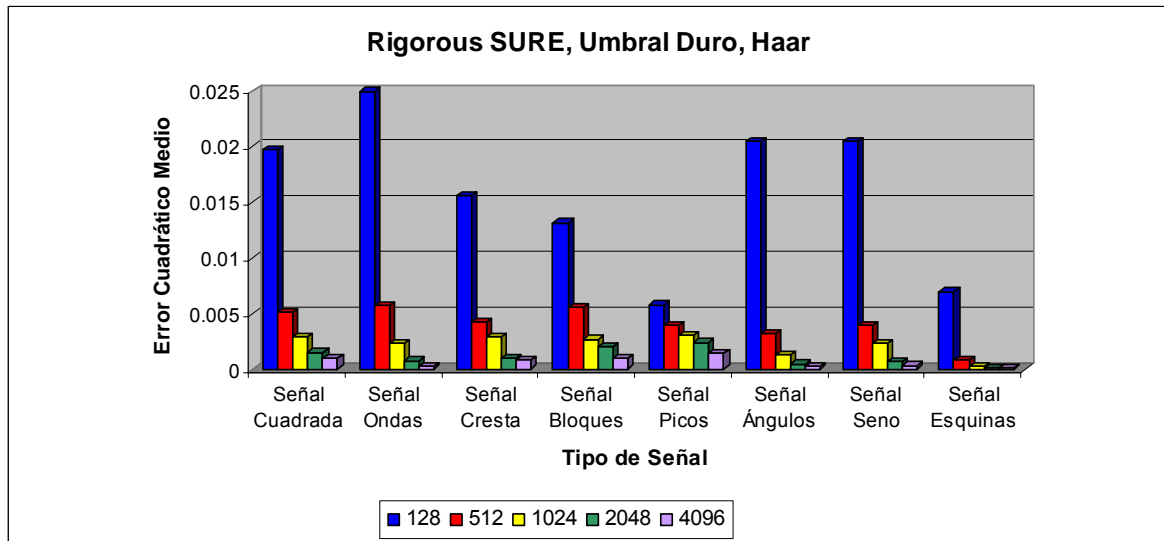


Fig. B.21: Gráfico comparativo de desempeño de las simulaciones con algoritmo Rigorous SURE, Umbral Duro, Haar

**Algoritmo: Rigorous SURE
Umbral Duro**

| Wavelet Daubechies 5 | | | | | |
|-----------------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| | MSE | MSE | MSE | MSE | MSE |
| Señal Cuadrada | 0.016158 | 0.004320 | 0.003191 | 0.001405 | 0.001049 |
| Señal Ondas | 0.013510 | 0.005043 | 0.000622 | 0.000095 | 0.000081 |
| Señal Cresta | 0.012212 | 0.003363 | 0.001549 | 0.000550 | 0.000362 |
| Señal Bloques | 0.010524 | 0.004378 | 0.002558 | 0.001308 | 0.000625 |
| Señal Picos | 0.005736 | 0.003825 | 0.002668 | 0.002126 | 0.000920 |
| Señal Ángulos | 0.023214 | 0.001786 | 0.000205 | 0.000162 | 0.000135 |
| Señal Seno | 0.019112 | 0.004334 | 0.000342 | 0.000128 | 0.000186 |
| Señal Esquinas | 0.003456 | 0.000090 | 0.000068 | 0.000070 | 0.000054 |
| | Umbral (λ) | Umbral (λ) | Umbral (λ) | Umbral (λ) | Umbral (λ) |
| Señal Cuadrada | 0.2375 | 0.1723 | 0.1778 | 0.1861 | 0.1781 |
| Señal Ondas | 0.2463 | 0.1435 | 0.1258 | 0.1315 | 0.1168 |
| Señal Cresta | 0.1801 | 0.1541 | 0.1468 | 0.1380 | 0.1360 |
| Señal Bloques | 0.2336 | 0.1543 | 0.1159 | 0.1155 | 0.1024 |
| Señal Picos | 0.2202 | 0.1446 | 0.1013 | 0.0852 | 0.0751 |
| Señal Ángulos | 0.1733 | 0.1556 | 0.1561 | 0.1374 | 0.1400 |
| Señal Seno | 0.2207 | 0.1463 | 0.1635 | 0.1499 | 0.1501 |
| Señal Esquinas | 0.1855 | 0.1122 | 0.1166 | 0.1147 | 0.1080 |

Tabla B.22: Resultados de las simulaciones con algoritmo Rigorous SURE, Umbral Duro, Daubechies 5

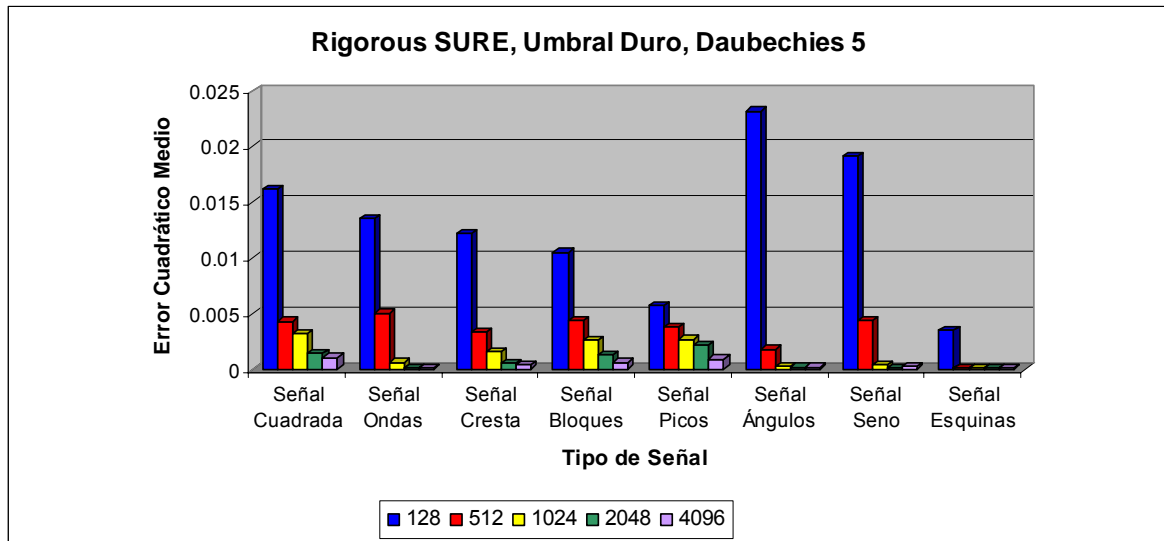


Fig. B.22: Gráfico comparativo de desempeño de las simulaciones con algoritmo Rigorous SURE, Umbral Duro, Daubechies 5

**Algoritmo: Rigorous SURE
Umbral Duro**

| Wavelet Coiflet 5 | | | | | |
|--------------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| | MSE | MSE | MSE | MSE | MSE |
| Señal Cuadrada | 0.016155 | 0.004422 | 0.002648 | 0.001326 | 0.000887 |
| Señal Ondas | 0.017830 | 0.005040 | 0.000241 | 0.000075 | 0.000092 |
| Señal Cresta | 0.013908 | 0.002633 | 0.001560 | 0.000615 | 0.000447 |
| Señal Bloques | 0.012363 | 0.004770 | 0.002281 | 0.001181 | 0.000608 |
| Señal Picos | 0.005718 | 0.003750 | 0.002823 | 0.002150 | 0.000883 |
| Señal Ángulos | 0.018266 | 0.000983 | 0.000173 | 0.000096 | 0.000152 |
| Señal Seno | 0.017671 | 0.002266 | 0.000166 | 0.000162 | 0.000151 |
| Señal Esquinas | 0.003655 | 0.000148 | 0.000078 | 0.000075 | 0.000058 |
| | Umbral (λ) | Umbral (λ) | Umbral (λ) | Umbral (λ) | Umbral (λ) |
| Señal Cuadrada | 0.2290 | 0.1541 | 0.1354 | 0.1304 | 0.1222 |
| Señal Ondas | 0.2160 | 0.1252 | 0.1189 | 0.1136 | 0.0966 |
| Señal Cresta | 0.2071 | 0.1397 | 0.1272 | 0.1122 | 0.1107 |
| Señal Bloques | 0.2008 | 0.1239 | 0.0948 | 0.0861 | 0.0891 |
| Señal Picos | 0.1877 | 0.1176 | 0.0791 | 0.0703 | 0.0683 |
| Señal Ángulos | 0.2133 | 0.1281 | 0.1109 | 0.1058 | 0.1077 |
| Señal Seno | 0.2246 | 0.1318 | 0.1194 | 0.1198 | 0.1083 |
| Señal Esquinas | 0.2124 | 0.1120 | 0.1028 | 0.1020 | 0.0950 |

Tabla B.23: Resultados de las simulaciones con algoritmo Rigorous SURE, Umbral Duro, Coiflet 5

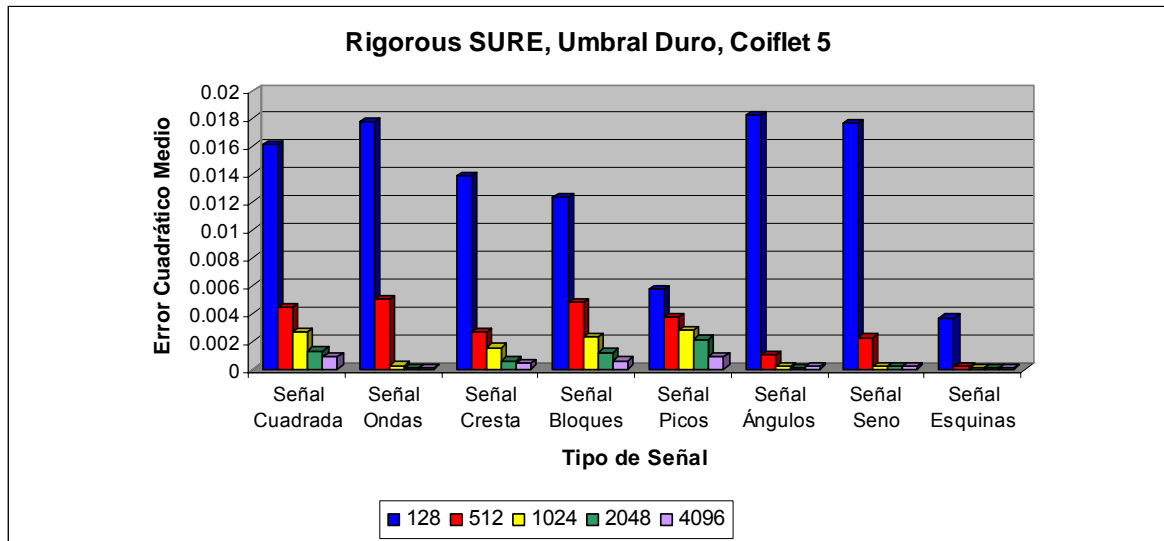


Fig. B.23: Gráfico comparativo de desempeño de las simulaciones con algoritmo Rigorous SURE, Umbral Duro, Coiflet 5

**Algoritmo: Rigorous SURE
Umbral Duro**

| Wavelet Symlet 5 | | | | | |
|-------------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| Número de Muestras (n) | 128 | 512 | 1024 | 2048 | 4096 |
| | MSE | MSE | MSE | MSE | MSE |
| Señal Cuadrada | 0.016816 | 0.004737 | 0.003135 | 0.001348 | 0.001017 |
| Señal Ondas | 0.007858 | 0.005022 | 0.000621 | 0.000094 | 0.000089 |
| Señal Cresta | 0.010431 | 0.003228 | 0.001337 | 0.000716 | 0.000324 |
| Señal Bloques | 0.009174 | 0.004395 | 0.003123 | 0.001236 | 0.000618 |
| Señal Picos | 0.005616 | 0.003832 | 0.002688 | 0.002127 | 0.000908 |
| Señal Ángulos | 0.026384 | 0.002052 | 0.000215 | 0.000169 | 0.000122 |
| Señal Seno | 0.025145 | 0.006636 | 0.000377 | 0.000178 | 0.000145 |
| Señal Esquinas | 0.003505 | 0.000111 | 0.000084 | 0.000061 | 0.000069 |
| | Umbral (λ) | Umbral (λ) | Umbral (λ) | Umbral (λ) | Umbral (λ) |
| Señal Cuadrada | 0.2017 | 0.1673 | 0.1725 | 0.1862 | 0.1775 |
| Señal Ondas | 0.2772 | 0.1317 | 0.1203 | 0.1253 | 0.1204 |
| Señal Cresta | 0.2199 | 0.1372 | 0.1339 | 0.1420 | 0.1321 |
| Señal Bloques | 0.2819 | 0.1136 | 0.1078 | 0.1072 | 0.1054 |
| Señal Picos | 0.2045 | 0.1484 | 0.1131 | 0.0735 | 0.0751 |
| Señal Ángulos | 0.1901 | 0.1405 | 0.1365 | 0.1284 | 0.1349 |
| Señal Seno | 0.2370 | 0.1490 | 0.1454 | 0.1407 | 0.1398 |
| Señal Esquinas | 0.2316 | 0.1258 | 0.1123 | 0.1095 | 0.1090 |

Tabla B.24: Resultados de las simulaciones con algoritmo Rigorous SURE, Umbral Duro, Symlet 5

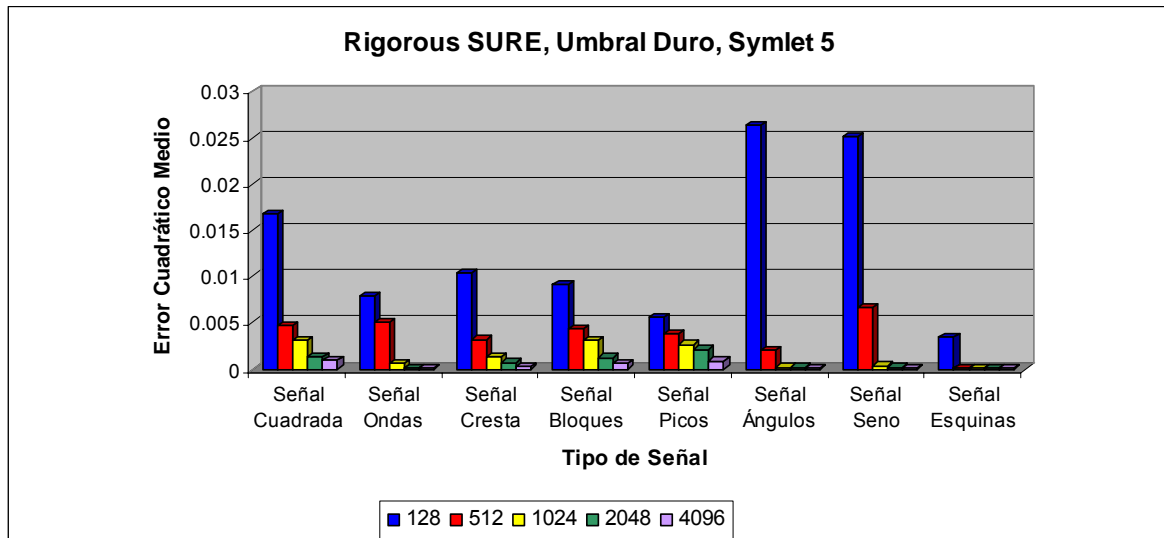


Fig. B.24: Gráfico comparativo de desempeño de las simulaciones con algoritmo Rigorous SURE, Umbral Duro, Symlet 5

Referencias Bibliográficas

- [1]. Adhemar Bultheel, *Wavelets with applications in signal and image processing*, Septiembre de 2003.
- [2]. Albert Boggess y Francis J. Narcowich, *A First Course in Wavelets with Fourier Analysis*, Prentice Hall, Universidad de Texas.
- [3]. Amara Graps, *An Introduction to Wavelets*, IEEE Computational Science and Engineering, 1995.
- [4]. Anestis Antoniadis, Jeremie Bigot y Theofanis Sapatinas, *Wavelet Stimators in Nonparametric Regression: A comparative Simulation Study*, Journal of Statistical Software 6, 2001.
- [5]. David Cuesta Frau, *Estudio de métodos para procesamiento y agrupación de señales electrocardiográficas*, Tesis Doctoral, Universidad Politécnica de Valencia, Septiembre de 2001.
- [6]. Eduardo Pedro Serrano, *Introducción a la Transformada Wavelet y sus Aplicaciones al Procesamiento de Señales de Emisión Acústica*, Escuela de Ciencia y Tecnología, Universidad Nacional de General San Martín
- [7]. Ferrel G. Stremmler, *Sistemas de Comunicación*, Fondo Educativo Interamericano, México, 1989.
- [8]. G. Kaiser, *A Really Friendly Guide to Wavelets*, Boston: Birkhäuser, 1994.
- [9]. H. Andrea Hajar C., Juan Pablo Newman A., María Ángeles Rodríguez B. y Rafael Peñaloza, *Wavelets*.

Referencias Bibliográficas

- [10]. Hamid Krim, Dewey Tucker, Stéphane Mallat y David Donoho, *On Denoising and Best Signal Representation*, IEEE Transactions on Information Theory, Vol. 45, No. 7, Noviembre de 1999.
- [11]. Hamid Krim y Irvin C. Schick, *Minimax Description Length for Signal Denoising and Optimized Representation*, IEEE Transactions on Information Theory, Vol. 45, No.3, Abril de 1999.
- [12]. Ingrid Daubechies, *Ten lectures on Wavelets*, Philadelphia: SIAM, 1992.
- [13]. Juan Antonio Pérez Ortiz, *Codificación Fractal de imágenes*, Julio de 1998.
- [14]. Junhui Qian, *Denoising by Wavelet Transform*, Rice University, Department of Electrical Engineering, 2000.
- [15]. M.C.E. Rosas-Orea, M. Hernández-Díaz, V. Alarcón-Aquino y L.G. Guerrero-Ojeda, *A Comparative Simulation Study of Wavelet Based Denoising Algorithms*, Departamento de Ingeniería Electrónica y Mecatrónica de la Universidad de las Américas, Puebla, CONIELECOMP 2005.
- [16]. Miguel Martinez Lozano, *Análisis y Medidas de Procesos No Estacionarios en el dominio Tiempo-Frecuencia*, Universidad Politécnica de Madrid, España, 2003-2004.
- [17]. Pablo Faundez y Álvaro Fuentes, *Procesamiento Digital de Señales Acústicas utilizando Wavelets*, Instituto de Matemáticas UACH.
- [18]. Radu Ranta, Valerie Louis-Dorr, Christian Heinrich y Didier Wolf, *Iterative Wavelet-Based Denoising Methods and Robust Outlier Detection*, IEEE Signal Processing Letters, Vol. 12, No.8, Agosto de 2005.

Referencias Bibliográficas

- [19]. Tapan K. Sarkar y C. Su, *A Tutorial on Wavelets form an Electrical Engineering Perspective, Part 2: The Continuos Case*, IEEE Antennas and Propagation Magazine, Vol. 40, No. 6, Diciembre de 1998.

- [20]. The Mathworks, Inc., *Wavelet Toolbox. Matlab: The Language of Technical Computing*, Versión 7.0.0.19920 Release 14, Mayo 6 de 2004.

- [21]. Wayne Tomasi, *Electronic Communications systems: fundamentals through advanced*, Prentice Hall 4ª Ed., New Jersey, USA, 2001

- [22]. Feng Qiao y Rachel Milam, *Daubechies Wavelet Filter Design*, Connexions module, Version 2.1, Junio de 2005

- [23]. Dong Wei, *Coiflet-Type Wavelets: Theory, Design and Applications*, Universidad d Texas, Austin, USA, 1998