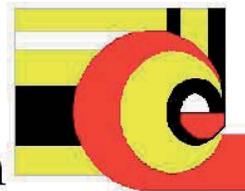




Instituto Politécnico Nacional
Centro de Investigación en Computación



Aplicación de la transformada de ondeleta para la estimación de canales PLC

T e s i s

que para obtener el grado de
Doctor en Ciencias de la Computación

presenta

M.I. Gerardo Abel Laguna Sánchez

Director de Tesis:

Dr. Ricardo Barrón Fernández

Jurado Calificador:

Presidente: Dr. Sergio Suárez Guerra

Secretario: Dr. Oleksiy Progrebinyak

Primer Vocal: Dr. Ricardo Barrón Fernández

Segundo Vocal: Dr. Alfonso Prieto Guerrero

Tercer Vocal: Dr. Luis Alfonso Villa Vargas

Suplente: Dr. José Luis Oropeza Rodríguez

Agradecimientos

Gracias

A mis abuelos, por su tesón y porque soy beneficiario de sus logros así como heredero de su estirpe de sangre antigua.

Gracias

A mis padres, por forjar mi carne y mi espíritu. Porque, según la ley del maíz, soy grano de su milpa. A mis hermanos, por ser la calabacita, el frijolito y el chilito con los que he crecido en el surco.

Gracias

A mi esposa Gaby, por su incondicional apoyo solidario y por complementarme refrendando el supremo principio dual.

Gracias

A mi maestro y director de tesis, Dr. Ricardo Barrón Fernández, por señalarme el camino, alentarme para transitarlo y empujarme más allá de los límites preconcebidos.

Gracias

A todos mis maestros, al Instituto Politécnico Nacional, a la Universidad Autónoma Metropolitana y a la Universidad Nacional Autónoma de México, por su noble valimiento y por ser germen de humanismo y hombres sabios.

Gracias

Al Consejo Nacional de Ciencia y Tecnología, CONACyT, por el apoyo económico proporcionado durante la realización del doctorado.

Gracias

Al espíritu...

Topohual:

Inic ce, ye in yelli.

Inic ome, ye in yotl.

Inic nahui, ye nemiliztli.

Inic chicome, tetzauh patlani.

Inic matlacyei, ye in xochitl.

Nuestra cuenta:

Es uno, así es su ser.

Es dos, así es su esencia.

Es cuatro, así es la vida.

Es siete, el imponente vuelo.

Es trece, así es su flor.

**Con especial dedicatoria
para mi hijo *Santiago Tonatiuh***

Índice general

Índice de Figuras	VI
Índice de Cuadros	IX
I Estimación del canal PLC y ondeletas	1
1. Introducción	3
1.1. Mercado potencial de la tecnología PLC	5
1.2. Maximización en el aprovechamiento del canal PLC	7
1.3. Motivación	8
1.4. Planteamiento del problema	9
1.5. Propuesta de la tesis	9
1.5.1. Alcance	10
1.6. Justificación	11
1.7. Hipótesis	12
1.8. Objetivo general	13
1.9. Objetivos particulares	13
1.10. Aportaciones	13
1.11. Plan de tesis	14
2. Modelo para el canal PLC y simulación del transreceptor OFDM	15
2.1. Los modelos existentes para un canal PLC	15
2.2. Modelo de múltiples trayectorias	17
2.3. Modelo analítico con matrices de transmisión	19
2.3.1. La matriz de transmisión y la función de transferencia	21
2.3.2. Función de transferencia de una red con topología de nodo en T	22
2.4. Modelo de ruido para el canal PLC	24
2.5. Canal ciclo-estacionario	25
2.6. Simulación del canal PLC	29
2.7. Simulación de transreceptor OFDM	31
2.7.1. Principio de operación de la técnica OFDM	31
2.7.2. La señal transmitida con OFDM	33

2.7.3.	Intervalo de guardia	34
2.7.4.	Especificación para el transreceptor OFDM simulado	35
2.7.5.	Metodología de simulación	38
2.8.	Conclusiones del capítulo	38
3.	Estimación de canal en sistemas OFDM	41
3.1.	Breve reseña sobre OFDM y DMT	41
3.2.	Estimación del canal	41
3.2.1.	Procesos básicos de cualquier receptor OFDM	42
3.2.2.	La señal OFDM recibida en un canal multi-trayectoria con sincronización perfecta	42
3.2.3.	Igualamiento y estimación del canal	44
3.2.4.	Tipos de estimadores de canal	45
3.3.	Evolución de la estimación del canal para sistemas OFDM	46
3.3.1.	Técnicas de estimación del canal únicamente con base en pilotos	47
3.3.2.	Técnicas regidas por el proceso de decisión	50
3.3.3.	Técnicas semi-ciegas de estimación de canal	51
3.3.4.	Técnicas de estimación del canal totalmente ciegas	51
3.4.	Ejemplos de estimadores de canal asistidos por símbolos pilotos	52
3.4.1.	Estimación del canal en el sentido de mínimos cuadrados (LS)	52
3.4.2.	Estimación del canal por el criterio del mínimo error cuadrático medio (MMSE)	53
3.4.3.	Estimación por filtrado en el dominio de transformación	54
3.5.	Estimación del espectro en potencia	56
3.6.	Estimación del canal PLC	57
3.7.	Conclusiones del capítulo	58
4.	Conceptos básicos sobre la transformada de ondeleta y su aplicación a la estimación	59
4.1.	Evolución de las ondeletas	59
4.1.1.	Transformada de ondeleta continua	62
4.1.2.	Análisis multi-resolución	62
4.1.3.	Noción de marco	63
4.1.4.	Generalización a M-bandas	64
4.1.5.	Ondeletas biortogonales	65
4.1.6.	Descomposición de paquete de ondeletas	67
4.2.	Relaciones elementales	68
4.3.	Transformada de ondeleta discreta	71
4.3.1.	El algoritmo en cascada	71
4.3.2.	Notación de operador de DWT	72
4.4.	Ondeletas y la estimación estadística	75
4.4.1.	Compactación de la energía	75
4.4.2.	Buena localización	75
4.4.3.	Manejo del principio de Heisenberg	76

4.4.4.	Propiedad de emblanquecimiento	76
4.4.5.	Control de suavidad	76
4.5.	Eficiencia de la transformada de ondeleta discreta	76
4.6.	Construcción de las ondeletas	78
4.7.	Momentos de desvanecimiento	79
4.8.	Familias de ondeletas básicas	79
4.8.1.	Ondeletas de Haar	79
4.8.2.	Ondeletas con soporte compacto de Daubechies	80
4.9.	Conclusiones del capítulo	81
5.	Estimación de canales PLC soportada por descomposición de ondeleta	85
5.1.	Antecedentes	85
5.2.	Orígenes y evolución de la estimación empleando ondeletas	86
5.3.	Trabajos relacionados con la estimación del canal empleando ondeletas	87
5.4.	Estimación del canal PLC con el enfoque de eliminación del ruido en el dominio de las ondeletas	87
5.4.1.	Descripción de estimación del canal con el criterio de Mínimos Cuadrados	88
5.4.2.	Umbralado en el dominio de las ondeletas	88
5.4.3.	Metodología propuesta para la estimación del canal soportada por ondeletas	92
5.4.4.	Simulaciones por computadora	96
5.4.5.	Sistema OFDM de referencia	97
5.4.6.	Resultados numéricos	97
5.4.7.	Discusión de resultados	97
5.5.	Conclusiones del capítulo	105
II	Estimación del canal PLC, heurísticas bio-inspiradas y GPU	107
6.	Estimación semi-ciega del canal PLC mediante un algoritmo inspirado en PSO	109
6.1.	Introducción	109
6.2.	Método de estimación semi-ciega propuesto	110
6.3.	Resultados numéricos	114
6.3.1.	Sistema OFDM de referencia	115
6.4.	Discusión de los resultados	115
6.5.	Conclusiones del capítulo	118
7.	Programación paralela del algoritmo PSO en un dispositivo de procesamiento de gráficos con múltiples hilos	119
7.1.	Antecedentes	120
7.2.	Trabajo relacionado	121
7.3.	Introducción a los dispositivos GPU y a la arquitectura de multi-hilos	121
7.3.1.	Arquitectura del GPU	122
7.3.2.	Arquitectura CUDA	123

7.4.	El algoritmo de optimización por cúmulo de partículas	125
7.5.	Puesta en práctica de las variantes paralelas de PSO en el GPU	125
7.6.	Experimentos y resultados	131
7.6.1.	Procedimiento experimental	132
7.6.2.	Métricas de desempeño para procesamiento paralelo	134
7.6.3.	Resultados experimentales	135
7.7.	Conclusiones del capítulo	142
8.	Conclusiones generales y trabajo futuro	143
8.0.1.	Trabajo futuro	144
III	Apéndices	146
A.	Notación empleada	147
B.	Glosario	149
C.	Efecto del número de niveles de descomposición en la estimación soportada por ondeletas	153
D.	Comparación de la estimación soportada por ondeletas contra técnicas clásicas de estimación lineal	157
E.	Código Matlab básico empleado en las simulaciones numéricas para la estimación del canal PLC	161
E.1.	Estructura del programa	161
E.2.	Instalación y ejecución	162
E.3.	Listado del código esencial	165
E.3.1.	Cuerpo del programa de simulación Monte Carlo	165
E.3.2.	Código para la construcción de tramas OFDM	170
E.3.3.	Código para la simulación del canal PLC	182
E.3.4.	Código para la simulación del ruido impulsivo	186
E.3.5.	Código para la simulación del transreceptor con modulación OFDM	187
E.3.6.	Código para la estimación del canal	193
E.3.7.	Código para técnicas clásicas de estimación lineal	197
F.	Estimación del espectro por el método de máxima entropía	201
F.1.	Espectro de máxima entropía	201
F.2.	Algoritmo eficiente para calcular el espectro MEM	205
F.3.	Código en Matlab para calcular el espectro MEM	206
F.4.	Código en Matlab para la recursión Levinson-Durbing	207
G.	Algoritmo de optimización por cúmulo de partículas (PSO)	209

H. Artículos publicados y sometidos	211
H.1. Artículos en revistas indexadas	211
H.2. Artículos en conferencias indexadas por IEEE	211
H.3. Artículos en conferencias arbitradas	211
H.4. Reportes técnicos	212
H.5. Artículos de divulgación	212
H.6. Otras participaciones	212
H.7. Artículos sometidos en revisión	212

Índice de figuras

1.1. Aplicación de tecnología PLC en interiores	5
1.2. Penetración de los servicios de banda ancha	6
1.3. Cobertura del servicio eléctrico	7
2.1. Respuesta en frecuencia típica de un canal PLC	16
2.2. Modelo general de un canal PLC	17
2.3. Una red simple con nodo en T	18
2.4. Modelo de referencia para una línea de transmisión	20
2.5. Red bipuerto conectada a una fuente de voltaje y una carga	21
2.6. Dos circuitos bipuerto en cascada	21
2.7. Línea de transmisión para una red con topología de nodo en T	22
2.8. Circuito equivalente para la Figura 2.7	23
2.9. Símbolo OFDM perturbado por mezcla de ruido Gaussiano más ruido impulsivo	26
2.10. Ruido impulsivo adicionado a un símbolo OFDM con una densidad de 6000 impulsos por ciclo de potencia (16.6 ms)	27
2.11. Espectro del ruido impulsivo de la Figura 2.10	27
2.12. Impedancia variante, en el tiempo y la frecuencia, de la carga simulada	29
2.13. Respuesta del canal simulado en el tiempo y la frecuencia	30
2.14. Respuestas en frecuencia del canal simulado	31
2.15. La función pulso rectangular $\Pi(t)$ y su espectro en frecuencia	33
2.16. Símbolo OFDM con intervalo de guardia	35
2.17. Simulación Monte Carlo de enlace OFDM para canal PLC	38
3.1. Arreglos básicos para distribuir los símbolos pilotos	47
3.2. Mejora de la estimación de canal mediante ventaneo de la respuesta al impulso	55
3.3. Principio de estimación paramétrica del espectro con modelo estocástico para $H(\omega)$	57
4.1. Un banco de filtros de M-bandas	65
4.2. Análisis/síntesis de M-bandas sobre 2 niveles de resolución	66
4.3. Análisis/síntesis multi-resolución M-bandas biortogonales sobre 2 niveles de resolución	66
4.4. Ejemplo de descomposición uniforme por paquete de ondeletas (análisis en sub-bandas iguales)	68

4.5.	Tiempo de ejecución promedio (tiempo consumido entre n) para la realización de la FFT (fft) y la transformada de ondeleta discreta (FWT-PO de Wavelab) en Matlab	77
4.6.	Gráficas para funciones de escalamiento y ondeleta de la familia Daubechies con $N=1, 3, 4$ y 6	82
4.7.	Gráficas para funciones de escalamiento y ondeleta de la familia Symmlet con $N=4, 6, 8$ y 10	83
5.1.	Mejora de la estimación del canal por umbralado en el dominio de la ondeleta	89
5.2.	Umbralado duro	90
5.3.	Umbralado suave	91
5.4.	Umbralado suave por encogimiento	91
5.5.	Máximo tiempo de dispersión para el canal simulado. Más allá de la muestra No. 76 la magnitud de la respuesta al impulso es despreciable	92
5.6.	Observación de la estimación del canal obtenida mediante un método asistido por pilotos	93
5.7.	Respuesta al impulso estimada a partir de la respuesta en frecuencia observada. La ventana centrada en la mitad de las muestras incluye todas las muestras que pueden ser consideradas ruido puro	94
5.8.	Estimación mejorada del canal mediante umbralado de la respuesta al impulso en el dominio de la expansión de ondeletas	94
5.9.	Escenario estacionario contaminado con ruido Gaussiano más ruido impulsivo con una densidad de 6000 pulsos por ciclo	99
5.10.	Escenario ciclo-estacionario contaminado con ruido Gaussiano más ruido impulsivo con una densidad de 6000 pulsos por ciclo	100
5.11.	Escenario estacionario contaminado con ruido Gaussiano más ruido impulsivo con una densidad de 600 pulsos por ciclo	101
5.12.	Escenario ciclo-estacionario contaminado con ruido Gaussiano más ruido impulsivo con una densidad de 600 pulsos por ciclo	102
5.13.	Escenario estacionario contaminado con ruido prácticamente Gaussiano (ruido impulsivo con una densidad de 10 pulsos por ciclo)	103
5.14.	Escenario ciclo-estacionario contaminado con ruido prácticamente Gaussiano (ruido impulsivo con una densidad de 10 pulsos por ciclo)	104
6.1.	Creación de cúmulo emergente y cúmulo rastreador	111
6.2.	Atracción del cúmulo rastreador por el cúmulo emergente después de algunas iteraciones del algoritmo	112
6.3.	Comparación de la probabilidad de error entre el método de estimación lineal y el inspirado en PSO, con una densidad de ruido impulsivo de 60 impulsos por ciclo de línea	116
6.4.	Comparación entre el método de estimación lineal y el inspirado en PSO, con una densidad de ruido impulsivo de 6000 impulsos por ciclo de línea	117
7.1.	Arquitectura del GPU NVIDIA empleado [111]	123
7.2.	Jerarquía de la memoria del GPU NVIDIA [111]	124

7.3.	Estructura del algoritmo PSO secuencial	128
7.4.	Estructura de la variante de Global_ev para el algoritmo PSO paralelo	129
7.5.	Estructura de la variante Global_ev+up para el algoritmo PSO paralelo	129
7.6.	Estructura de la variante incrustada para el algoritmo PSO paralelo	130
7.7.	Generación de números aleatorios dentro del GPU	132
7.8.	Desempeño de las variantes de PSO en términos de la complejidad de la función de aptitud. En este caso, complejidad (F01) < complejidad (F02) < complejidad (F03)	137
7.9.	Error de variantes PSO, ejecutando la optimización F03, con diferente número de iteraciones	138
7.10.	Tiempo de procesamiento consumido por las variantes de PSO, durante la optimización de F03, como función del número de iteraciones	139
7.11.	Tiempo de procesamiento consumido por las variantes de PSO, durante la optimización de F03, como función del número de partículas	139
7.12.	Aceleramiento para las variantes paralelas de PSO como función del número de iteraciones	140
7.13.	Aceleramiento para las variantes paralelas de PSO como función del número de partículas	140
7.14.	Eficiencia para las variantes paralelas de PSO como función del número de iteraciones	141
7.15.	Eficiencia para las variantes paralelas de PSO como función de número de partículas	141
C.1.	Desempeño con descomposición hasta obtener aproximación con 1 coeficiente . . .	154
C.2.	Desempeño con descomposición hasta obtener aproximación con 4 coeficientes . .	154
C.3.	Desempeño con descomposición hasta obtener aproximación con 16 coeficientes .	155
C.4.	Desempeño con descomposición hasta obtener aproximación con 64 coeficientes .	155
C.5.	Desempeño con descomposición hasta obtener aproximación con 256 coeficientes .	156
D.1.	Escenario estacionario contaminado con ruido Gaussiano más ruido impulsivo con una densidad de 6000 pulsos por ciclo	158
D.2.	Escenario estacionario contaminado con ruido Gaussiano más ruido impulsivo con una densidad de 600 pulsos por ciclo	159
D.3.	Escenario estacionario contaminado con ruido prácticamente Gaussiano (ruido impulsivo con una densidad de 10 pulsos por ciclo)	160
E.1.	Estructura del programa de simulación	162
E.2.	Flujo de datos en el enlace OFDM simulado	164

Índice de cuadros

1.1. Tecnologías y estándares de ingreso a la Internet	5
1.2. Sinopsis de ventajas de la descomposición de ondeletas	12
1.3. Sinopsis de ventajas de la estimación soportada por ondeletas	12
2.1. Parámetros del transreceptor convencional simulado, referidos a la señal analítica	36
4.1. Notación empleada en la sección 4.1	61
7.1. Distribución del costo computacional en función del número de partículas	136
7.2. Distribución del costo computacional en función del número de dimensiones	136
A.1. Notación empleada	148
E.1. Lista de funciones básicas	163

Resumen

La transmisión de datos a través de la red eléctrica, llamada tecnología PLC (Power-line Communication), requiere de técnicas avanzadas de procesamiento digital de señales para enfrentar de la mejor manera posible los efectos de la distorsión y el ruido que predominan en un medio que originalmente no fue diseñado para propósitos de comunicación digital. El propósito es maximizar el aprovechamiento del canal PLC y uno de los requisitos fundamentales para ello es tener conocimiento del estado del canal, lo que se logra mediante técnicas apropiadas para la *estimación del canal*. En este trabajo se propone aplicar técnicas de procesamiento de datos soportadas por la descomposición de ondeletas (*wavelet expansion*) para la estimación de los canales PLC. También se plantea la aplicación en este campo de algunas heurísticas bio-inspiradas, así como el empleo de la arquitectura de cómputo denominada unidad de procesamiento de gráficos (GPU) como un promotor medio para hacer práctica la aplicación de dichas heurísticas, considerando el gran poder de cómputo que demandan, en los sistemas PLC futuros.

En especial, se presenta una metodología para la estimación de la función de transferencia de un canal PLC plagado de ruido impulsivo de alta densidad con fundamento en la eliminación de ruido en el dominio de las ondeletas. Para tal fin, se prueban diferentes técnicas de umbralado con el modelo de ruido conocido como de clase A de Middleton. Los resultados obtenidos a partir de una estimación de mínimos cuadrados (LS) muestran que el mejor desempeño, en términos de la figura de mérito representada por la probabilidad de error en los bits (BER) vs. la relación señal a ruido (SNR), se alcanza cuando la técnica de eliminación de ruido en el dominio de las ondeletas emplea la regla de encogimiento de Donoho y Johnstone.

También, se propone una nueva técnica de estimación semi-ciega inspirada en las propiedades de auto-organización y rastreo de las poblaciones de optimización por cúmulo de partículas (PSO). Los resultados de las simulaciones muestran el potencial del procedimiento de estimación propuesto en entornos PLC ruidosos.

Finalmente, se presenta un estudio empírico sobre la programación paralela de un algoritmo PSO, puesta en práctica en una unidad de procesamiento de gráficos (GPU) con capacidades de multi-hilos, en conjunto con el modelo más reciente de programación paralela. Se demuestra que es posible acelerar considerablemente al algoritmo PSO por medio de una programación paralela simple y directa.

Abstract

The power-line communication technology (PLC) uses advanced digital processing signal techniques to deal with distortion and noise effects on digital communications over a network originally designed for power distribution. The goal is to maximize the channel capacity and one requirement is to count on channel's knowledge, which is obtained by mean of convenient *channel estimation* (CE) methods. In this work, the application to the PLC field of some wavelet-based channel estimation methods is proposed. The application to this field of some bio-inspired heuristics is also studied. A new computing architecture (Graphics Processing Unit, GPU) is considered as a promising way to make such heuristics application practical into the PLC field since they demand high computational power.

Specifically, a methodology for the transfer function estimation of a PLC channel, impaired by a high density impulsive noise, is presented. This methodology is based on the denoising abilities of the wavelet decomposition. Some thresholding techniques are tested with a Middleton's Class A noise model. Departing from a Least Squares (LS) Estimation, the results show that the best performance, in terms of the error probability (BER) vs. SNR as figure of merit, is reached when the wavelet-based denoising is carried out with a shrinkage rule of Donoho and Johnstone.

Also, a new semi-blind channel estimation technique inspired on the self organizing and tracking features of populations of the Particle Swarm Optimization (PSO) algorithm is proposed. Numerical simulations show the potentiality of the proposed estimation procedure in noisy PLC environments.

Finally, an empirical study of a parallelized PSO algorithm, implemented on a Graphic Process Unit (GPU) device with multi-thread support is presented using the most recent model of parallel programming. It is shown that, with the help of a multithreading GPU, is possible to significantly improve the PSO algorithm performance by means of a simple and straightforward parallel programming.

Parte I

Estimación del canal PLC y ondeletas

Capítulo 1

Introducción

La línea eléctrica es una red prácticamente omnipresente, con contactos disponibles prácticamente en cualquier lugar donde se puede requerir una terminal de comunicación, ya sea computadora o cualquier dispositivo electrónico. Esta es, principalmente, la razón por la que la tecnología PLC (*Power-line Communication*, también conocida como *Broadband Power-line Communication* o BPL) es un área de investigación muy atractiva.

Dentro del campo de las comunicaciones digitales, la tecnología PLC representa el formidable reto de transmitir datos a través de un medio que no fue originalmente diseñado para ello sino para la distribución de energía eléctrica. En particular, los investigadores e ingenieros deben considerar algunas características de este medio que lo hacen, en cierta forma, un “pésimo” canal de comunicación [17, 11], a saber:

- Función de transferencia del canal variante en el tiempo y en la frecuencia.
- Comportamiento con gran dependencia de la ubicación y la topología de la red, así como de las cargas conectadas.
- La presencia de diferentes tipos de ruido significativo.

En un canal PLC se presentan atenuaciones de banda angosta, debido a que las señales no se propagan a lo largo de una trayectoria única, sino que sufren de reflexiones provocadas por los desacoplamiento de impedancia que, a su vez, resultan en reforzamientos y cancelaciones de la señal [24]. Así, el canal PLC puede ser considerado como un medio de comunicación multi-trayectoria y con atenuación selectiva en frecuencia. Más aún, debido a los frecuentes cambios en la impedancia, que ocurren cuando se conectan o desconectan cargas (se prenden/apagan focos o cualquier aparato eléctrico), el canal PLC también tiene un comportamiento variante en el tiempo [117].

Para minimizar los efectos de la atenuación, el ruido y, al mismo tiempo, maximizar el aprovechamiento del canal, se emplean técnicas avanzadas de procesamiento de señales y comunicaciones digitales, tales como el multiplexado por repartición de frecuencias ortogonales (*Orthogonal Frequency-Division Multiplexing*, OFDM) y particularmente la técnica de asignación de bits conocida como esquema de multi-tono discreto (*Discrete Multi-Tone*, DMT), que han sido adoptadas

como técnicas básicas dentro del ámbito PLC [9].

La tecnología PLC se considera emergente, es decir, en vías de maduración. Su principal atractivo es que, en teoría, permitiría proporcionar el servicio de Internet y de comunicaciones digitales de alta velocidad a mayores segmentos de la población que cuentan con servicio eléctrico, más que con teléfono, televisión por cable, o cobertura de otras tecnologías de comunicación digital que hagan posible la conexión denominada como de “último kilómetro” (*last mile*), es decir el último tramo de infraestructura que permite llevar el servicio de datos de banda ancha a los usuarios finales. Esta característica de virtual omnipresencia ha hecho especialmente atractiva a la tecnología PLC para los países que han adquirido el compromiso de llevar los servicios de banda ancha a todos sus ciudadanos [123]. En particular, en los países en vías de desarrollo como México existe el interés en disminuir la brecha tecnológica en cuanto a la penetración de la banda ancha y se han propuesto estrategias para la adopción de tecnologías de información y de comunicaciones (ICT, por sus siglas en inglés) [27], dentro de las que se encuentra precisamente la tecnología PLC [6].

Paralelamente, dos organizaciones mundiales han impulsado importantes desarrollos en el sector de la tecnología PLC: la *Open PLC European Research Alliance*, OPERA [114], y su contraparte en América la *HomePlug Power-line Alliance* [69]. Gracias a este impulso y con el reciente avance en el campo de la integración de muy gran escala (VLSI), así como del procesamiento digital de señales (DSP), el sueño del “hogar inteligente” finalmente se ha alcanzado y la factibilidad de una red doméstica soportada por la tecnología PLC es una realidad.

En años recientes el mercado de las comunicaciones y, en particular el de los servicios de Internet, ha presentado un crecimiento espectacular y una competencia feroz, por ello, el desarrollo de nuevas tecnologías se ha enfocado principalmente en aumentar la cobertura de conectividad, disminuyendo el costo y aumentando el ancho de banda. Dicha conectividad, en un sentido más amplio, trasciende a la simple transmisión de datos entre computadoras o ingreso a la Internet, e incluye los servicios de multimedios como, por ejemplo, el esquema denominado *Triple Play*, es decir, voz (Telefonía), banda ancha (Internet) y televisión por la misma conexión. Por lo anterior, resultan muy atractivas aquellas tecnologías que, sobre todo, permiten llevar los servicios de multimedios al interior de los hogares. Hasta el momento, las tecnologías inalámbricas no ofrecen una solución madura con verdadera capacidad multimedia, ni precios competitivos en el mercado masivo de los hogares convencionales [1]. En particular, la tecnología PLC, como alternativa cableada, tiene las siguientes ventajas:

- En términos de desempeño, la tecnología PLC ha probado alcanzar velocidades de hasta 200 Mbps (millones de bits por segundo) con buena calidad en el servicio [1, 10].
- La tecnología PLC posibilita de manera natural la conectividad de virtualmente cualquier aparato doméstico (refrigeradores, lavadoras, etc.) que se encuentre conectado a la línea eléctrica y, con ello, abre la puerta al control y automatización de cualquier lugar como lo prevé el concepto del “hogar inteligente”.

Sin embargo, conviene considerar a las tecnologías de conectividad, sean cableadas o inalámbricas, como soluciones complementarias más que como competidoras, ya que sus campos de aplicación

Cuadro 1.1: Tecnologías y estándares de ingreso a la Internet

Tipo de red	alambradas					inalámbricas		
	Fibra óptica	Cable coaxial	Par trenzado	Línea telefónica	Línea eléctrica	Bandas libres	Bandas reguladas	Satélite
LAN	Ethernet	G.hn	Ethernet	HomePNA, G.hn	G.hn	Wi-Fi, Bluetooth, DECT, Wireless USB		
WAN	PON, Ethernet	DOCSIS	Ethernet	Dial-up, ISDN, DSL	BPL	Muni Wi-Fi	GPRS, iBurst, WiBro/WiMAX, UMTS-TDD, HSPA, EVDO, LTE	Satélite

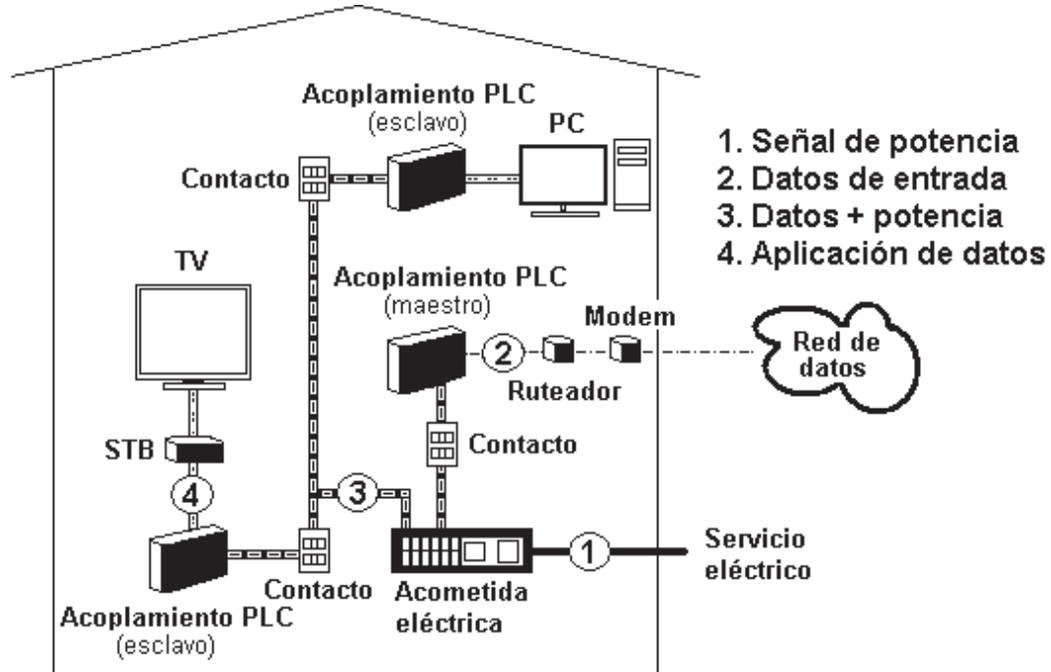


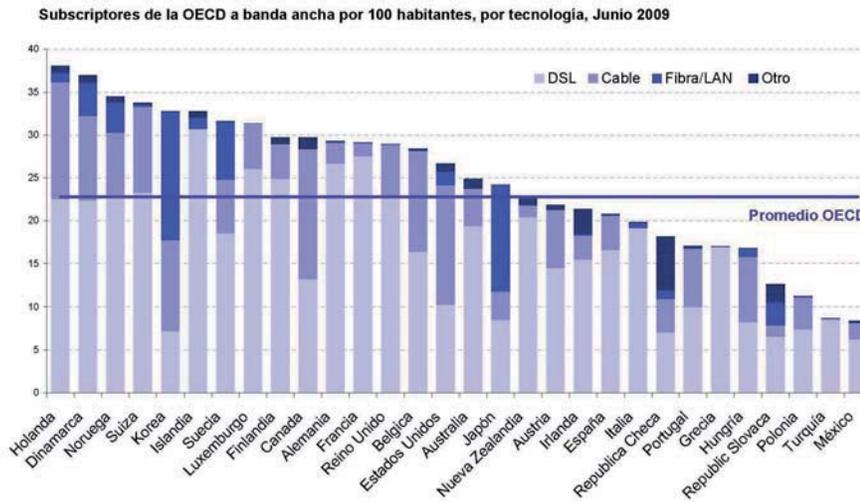
Figura 1.1: Aplicación de tecnología PLC en interiores

son diversos y, sin duda, todas ellas caracterizarán la conectividad de los hogares futuros (ver cuadro 1.1).

1.1. Mercado potencial de la tecnología PLC

Como ya se mencionó, la tecnología PLC se considera como una atractiva alternativa para proporcionar el servicio de banda ancha en el “último kilómetro”. De hecho, existen en la tecnología PLC tres importantes campos de aplicación:

- tecnología PLC en exteriores (*Outdoor PLC*),
- tecnología PLC en interiores (*Indoor PLC*) y



Fuente: OECD (2009)

Figura 1.2: Penetración de los servicios de banda ancha

- tecnología PLC dentro de vehículos.

La tecnología PLC de exteriores posibilita el empleo de la red de distribución eléctrica de mediana tensión (de la subestación eléctrica hasta el lado primario del transformador de baja tensión) para llevar un punto de ingreso a la red (*access point*) a cierto grupo de usuarios. La tecnología PLC en interiores considera el enlace desde el lado secundario del transformador de baja tensión hasta los contactos dentro de los domicilios de los usuarios finales (ver Figura 1.1). Por último, en la tecnología PLC aplicada a vehículos (terrestres, aéreos y marítimos), se emplea dentro de automotores, aviones y barcos como red local de comunicación de datos. En todos los casos mencionados arriba, el punto en común es que no se necesita de un nuevo cableado para la red de comunicación sino que simplemente se emplea a la red eléctrica existente como medio de propagación de datos.

La aparente desventaja de la tecnología PLC respecto de la movilidad de los usuarios se resuelve mediante el empleo de esquemas híbridos de conectividad, donde se recurre a tecnologías de comunicación inalámbricas para cubrir el aspecto de movilidad mientras que se emplea a la red eléctrica como un medio principal para acercar los puntos de ingreso a los usuarios finales.

En este sentido, el principal mercado potencial de la tecnología PLC es, sin duda, el de las comunidades rurales y aisladas, donde los prestadores de servicios convencionales de banda ancha no están interesados en invertir debido al reducido mercado que les representa, pero que cuentan con cobertura del servicio eléctrico y entonces con la posibilidad de acceder al servicio de banda

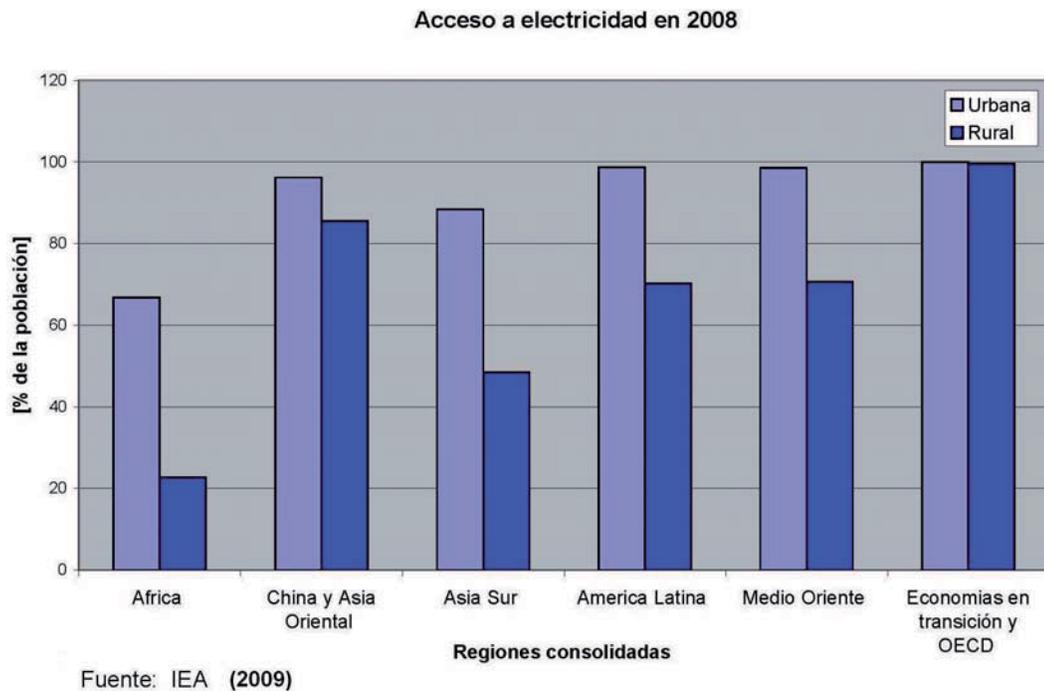


Figura 1.3: Cobertura del servicio eléctrico

ancha mediante la tecnología PLC.

El potencial de aplicación de la tecnología PLC se puede vislumbrar al observar las brechas en cuanto a penetración de los servicios de banda ancha a nivel mundial (ver Figura 1.2), así como la cobertura del servicio eléctrico alcanzada tanto en zonas urbanas como en zonas rurales (ver Figura 1.3). Es claro que la penetración de los servicios de banda ancha tiene aún que alcanzar mayores niveles que los actuales pero, sobre todo, que la red de distribución eléctrica tiene una mayor cobertura, en relación con la penetración de los servicios de banda ancha, en cada una de las regiones del mundo que se reportan.

1.2. Maximización en el aprovechamiento del canal PLC

La tecnología PLC surge de la idea de montar una señal de comunicación sobre la senoide que distribuye la energía eléctrica. Entonces se requiere de circuitos de acoplamiento que permiten enlazar dos sistemas que tienen características antagónicas, a saber [81]:

- La energía eléctrica se distribuye con una senoide de gran potencia y baja frecuencia (120V/60 Hz).
- La señal de comunicación es de baja potencia y de alta frecuencia (3-30 MHz).

Resuelto el asunto del acoplamiento, una cuestión importante consiste en determinar cuál es la máxima velocidad de transmisión de datos que se puede alcanzar en una red PLC, así como su

respectiva calidad de comunicación. No obstante, a pesar de los importantes esfuerzos de investigación que se han enfocado en el desarrollo sistemas PLC que operen cerca del límite de Shannon, esta pregunta aún permanece sin respuesta [7].

Como sabemos, la capacidad de un canal de comunicación es la máxima velocidad de transmisión de datos que éste permite. A partir de los resultados de la teoría de información de Shannon, se concluye que la única forma de alcanzar una velocidad de datos cercana a la capacidad teórica del canal es mediante la optimización del sistema de comunicación como, por ejemplo, mediante la optimización del esquema de modulación. En este sentido, se han propuesto diversos esquemas para las redes PLC [41], pero los esquemas más ampliamente adoptados en el ámbito PLC son aplicaciones de la técnica denominada como OFDM-DMT, que emplea modulación multi-portadora para dividir eficientemente un flujo secuencial de datos entrantes en muchos de flujos paralelos de datos, aunque de menor velocidad, asignados a un conjunto de subportadoras (subcanales) ortogonales. Algunas ventajas de esta técnica son la adaptación de la velocidad de datos en función de la relación señal a ruido (*Signal to Noise Ratio*, SNR), eficiencia espectral, resistencia a la interferencia entre símbolos (*Inter-Symbol Interferente*, ISI), inmunidad al ruido impulsivo y minimización de la interferencia de radio frecuencia (RFI, por sus siglas en inglés, también conocida como compatibilidad electromagnética o EMC).

Las técnicas de asignación de bits aplicadas dentro del esquema OFDM-DMT, estiman la SNR de cada canal en una sesión inicial y se reparte el número total de bits por transmitir en los subcanales de acuerdo a la SNR de cada uno de estos, de tal manera que los subcanales con mayor SNR transmitirán más bits que el resto. Así, cada símbolo DMT mapea tantos bits como lo requiere el nivel de atenuación y el ruido observado en la frecuencia de su subcanal.

En todos estos procesos se asume un conocimiento perfecto del estado del canal [9], es decir, la determinación ideal de los coeficientes complejos de atenuación de la señal (la función de transferencia) y la potencia del ruido de cada subcanal [1]. De lo antes expuesto, se desprende que el tema de la estimación del canal es un asunto de gran importancia en el diseño de un sistema OFDM y, por ende, para la realización de un sistema PLC con base en estos esquemas de modulación multiportadora.

1.3. Motivación

El procedimiento de estimación del canal es necesario en cualquier sistema OFDM ya que

- Los sistemas OFDM emplean detección coherente y requieren un conocimiento perfecto de la fase, de la atenuación del canal y de la potencia del ruido en cada subportadora.
- El conocimiento del canal posibilita que el sistema adapte la señal transmitida según el canal de propagación (*modulación adaptativa*).

Asumiendo que la topología y las cargas de una red PLC cambian esporádicamente (manteniéndose sin cambios por minutos, incluso por horas), se puede afirmar que el canal PLC es un

canal ciclo-estacionario ya que sólo presenta variaciones de corta duración sincronizadas con la frecuencia de la línea (60 Hz). Desde la perspectiva de las señales de comunicación, las variaciones de corta duración del canal son graduales y el canal PLC puede ser visto entonces como un sistema lineal a la vez que periódicamente variante en el tiempo (LPTV, por sus siglas en inglés) [23]. Así la tarea de estimación y seguimiento del canal PLC representa un problema significativo debido a que:

- Se trata de un canal selectivo en frecuencia y variante en el tiempo.
- Existe la presencia de diversos tipos de ruido, donde el ruido que produce más daño es el ruido impulsivo.

1.4. Planteamiento del problema

En lo arriba expuesto se resume la problemática que se aborda en este trabajo. Se trata de obtener una aproximación (estimación) de la función de transferencia de un canal que tiene las agravantes de presentar las distorsiones en magnitud y fase propias de un canal selectivo en frecuencia, estar sujeto a interferencia producida por un ambiente ruidoso y requerir del seguimiento de las variaciones de corta duración a fin de optimizar al sistema.

1.5. Propuesta de la tesis

Partiendo del hecho de que realizar una estimación implica obtener una aproximación de cierta señal o función (en este caso la función de transferencia) y de que dicha aproximación se hace a partir de observaciones ruidosas, se puede ver el problema como un asunto de eliminación de ruido de la observación. En la literatura podemos encontrar una muy abundante cantidad de trabajos que han abordado este problema desde muy diferentes enfoques, dentro de los cuales sobresalen los enfoques clásicos con fundamento en filtros lineales óptimos (filtros Wiener) o sub-óptimos. En general, los filtros óptimos requieren de las estadísticas del canal y son de difícil realización práctica ya que pueden involucrar el cálculo de la inversa de una matriz con gran demanda de recursos computacionales.

El arribo del análisis de señales mediante la descomposición con bases de ondeletas (ondas de corta duración) introdujo un nuevo enfoque para realizar estimaciones con base en la información resultante del denominado análisis multi-resolución (MRA, por sus siglas en inglés) de Mallat [96]. Donoho [46] notó que la “heurística de Mallat” conduce a la noción de que una función con suavidad variante (con “quebres” y “picos’intermitentes conforme se recorre el dominio) puede representarse económicamente (representación más simple) por una base de ondeletas suaves y, por lo mismo, ser más fácilmente estimada [75]. Aquel primer trabajo sobre la estimación con base en la ondeleta derivó en técnicas de eliminación de ruido que, con ayuda de reglas de umbralado apropiadas, obtuvieron mejores resultados que los filtros lineales con fundamento en las estadísticas de la señal, como es el caso de los filtros Wiener o los filtros de promedio.

En esta tesis se aborda el problema de la estimación del canal, fundamentalmente con el enfoque de la descomposición de ondeletas discretas (DWT, por sus siglas en inglés), con el propósito de enfrentar mejor dos condiciones de las aplicaciones PLC: un canal ruidoso y un comportamiento ciclo-estacionario. Es decir, se trata de mejorar la extracción de la función de transferencia a partir de una observación ruidosa y, luego, de seguir las variaciones de la misma a lo largo del tiempo lo más fielmente posible.

Aunque en esta tesis se aborda el problema de la estimación del canal fundamentalmente con el enfoque de la descomposición de ondeletas, durante la investigación también se exploraron otras posibilidades para la estimación del canal PLC. En particular, se estudió la aplicación de heurísticas bio-inspiradas, tanto para la estimación del canal PLC como para su seguimiento. Debido a que las heurísticas bio-inspiradas consideradas, tales como el algoritmo de cúmulo de partículas (PSO, por *Particle Swarm Optimization*), demandan importantes recursos computacionales que por ahora se encuentran fuera del alcance de los actuales procesadores digitales de señales (DSP), se incursionó en el empleo de nuevas arquitecturas de cómputo (como es el caso de las unidades de procesamiento de gráficos o GPU, por sus siglas en inglés) que plantean también nuevos paradigmas de programación y que, según las tendencias actuales, se prevé que, en un futuro no muy lejano, dichos recursos computacionales también se encuentren disponibles en los nuevos DSP que salgan al mercado, con lo que se haría perfectamente realizables aplicaciones de tiempo real que contengan heurísticas bio-inspiradas, tanto en el campo del procesamiento digital de señales como en el campo de las comunicaciones digitales.

1.5.1. Alcance

Este trabajo se planteó desde un principio con la premisa de realizar contribuciones para maximizar el aprovechamiento del canal PLC. En particular, este trabajo se enfocó en los entornos contaminados por ruido impulsivo de gran densidad y en las alternativas de solución para enfrentar mejor este problema. Entonces, fundamentalmente se trató de minimizar el efecto del ruido impulsivo para incrementar la capacidad del canal de comunicación mediante el empleo de técnicas avanzadas de procesamiento de señales (en particular la aplicación de la descomposición de ondeletas discretas), la aplicación de heurísticas bio-inspiradas, así como el empleo de nuevas arquitecturas de cómputo masivo.

Durante el desarrollo de esta investigación, las vertientes de investigación seguidas fueron las siguientes:

1. Primero, fue necesario programar el entorno de simulación tanto para el canal como para el transreceptor. Lo anterior debido a que dicho entorno es necesario para la realización de las simulaciones Monte Carlo que permiten comparar el desempeño de las técnicas propuestas en términos de la figura de mérito conocida como probabilidad de error en bits (BER) vs. la relación señal a ruido (SNR). Se puso especial énfasis en el modelo de ruido y se adoptó el modelo de ruido de clase A de Middleton. También se incorporaron parámetros para obtener un canal selectivo en frecuencia y, opcionalmente, con características de ciclo-estacionalidad.

2. Un vez que se completó el entorno de simulación, se investigó la aplicación de la descomposición de ondeletas para la estimación del un canal PLC plagado de ruido impulsivo. En esta etapa quedó en claro que un canal con variaciones temporales de corta duración, como puede ser el caso del canal PLC, requiere adicionalmente de un proceso de seguimiento del canal para minimizar los errores en la detección debidos al movimiento de los coeficientes complejos de igualamiento.
3. Después, justamente para dar seguimiento al canal, se consideró la aplicación de heurísticas bio-inspiradas en virtud de las propiedades intrínsecas de rastreo y seguimiento de algunas heurísticas poblacionales como el algoritmo de optimización por cúmulo de partículas (PSO, por sus siglas en inglés) o el de evolución diferencial (DE, por sus siglas en inglés). En esta etapa se obtuvieron resultados prometedores para enfrentar al ruido impulsivo, pero el tiempo consumido en cada simulación sugirió que tales propuestas no serían prácticas a menos de contar con gran poder computacional a nivel de los dispositivos de procesamiento de señales (DSP) empleados en los módems PLC comerciales.
4. Finalmente, se inició la migración de los algoritmos bio-inspirados y poblacionales a una arquitectura de cómputo masivo que emplea el poder computacional disponible en las tarjetas de procesamiento de gráficos (GPU). En esta etapa, sólo se lograron realizar pruebas de desempeño de los algoritmos poblacionales programados en paralelo, pero arrojó interesantes resultados sobre los efectos de emplear un GPU multi-hilos para ejecutar paralelamente dichas heurísticas. Al respecto de este tópico, queda como trabajo futuro migrar al GPU el entorno de simulación y los algoritmos de estimación y seguimiento propuestos.

1.6. Justificación

En la mayoría de los trabajos sobre estimación soportada por la descomposición de ondeletas se asume que el ruido contaminante es Gaussiano. Entonces resulta de gran interés investigar el potencial de las ondeletas para enfrentar una mezcla de ruido Gaussiano con ruido impulsivo de gran densidad. En particular, los resultados de la investigación tienen aplicación a los entornos PLC donde el ruido impulsivo es la principal fuente de errores y, por lo tanto, la principal limitante para maximizar el aprovechamiento del canal [9].

En el cuadro 1.2 se muestran las características más importantes de la transformada de ondeleta en sí misma, mientras que en el cuadro 1.3 se pueden apreciar las características inherentes a la estimación soportada por ondeletas, en particular de la estimación mediante eliminación de ruido por umbralado.

En términos generales podemos resaltar las siguientes ventajas que hacen atractivo el empleo de la descomposición de ondeleta para la estimación del canal:

- La estimación no lineal obtenida con base en la descomposición de ondeletas ha probado ser más eficaz (efectiva y eficiente) que la estimación de los filtros lineales con fundamento en las estadísticas de los datos y análisis de Fourier puro.

Cuadro 1.2: Sinopsis de ventajas de la descomposición de ondeletas

Característica	FFT	DWT
Localización frecuencia-tiempo	No	Sí
Representación económica	generalmente No	generalmente Sí
Eficiencia (complejidad temporal)	$n \log(n)$	n

Cuadro 1.3: Sinopsis de ventajas de la estimación soportada por ondeletas

Característica	Clásica	Ondeletas
Paramétrica	Sí	No
Requiere estadísticas del canal	Sí	No
Linealidad	Sí	No
Costo computacional	Calcula matrices de autocorrelación	Regla de umbralado
Eficacia al eliminar ruido	Aceptable	Buena

- Las técnicas de estimación lineales clásicas usualmente requieren del conocimiento de las estadísticas del canal y pueden necesitar el cálculo de la inversa de alguna matriz, todo lo cual es muy demandante de recursos de cómputo.
- Existen técnicas muy rápidas para calcular la transformada de ondeleta discreta (DWT).
- Particularmente, dado que la tecnología PLC es una realidad, cualquier mejora en la estimación del canal es de gran interés ya que ello aumenta la capacidad del canal de comunicación.

Adicionalmente, considerando la exitosa aplicación de las heurísticas bio-inspiradas en diversas disciplinas de la ciencia aplicada y dada la escasez de trabajos publicados en lo relativo a la aplicación de estas heurísticas al campo de la estimación del canal, se exploró el potencial de algunos algoritmos poblacionales bio-inspirados aplicados para este propósito.

1.7. Hipótesis

La descomposición en sub-espacios que realiza la Transformada de Ondeleta Discreta (DWT, por sus siglas en inglés) produce representaciones más simples (económicas) con elementos de información que son más fáciles de modelar y de estimar, eliminando el ruido de una señal en forma efectiva. Asumiendo a la función de transferencia del canal PLC como la señal por aproximar

y considerando que el canal PLC está plagado de ruido, es factible aprovechar las cualidades de la descomposición de ondeletas para enfrentar el ruido y obtener una estimación mejorada.

1.8. Objetivo general

Aplicar al ámbito PLC técnicas de estimación de canal con el enfoque de la descomposición de ondeletas, técnicas bio-inspiradas y cómputo intensivo.

1.9. Objetivos particulares

1. Realizar el modelo del canal PLC y el simulador del transreceptor OFDM-DMT, para llevar a cabo simulaciones Monte Carlo y evaluar el desempeño de las técnicas propuestas.
2. Estimar el canal PLC en el dominio de la transformada de ondeleta discreta (DWT).
3. Estimar el canal PLC con técnicas bio-inspiradas.
4. Poner en práctica en forma paralela las heurísticas bio-inspiradas, empleadas para la estimación de canal, en unidades de procesamiento de gráficos (GPU) y evaluar el desempeño.

1.10. Aportaciones

En este trabajo se realizan aportaciones al campo de la estimación del canal PLC mediante el aprovechamiento de las ventajas ofrecidas por el procesamiento de datos con base en la transformada de ondeleta. También se plantea la aplicación en este campo de algunas heurísticas bio-inspiradas, así como el empleo de la arquitectura de cómputo denominada unidad de procesamiento de gráficos (GPU) como un prometedor medio para hacer práctica la aplicación de dichas heurísticas, considerando el gran poder de cómputo que demandan, en los sistemas PLC futuros. Las aportaciones se pueden resumir como sigue:

- Entorno de simulación para el canal PLC y el transreceptor OFDM. Esta herramienta queda a disposición de los interesados para que puedan usarla en sus investigaciones ya que no existe una herramienta similar disponible con código abierto.
- Modelo del canal PLC y estimación del canal con el enfoque de la descomposición de ondeletas, en un entorno afectado por una mezcla de ruido Gaussiano y ruido impulsivo de gran densidad, que es el ambiente de operación para los canales PLC que se encuentra frecuentemente en la práctica. En la literatura, no existe publicado ningún trabajo donde se aplique la descomposición de ondeletas para enfrentar el ruido impulsivo para la estimación del canal PLC.

- Algoritmo de optimización por cúmulo de partículas (PSO) para la estimación semi-ciega y seguimiento de canales PLC. La aplicación de heurísticas bio-inspiradas para la estimación del canal es un campo poco explorado y por lo mismo existen pocas publicaciones al respecto.
- realización paralela de la heurística bio-inspirada PSO en un procesador de gráficos (GPU) y evaluación de su desempeño. Debido a que este es un campo de muy reciente aparición, los resultados producidos por esta vertiente de la investigación han generado gran interés y han sido bien acogidos para su publicación.

1.11. Plan de tesis

Este trabajo se compone del capítulo que contiene esta introducción más 7 capítulos con el cuerpo principal de la tesis.

En el capítulo 2 se presenta una revisión de los modelos de canal, tanto para la función de transferencia como para el ruido, así como una breve especificación del simulador empleado para la simulación numérica del canal y del transreceptor OFDM empleado.

En el capítulo 3 se hace una reseña de las técnicas de estimación de canal que típicamente han sido adaptadas para el caso PLC.

En el capítulo 4 se presenta una breve revisión de la evolución (*state of the art*) de la transformada de ondeleta, nociones sobre la transformada de ondeleta discreta (DWT) y sus aplicaciones al campo de la estimación.

En el capítulo 5 se presenta el desarrollo de las técnicas propuestas de estimación de canal PLC soportadas por la descomposición de ondeletas.

En el capítulo 6 se presenta una nueva técnica de estimación ciega con base en el algoritmo de optimización por cúmulo de partículas (PSO).

En el capítulo 7 se presenta la programación paralela del algoritmos de optimización por cúmulo de partículas (PSO) en previsión de la eventual disponibilidad de dicha tecnología a nivel de los dispositivos de procesamiento de señales (DSP), lo que posibilitaría la aplicación de dichas heurísticas tanto para el procesamiento de señales como para la estimación de canal en los equipos de comunicación digital y, en particular, en los transreceptores PLC reales. Finalmente, en el capítulo 8 se presentan las conclusiones.

Capítulo 2

Modelo para el canal PLC y simulación del transreceptor OFDM

En este capítulo se presenta una breve introducción a los modelos existentes para el canal PLC. Se describe el modelo de múltiples trayectorias y los modelos analíticos de matrices. También se presenta el modelo de ruido con especial énfasis en el modelo de clase A de Middleton (mezcla de ruido Gaussiano más ruido impulsivo) y el entorno de simulación OFDM con el correspondiente modelo de señal que se emplea a lo largo de este trabajo.

2.1. Los modelos existentes para un canal PLC

Aunque muchos investigadores han propuesto diferentes modelos para el canal PLC, hasta ahora no existe un modelo universalmente aceptado [139]. El canal PLC es extremadamente difícil de modelar debido a que su respuesta es variante tanto en el tiempo como en la frecuencia (ver Figura 2.1); a que su comportamiento es muy dependiente de la ubicación y topología de la red; y a que normalmente se encuentra contaminado por diferentes tipos de ruido significativo [11]. Independientemente del enfoque, en todos los modelos se recurre a extensas campañas de mediciones, tanto para validar como para parametrizar al modelo [82].

Un esquema general ampliamente utilizado para representar al modelo de canal PLC es el mostrado en la Figura 2.2 [117, 17]. Por otro lado, podemos clasificar a los modelos del canal PLC, de acuerdo con el objeto de estudio, como sigue:

- Modelado de la función de transferencia con enfoque empírico. Este es un enfoque fenomenológico y podemos decir que en este caso la función de transferencia, $H(f, t)$, se calcula con una metodología *a posteriori*. Un modelo empírico determina los parámetros del modelo a partir de mediciones. Esta técnica de modelado se pone en práctica fácilmente y es eficiente desde el punto de vista computacional, pero es propensa a los errores de medición.
- Modelado de la función de transferencia con enfoque determinista. Este es un enfoque analítico y podemos decir que en este caso la función de transferencia, $H(f, t)$, se calcula con una

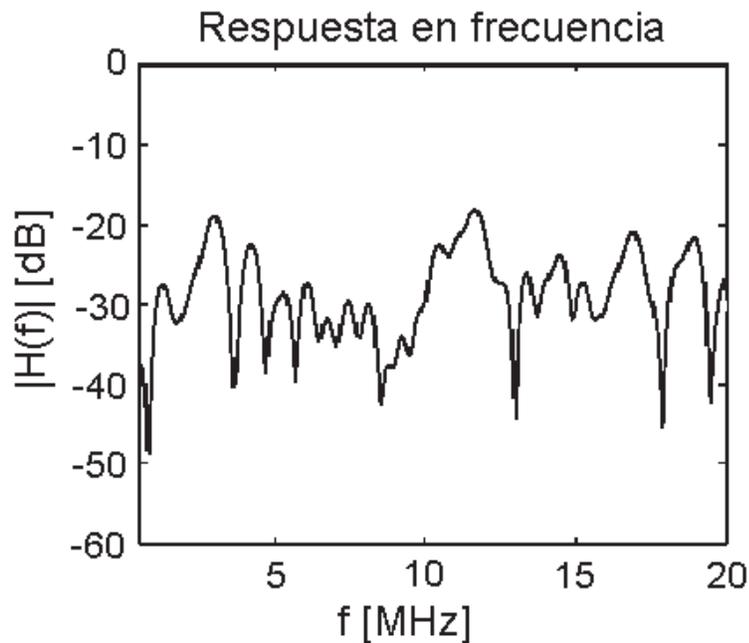


Figura 2.1: Respuesta en frecuencia típica de un canal PLC

metodología *a priori*. Un modelo determinista obtiene los parámetros del modelo con fundamento en la teoría. Esta técnica de modelado es, desde el punto de vista computacional, más intensiva con respecto al enfoque empírico, pero permite predecir los efectos de cualquier cambio en la red sobre la función de transferencia, ya que todos los parámetros se encuentran formulados [98].

- Modelado de Ruido. La densidad espectral del ruido es muy dependiente del lugar y de la hora. Más aún, las mediciones muestran que ésta se compone de ruido de fondo “cromático” (es decir, ruido con espectro no uniforme), ruido impulsivo periódico (tanto síncrono como asíncrono, respecto a la línea de 60 Hz), ruido de banda angosta (debido fundamentalmente a radiodifusoras AM y radio aficionados) y ruido impulsivo asíncrono (ocasionado por la conexión y desconexión de cargas, así como por fallas en la línea eléctrica) [24]. Típicamente, a fin de modelar el ruido aditivo, se extraen sus parámetros estadísticos después de extensas mediciones y se construye un modelo con base en variables aleatorias y cierta función de densidad de probabilidad (pdf, por sus siglas en inglés) [61].

A su vez, cualquiera de los enfoques de modelado para la función de transferencia, ya sea empírico o determinista, puede ser desarrollado tanto en el dominio del tiempo como en el de la frecuencia:

- Dominio del tiempo. El canal PLC es visto como un ambiente de múltiples trayectorias (multi-trayectoria) y típicamente se describe al sistema con un “modelo de eco”. Aunque un modelo de eco es fácil de ponerse en práctica con un enfoque de modelado empírico, su complejidad se incrementa dramáticamente cuando cada unión del cableado tiene múltiples

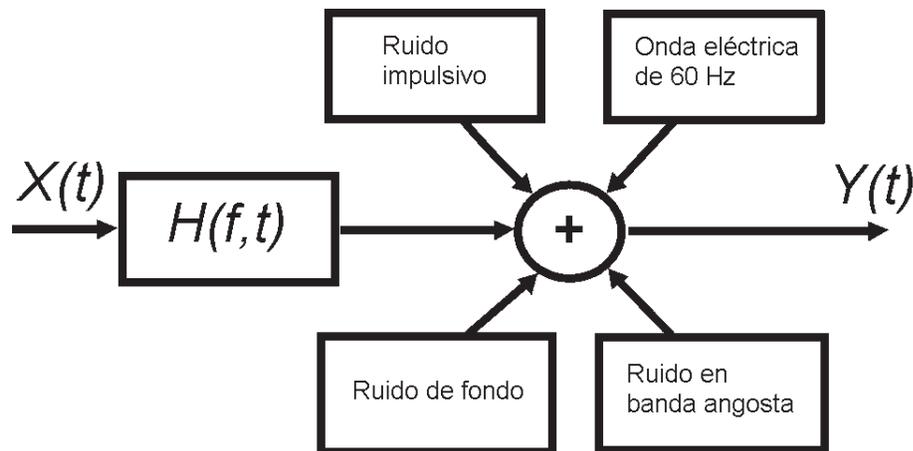


Figura 2.2: Modelo general de un canal PLC

derivaciones [98]. Hasta ahora, el más conocido de los modelos empíricos para el canal PLC es el modelo de multi-trayectoria propuesto tanto por Philipps [119] como por Zimmermann y Dostert [145].

- **Dominio de la frecuencia.** En este caso, la red PLC se descompone en partes más pequeñas que pueden ser vistas como elementos distribuidos en cascada, cada uno de estos con características de sistema descritas por matrices de transmisión o por matrices de dispersión. La gran ventaja de la representación matricial es que, independientemente de la complejidad de la red, ésta considera en forma intrínseca todas las discontinuidades de impedancia. La mayoría de los modelos deterministas para canal PLC emplean una representación matricial.

Para tener un panorama general sobre los diferentes modelos para el canal PLC que se proponen en la literatura reciente, tanto para la función de transferencia como para el ruido, el lector puede referirse a [83].

2.2. Modelo de múltiples trayectorias

La ventaja del modelo de multi-trayectoria es que una vez que las mediciones han sido realizadas y que los parámetros han sido obtenidos, la simulación del canal requiere poca potencia computacional al mismo tiempo que la puesta en práctica del algoritmo es relativamente simple. La desventaja es que si la red cambia, entonces se hace necesario realizar de nuevo el conjunto de mediciones de parametrización. Lo anterior es un serio inconveniente si se pretende emplear el modelo de multi-trayectoria como base para una herramienta de simulación de propósito general. Sin embargo es muy conveniente y económico si se trata de representar el efecto de un hipotético canal con dispersión temporal sobre las señales transmitidas.

La metáfora del eco surge debido a que la red de distribución eléctrica se compone de muchas derivaciones que se originan en las uniones del cableado y que resultan en múltiples trayectorias so-

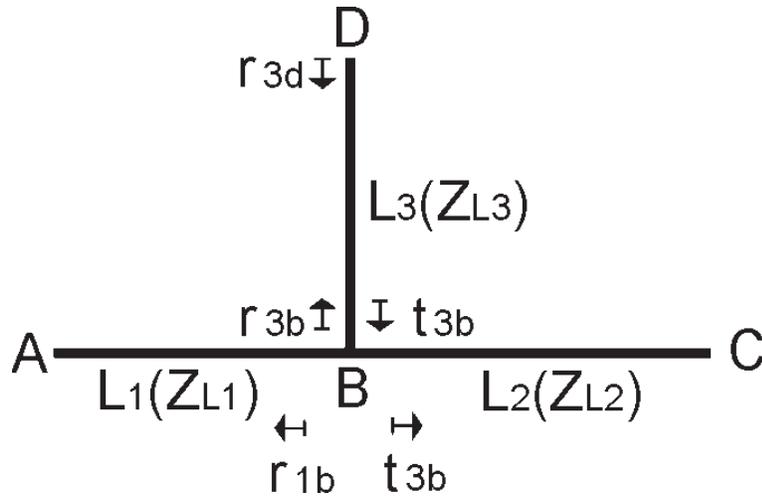


Figura 2.3: Una red simple con nodo en T

bre las que se propagan las señales transmitidas. Debido a que la impedancia característica así como la impedancia de las cargas conectadas a las derivaciones usualmente no se encuentran acopladas, una parte de la señal es reflejada en cada unión, carga y circuito abierto. La señal transmitida y los ecos arribados constituyen al fenómeno de propagación multi-trayectoria que produce, en el punto de recepción, una señal compleja de amplitud atenuada. Debido a que la impedancia característica de los cables y la impedancia de las cargas conectadas cambian con la frecuencia de la portadora, los coeficientes de transmisión y de reflexión en cada unión producen diferentes patrones multi-trayectoria para cada frecuencia, lo cual resulta en un canal atenuante y selectivo en frecuencia, como puede verse en el ejemplo de la Figura 2.1.

A fin de ilustrar el fenómeno de propagación multi-trayectoria, el empleo de una red simple con un nodo en T puede ser muy útil. La Figura 2.3 muestra una red con topología en T en la que las derivaciones L_2 y L_3 , con longitudes l_1 , l_2 y l_3 , son conectadas en el punto B. Las impedancias características de las tres líneas son Z_1 , Z_2 y Z_3 , respectivamente.

Por simplicidad, podemos asumir que las impedancias terminales en los puntos A y C están acopladas con la red, lo que significa que $Z_A = Z_{L1}$ y $Z_C = Z_{L2}$. Así, la señal es reflejada solamente en los puntos B y D. De acuerdo a la teoría de los campos electromagnéticos, los factores de reflexión en los puntos B y D pueden ser escritos, respectivamente, como sigue [144]:

$$r_{1b} = \frac{(Z_{L2} || Z_{L3}) - Z_{L1}}{(Z_{L2} || Z_{L3}) + Z_{L1}}$$

$$r_{3d} = \frac{Z_D - Z_{L3}}{Z_D + Z_{L3}}$$

$$r_{3b} = \frac{(Z_{L2} || Z_{L1}) - Z_{L3}}{(Z_{L2} || Z_{L1}) + Z_{L3}}$$

mientras que los factores de transmisión están dados por

$$\begin{aligned} t_{1b} &= 1 - |r_{1b}| \\ t_{3b} &= 1 - |r_{3b}| \end{aligned}$$

Por otro lado, el tiempo de retardo para la i – ésima trayectoria esta definido por

$$\tau_i = \frac{d_i \sqrt{\epsilon_r}}{c_0} = \frac{d_i}{v_p} \quad (2.1)$$

donde ϵ_r es la constante dieléctrica, c_0 es la velocidad de la luz en el vacío y d_i es la longitud de la i – ésima trayectoria. Con fundamento en el análisis de las reflexiones, y después de considerar algunas simplificaciones hechas a partir del comportamiento observado en las campañas de mediciones, la función de transferencia de la red puede ser representada por la siguiente ecuación simplificada [8]:

$$H(f) = \sum_{i=1}^N \underbrace{g_i}_{\text{ganancia}} \cdot \underbrace{A(f, d_i)}_{\text{atenuación}} \cdot \underbrace{e^{-j2\pi f \tau_i}}_{\text{retraso}} = \sum_{i=1}^N C_i \cdot e^{-j2\pi f \tau_i} \quad (2.2)$$

donde, para N trayectorias relevantes, g_i representa el factor de peso que cambia la amplitud debido al fenómeno de reflexión/transmisión, $A(f, d_i)$ es el término de atenuación dependiente tanto de la frecuencia como de la longitud de propagación, y $e^{-j2\pi f \tau_i}$ se refiere a la diferencia de fase debida a los retrasos.

Después de realizar extensas mediciones en el rango de los 500kHz a los 40MHz, Zimmermann y Dostert propusieron el siguiente modelo para el término de atenuación:

$$A(f, d_i) = e^{-(a_0 + a_1 f^k) d_i} \quad (2.3)$$

donde a_0 , a_1 son los parámetros de atenuación y k es un exponente con valores típicos entre 0.2 y 1. Finalmente, en el dominio dual obtenido mediante transformada de Fourier, la respuesta al impulso del canal se puede expresar en forma simple con base en el modelo de múltiples trayectorias:

$$h(t) = \sum_{i=1}^N C_i \cdot \delta(t - \tau_i) \quad (2.4)$$

El lector puede obtener una descripción detallada del procedimiento de parametrización para el modelo de multi-trayectoria en [8].

2.3. Modelo analítico con matrices de transmisión

Debido a que, en principio, los cables de la fase y el neutro de la línea eléctrica corren paralelamente, es posible modelarlos como una línea de transmisión. De la teoría de las líneas de

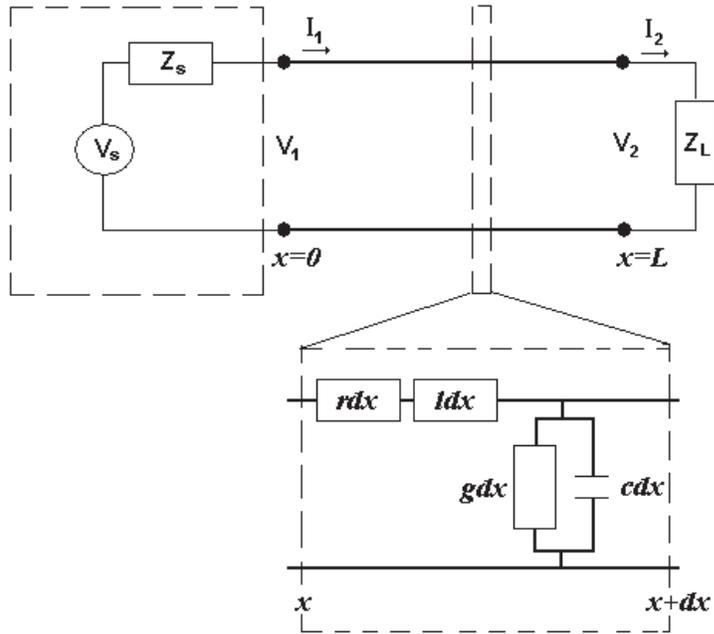


Figura 2.4: Modelo de referencia para una línea de transmisión

transmisión, resultan las siguientes expresiones que describen el voltaje y la corriente a lo largo de una línea (ver Figura 2.4):

$$V_1(x) = V_2 \cosh(\underline{\gamma}x) + I_2 \underline{Z} \sinh(\underline{\gamma}x) \quad (2.5)$$

$$I_1(x) = I_2 \cosh(\underline{\gamma}x) + \frac{V_2}{\underline{Z}} \sinh(\underline{\gamma}x) \quad (2.6)$$

donde \underline{Z} es la impedancia característica y $\underline{\gamma}$ es la constante de propagación que, a su vez, pueden ser calculadas con base en los parámetros por unidad de resistencia r , inductancia l , conductancia g y capacitancia c del cable empleado [116]:

$$\underline{Z} = \sqrt{\frac{r + j\omega l}{g + j\omega c}} \quad (2.7)$$

$$\underline{\gamma} = \sqrt{(r + j\omega l)(g + j\omega c)} \quad (2.8)$$

con r en Ω/m , l en H/m , g en S/m y c en F/m .

Las ecuaciones (2.5) y (2.6) pueden expresarse en su forma matricial, representando un circuito o red de dos puertos (bipuerto), tal y como se muestra en la Figura 2.5:

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_2 \\ I_2 \end{bmatrix} \quad (2.9)$$

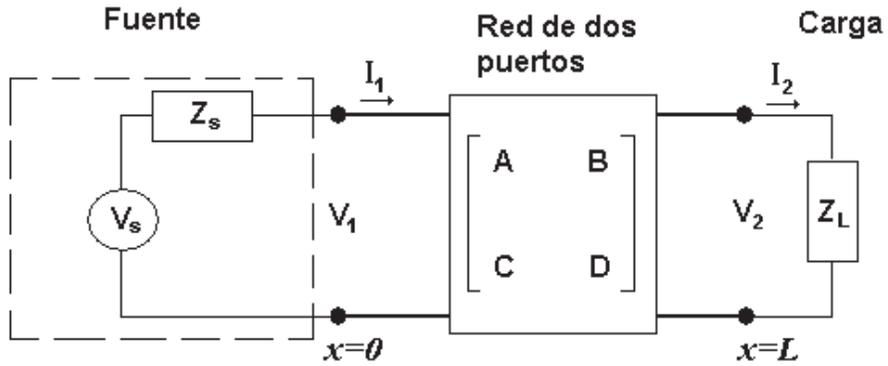


Figura 2.5: Red bipuerto conectada a una fuente de voltaje y una carga

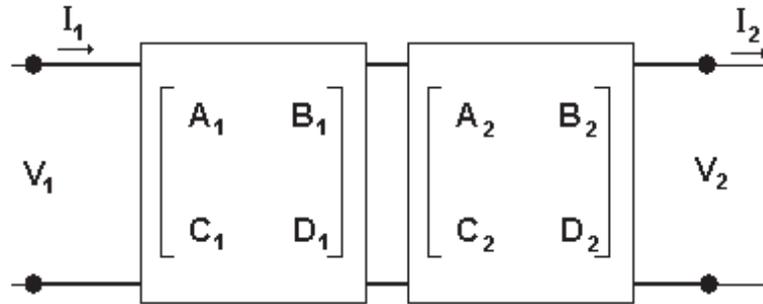


Figura 2.6: Dos circuitos bipuerto en cascada

con

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} \cosh(\underline{\gamma}L) & \underline{Z} \sinh(\underline{\gamma}L) \\ \frac{1}{\underline{Z}} \sinh(\underline{\gamma}L) & \cosh(\underline{\gamma}L) \end{bmatrix} \quad (2.10)$$

esta matriz de transmisión también se conoce como matriz ABCD y permite, como se verá, el cálculo de la función de transferencia de una red PLC. Para ejemplificar el procedimiento de obtención de la función de transferencia, a continuación se mostrará el cálculo para una red simple con topología en T, a partir de lo cual se puede generalizar el procedimiento.

2.3.1. La matriz de transmisión y la función de transferencia

La representación de un circuito bipuerto con matrices de transmisión es muy conveniente para el cálculo de la función de transferencia. Es bien conocido que si se tiene una secuencia (arreglo que también se denomina como *en cascada*) de dos circuitos bipuerto, como se muestra en la Figura 2.6, la matriz equivalente es el resultado de la multiplicación de las dos matrices de transmisión de

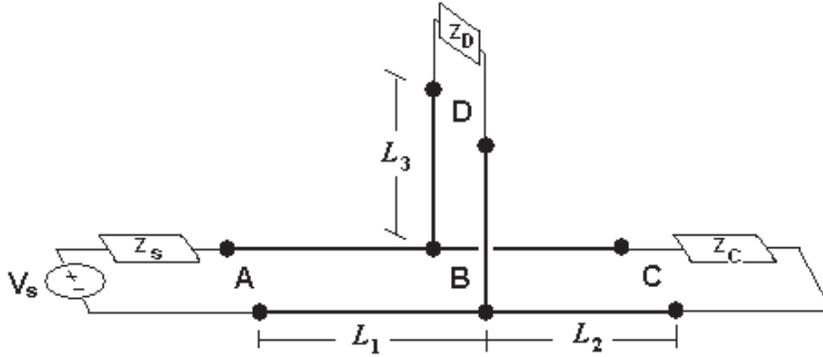


Figura 2.7: Línea de transmisión para una red con topología de nodo en T

los circuitos bipuerto individuales, es decir,

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix} \begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix} \quad (2.11)$$

Ahora, si en la Figura 2.5 hacemos que V e I sean, respectivamente, valores de voltaje y corriente en función de la frecuencia, entonces obtenemos la función de transferencia del circuito mediante la siguiente expresión $H = \frac{V_2}{V_s} = \frac{V_L}{V_s}$.

Empleando (2.9) y el hecho de que $I_2 = \frac{V_2}{Z_L}$, nos lleva a la siguiente expresión para la función de transferencia:

$$H = \frac{V_L}{V_s} = \frac{Z_L}{AZ_L + B + CZ_L Z_s + DZ_s} \quad (2.12)$$

o incluso,

$$H' = \frac{V_L}{V_1} = \frac{Z_L}{AZ_L + B} \quad (2.13)$$

si únicamente se considera la señal disponible en la entrada de la red bipuerto. Así, la función de transferencia del circuito representado en la Figura 2.5 está dada por (2.12) mientras que los parámetros de la línea de transmisión se calculan con (2.10).

2.3.2. Función de transferencia de una red con topología de nodo en T

La función de transferencia de redes con derivaciones puede ser obtenida directamente mediante la expresión (2.11) si se calculan los parámetros de la matriz de transmisión equivalente. Considere, por ejemplo, el circuito de la Figura 2.7, el cual muestra una red con topología en T.

Si se reemplaza la derivación, que parte del punto B y termina en la carga conectada al punto D, con una impedancia equivalente Z_{eq} , el circuito puede simplificarse como se muestra en la Figura 2.8. Si la impedancia de una línea es $Z = V_1/I_1$, también se expresa

$$Z = \frac{AZ_L + B}{CZ_L + D} \quad (2.14)$$

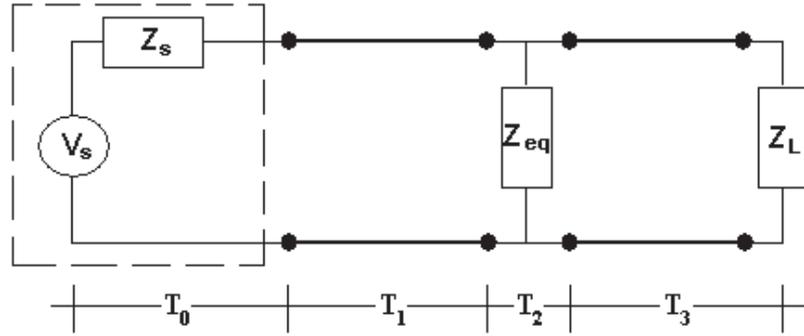


Figura 2.8: Circuito equivalente para la Figura 2.7

Entonces, la impedancia equivalente Z_{eq} de la derivación con longitud L_3 , que resulta del paralelo de la impedancia Z_3 con la impedancia Z_D , es

$$Z_{eq} = \frac{Z_D + Z_3 \tanh(\gamma_3 L_3)}{2 + \frac{Z_D}{Z_3} \tanh(\gamma_3 L_3) + \frac{Z_3}{Z_D} \tanh(\gamma_3 L_3)} \quad (2.15)$$

Valor que, después de una simplificación por análisis dimensional, puede aproximarse como sigue [56],

$$Z_{eq} \approx Z_3 \frac{Z_D + Z_3 \tanh(\gamma_3 L_3)}{Z_3 + Z_D \tanh(\gamma_3 L_3)} \quad (2.16)$$

Así, el circuito puede ser dividido en cuatro sub-circuitos bipuerto en cascada. Se puede calcular la matriz de transmisión para cada sub-circuito ($\mathbf{T}_i, i = 0, 1, 2, 3$) y, después, obtener la matriz equivalente total \mathbf{T} ,

$$\mathbf{T} = \prod_0^3 \mathbf{T}_i \quad (2.17)$$

El cálculo de las matrices de los sub-circuitos es relativamente simple y, para nuestra red simple con nodo en T, se puede resumir como sigue,

$$\mathbf{T}_0 = \begin{bmatrix} 1 & Z_s \\ 0 & 1 \end{bmatrix} \quad (2.18)$$

$$\mathbf{T}_1 = \begin{bmatrix} \cosh(\gamma_1 L_1) & Z_1 \sinh(\gamma_1 L_1) \\ \frac{1}{Z_1} \sinh(\gamma_1 L_1) & \cosh(\gamma_1 L_1) \end{bmatrix} \quad (2.19)$$

$$\mathbf{T}_2 = \begin{bmatrix} 1 & 0 \\ \frac{1}{Z_{eq}} & 1 \end{bmatrix} \quad (2.20)$$

$$\mathbf{T}_3 = \begin{bmatrix} \cosh(\gamma_2 L_2) & Z_2 \sinh(\gamma_2 L_2) \\ \frac{1}{Z_2} \sinh(\gamma_2 L_2) & \cosh(\gamma_2 L_2) \end{bmatrix} \quad (2.21)$$

donde \underline{Z}_1 , $\underline{\gamma}_1$, \underline{Z}_2 y $\underline{\gamma}_2$ son las impedancias características y las constantes de propagación para las líneas de transmisión de longitud L_1 (sub-circuito 1) y L_2 (sub-circuito 3), respectivamente.

Sustituyendo (2.18)-(2.21) en (2.17) se obtienen los parámetros de la línea de transmisión para la matriz equivalente que corresponde al circuito de la Figura 2.7. Haciendo

$$\begin{aligned}\kappa_1 &= \cosh(\underline{\gamma}_1 L_1) & \varsigma_1 &= \sinh(\underline{\gamma}_1 L_1) \\ \kappa_2 &= \cosh(\underline{\gamma}_2 L_2) & \varsigma_2 &= \sinh(\underline{\gamma}_2 L_2)\end{aligned}$$

tenemos que

$$A = \kappa_2 \alpha + \frac{\varsigma_2}{\underline{Z}_2} \beta \quad (2.22)$$

$$B = \underline{Z}_2 \varsigma_2 \alpha + \kappa_2 \beta \quad (2.23)$$

$$C = \kappa_2 \delta + \frac{\varsigma_2}{\underline{Z}_2} \kappa_1 \quad (2.24)$$

$$D = \underline{Z}_2 \varsigma_2 \delta + \kappa_2 \kappa_1 \quad (2.25)$$

donde

$$\alpha = \kappa_1 + \frac{Z_s}{\underline{Z}_1} \varsigma_1 + \frac{\underline{Z}_1 \varsigma_1 + Z_s \kappa_1}{Z_{eq}} \quad (2.26)$$

$$\beta = \underline{Z}_1 \varsigma_1 + Z_s \kappa_1 \quad (2.27)$$

$$\delta = \frac{\underline{Z}_1 \kappa_1 + Z_{eq} \varsigma_1}{\underline{Z}_1 Z_{eq}} \quad (2.28)$$

Finalmente, se obtiene la función de transferencia usando la ecuación (2.12). Para canales con más derivaciones, el procedimiento es esencialmente el mismo pero la expresión resultante para cada parámetro de la línea de transmisión se va complicando.

2.4. Modelo de ruido para el canal PLC

La red eléctrica presenta una mezcla de ruido que no puede clasificarse como ruido blanco. Además del ruido de fondo con espectro de potencia decreciente en la frecuencia (ruido “cromático”), el canal es afectado por interferencias de banda angosta y ruido impulsivo. De todos estos, el ruido más nocivo es el ruido impulsivo y es la fuente más importante de errores debido a que su densidad de potencia espectral puede superar por más de 40 dB al ruido de fondo [9].

La principal fuente de ruido impulsivo se constituye por los transitorios debidos a conmutaciones en la red. Los impulsos ocurren en forma aleatoria y pueden tener duraciones desde microsegundos hasta algunos milisegundos. El modelo de ruido aditivo blanco de clase A (AWCN, por sus siglas en inglés) es considerado en la literatura especializada como el adecuado para representar el ruido impulsivo en un canal PLC [61, 118, 17]. El modelo AWCN es también conocido como ruido de clase A de Middleton y es una combinación de ruido aditivo blanco más ruido impulsivo (ver Figura 2.9). Entonces, la función de densidad de probabilidad (pdf), para una variable

aleatoria x , es la sumatoria infinita de Gaussianas ponderadas con pesos decrecientes conforme aumenta la varianza:

$$p(x) = \frac{1}{2\pi} \sum_{m=0}^{\infty} \frac{1}{\sigma_m^2} \alpha_m \exp\left(-\frac{x^2}{2\sigma_m^2}\right) \quad (2.29)$$

con

$$\alpha_m = e^{-A} \frac{A^m}{m!} \quad (2.30)$$

$$\sigma_m^2 = \sigma^2 \frac{\frac{m}{A} + \Gamma}{1 + \Gamma} \quad (2.31)$$

$$\sigma^2 = \sigma_g^2 + \sigma_I^2 \quad (2.32)$$

$$\Gamma = \frac{\sigma_g^2}{\sigma_I^2} \quad (2.33)$$

donde A representa el tiempo promedio entre impulsos arribados, σ_I^2 es la potencia promedio del ruido impulsivo y σ_g^2 es la potencia promedio de la componente de ruido Gaussiano. A Γ también se le conoce como relación de potencias entre el ruido Gaussiano y el impulsivo. En este modelo, una muestra de ruido $n(m)$ se puede expresar como

$$n(m) = x_G(m) + y(m)\sqrt{k_m} \quad (2.34)$$

donde x_G es una variable aleatoria Gaussiana, con media cero y varianza σ_g^2 , que representa al ruido de fondo; $y(m)$ es otra variable aleatoria Gaussiana con media cero y varianza $\frac{\sigma_I^2}{A}$, que representa a la potencia del ruido impulsivo; y k_m es una variable aleatoria de Poisson [77], que representa la densidad de pulsos arribados, cuya pdf se encuentra caracterizada por A . Todas estas variables son estadísticamente independientes (ver Apéndice E.3.4).

Es muy importante hacer énfasis en dos características del ruido impulsivo:

- Gran intensidad y
- corta duración.

Lo anterior en relación a la magnitud y duración de un símbolo OFDM. En la Figura 2.10 se puede observar el ruido impulsivo generado durante la simulación de un símbolo OFDM con su intervalo de guardia. La característica de corta duración implica un espectro de frecuencia amplio, como se puede observar en la Figura 2.11 donde se presenta el espectro calculado para la señal de ruido de la Figura 2.10.

2.5. Canal ciclo-estacionario

Cañete y sus colaboradores [23, 38, 25, 125] demostraron que los canales PLC muestran variaciones a escala temporal del ciclo de la línea, debido a que, a frecuencias elevadas, los parámetros

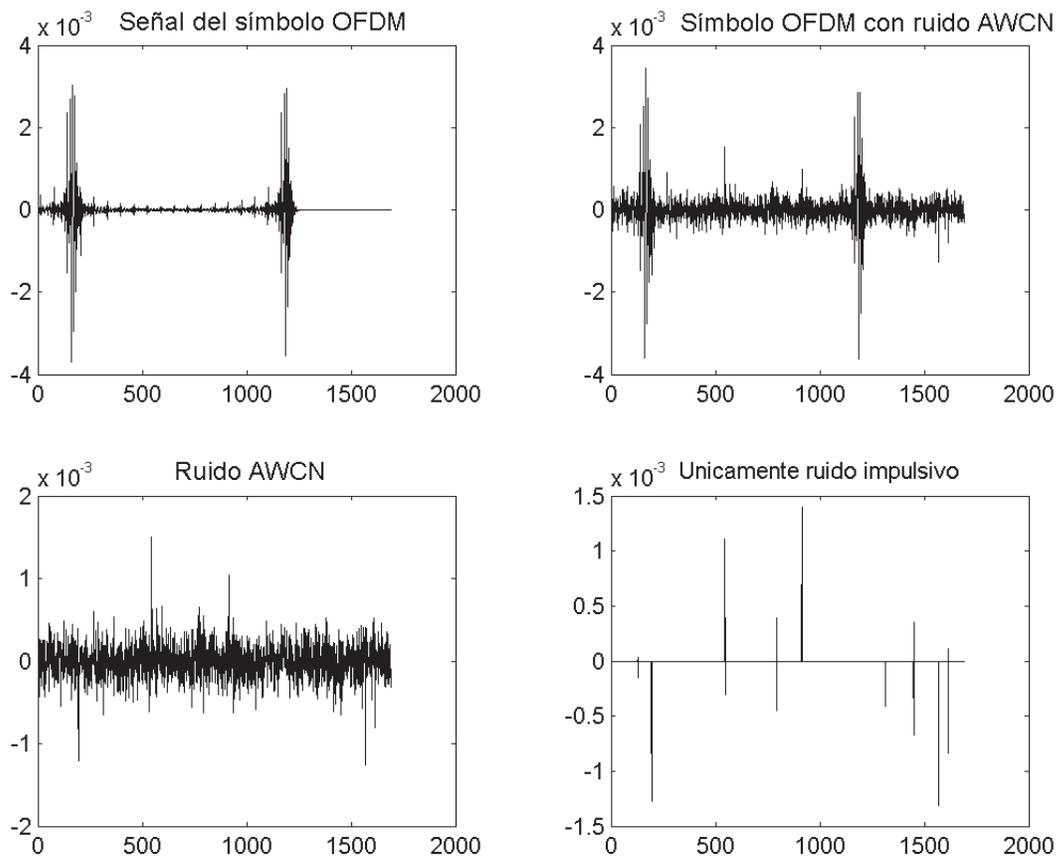


Figura 2.9: Símbolo OFDM perturbado por mezcla de ruido Gaussiano más ruido impulsivo

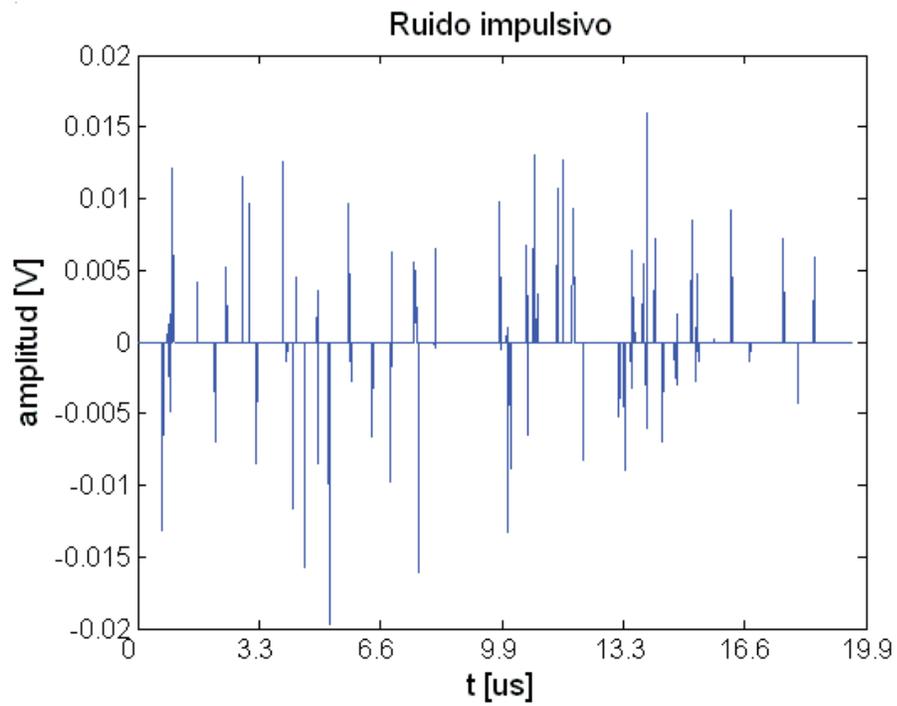


Figura 2.10: Ruido impulsivo adicionado a un símbolo OFDM con una densidad de 6000 impulsos por ciclo de potencia (16.6 ms)

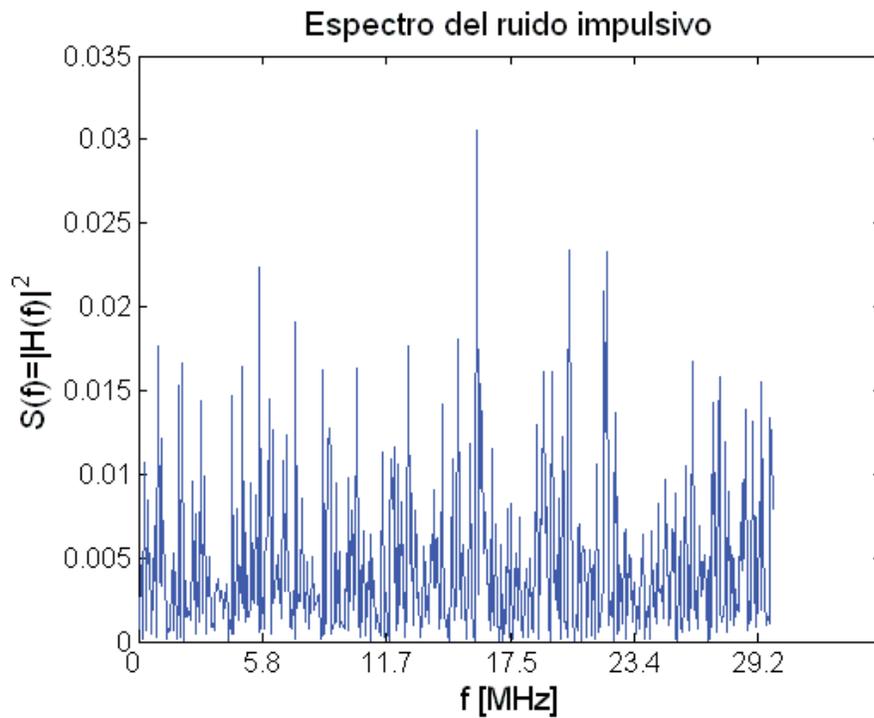


Figura 2.11: Espectro del ruido impulsivo de la Figura 2.10

de las cargas conectadas dependen de la amplitud instantánea de la señal de potencia, todo lo cual representa un comportamiento no lineal.

Como se explica en [23], las cargas no lineales producen variaciones, a pequeña escala temporal, sincronizadas con la línea. Estas variaciones del canal son graduales, desde la perspectiva de las señales pequeñas, y el canal PLC puede ser visto como un sistema lineal a la vez que periódicamente variante en el tiempo (LPTV, por sus siglas en inglés).

En este trabajo se propone un modelo de aproximación bidimensional, por partes (*piece wise*), para simular el comportamiento no lineal de las impedancias de carga. Aquí, nuestro propósito no es generar un modelo representativo para una carga convencional, sino tomar un caso extremo que muestre claramente los efectos de un canal ciclo-estacionario. Por ejemplo, el siguiente modelo fue derivado en forma empírica, a fin de obtener comportamientos similares a los reportados en [25] para el caso de cargas tales como rasuradoras eléctricas y lámparas de baja potencia:

$$Z(t, f) = r(t, f) + \alpha * k_o \quad (2.35)$$

con

$$r(t, f) = \begin{cases} (g(t) + \beta * (1 + r_3)) * \exp(-k_{f1} * f) & \text{Si } (0 < t) \text{ y } (t \leq \tau) \\ g(t) * (1 + r_5) * \text{gaussian}(f, w_p, \sigma) * \\ (\frac{1}{2}|\text{sinc}(k_{f2} * (f - w_p))| + |\cos(k_{f3} * (f - w_p))| + 3) & \text{Si } (\tau < t) \text{ y } (t \leq (\frac{T}{2} - \tau)) \\ (g(t) + \beta * (1 + r_4)) * \exp(-k_{f1} * f) & \text{Si } ((\frac{T}{2} - \tau) < t) \text{ y } (t \leq (\frac{T}{2} + \tau)) \\ g(t) * (1 + r_6) * \text{gaussian}(f, w_p, \sigma) * \\ (\frac{1}{2}|\text{sinc}(k_{f2} * (f - w_p))| + |\cos(k_{f3} * (f - w_p))| + 3) & \text{Si } ((\frac{T}{2} + \tau) < t) \text{ y } (t \leq (T - \tau)) \\ (g(t) + \beta * (1 + r_3)) * \exp(-k_{f1} * f) & \text{Si } (T - \tau) < t \text{ y } t \leq T \end{cases}$$

y

$$g(t) = \alpha(1 + r_1)\text{sinc}(k_t(t - \frac{1}{4}T)) + \alpha(1 + r_2)\text{sinc}(k_t(t - \frac{3}{4}T)) + \text{alfa} * k_o \quad (2.36)$$

$$\tau = (1 + r_7)\frac{T}{8} \quad (2.37)$$

$$\text{gaussian}(y, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(\frac{-1}{2\sigma^2}(y - \mu)^2) \quad (2.38)$$

$$\text{sinc}(x) = \begin{cases} 1 & \text{Si } x = 0 \\ \frac{\sin(\pi x)}{\pi x} & \text{en otro caso.} \end{cases} \quad (2.39)$$

donde t es el tiempo en milisegundos; f es la frecuencia (rad) normalizada a un máximo en 2π ; α es la impedancia pico máxima (Ohms) con $f = w_p$; w_p es la frecuencia con un máximo local, también normalizada respecto a 2π ; σ es la desviación estándar para la envolvente Gaussiana en el eje de la frecuencia, ponderada por el máximo local en w_p ; β es impedancia máxima (Ohms) con $f = 0$; T el periodo del ciclo de la línea en ms; k_o es un nivel mínimo (*offset*) para evitar valores negativos de impedancia. r_1 a r_7 son valores, entre -0.5 y 0.5, para generar variaciones entre semiciclos. A saber: r_1 y r_2 para la amplitud de las componentes $\text{sinc}()$ en el dominio del tiempo; r_3 y r_4 son los

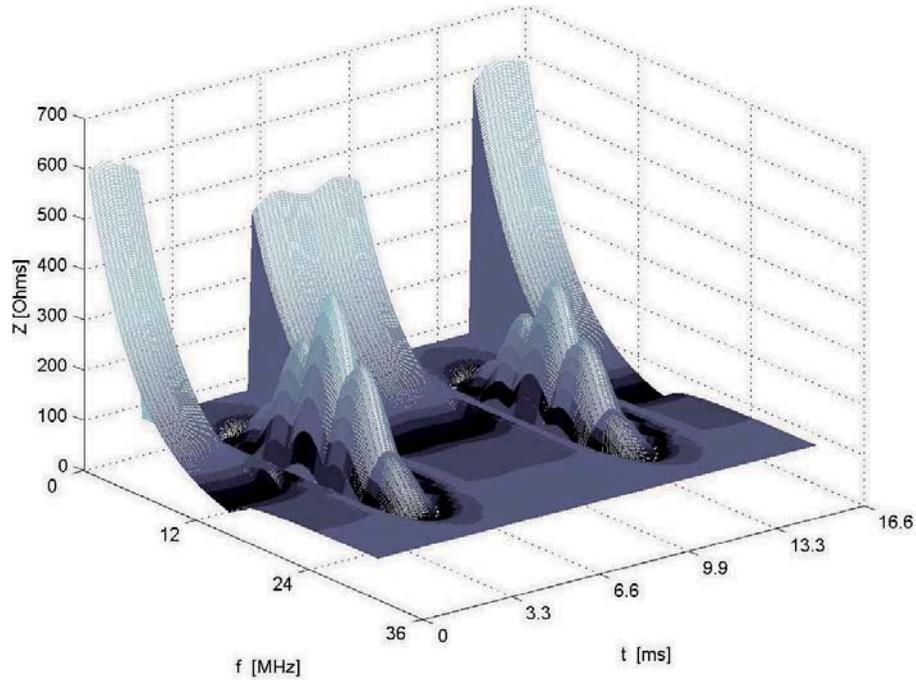


Figura 2.12: Impedancia variante, en el tiempo y la frecuencia, de la carga simulada

valores máximos de la impedancia con envolvente exponencial en frecuencia, durante el primer y segundo semiciclo; r_5 y r_6 son los valores máximos de la impedancia con envolvente Gaussiana en frecuencia, durante el primer y segundo semiciclo; r_7 es la duración de las ranuras de tiempo para conmutar de una envolvente exponencial a una Gaussiana. Finalmente, tenemos a las constantes de escalamiento k_t , k_{f1} , k_{f2} y k_{f3} . k_t es para la envolvente de la función $\text{sinc}()$ en el dominio temporal; k_{f1} es para la envolvente exponencial en el dominio en frecuencia; k_{f2} es para la envolvente $\text{sinc}()$ en el dominio de la frecuencia y k_{f3} es para la envolvente $\text{cos}()$ en el dominio de la frecuencia.

Por ejemplo, el comportamiento observado en la Figura 2.12 se obtuvo con los siguientes valores (ver `ImpCicloest04()` en Apéndice E.3.3): $\alpha = 250$, $w_p = 3$, $\sigma = 1$, $\beta = 500$, $T = 16,6$, $k_o = 0,05$, $r_1 = -0,5$, $r_2 = -0,6$, $r_3 = 0,15$, $r_4 = 0,6$, $r_5 = -0,15$, $r_6 = 0,5$, $r_7 = 0$, $k_t = 0,75$, $k_{f1} = 1$, $k_{f2} = 2$, $k_{f3} = 4$.

2.6. Simulación del canal PLC

A lo largo de este trabajo se realizarán simulaciones de un canal PLC con la topología de la red de la Figura 2.7 con $L_1 = L_2 = 15\text{m}$, $L_3 = 9,75\text{m}$, y los siguientes valores por unidad: $r = 0,0084\Omega/\text{m}$, $l = 0,1\mu\text{H}/\text{m}$, $c = 3,5\text{nF}/\text{m}$ and $g = 3\text{pS}/\text{m}$. Se eligió esta topología por ser la más simple que puede existir con una derivación, al interior de un domicilio de dimensiones regulares. Esta topología también fue alambrada en el laboratorio para realizar mediciones de la respuesta en frecuencia y una estimación aproximada de los parámetros por unidad de la misma. La información

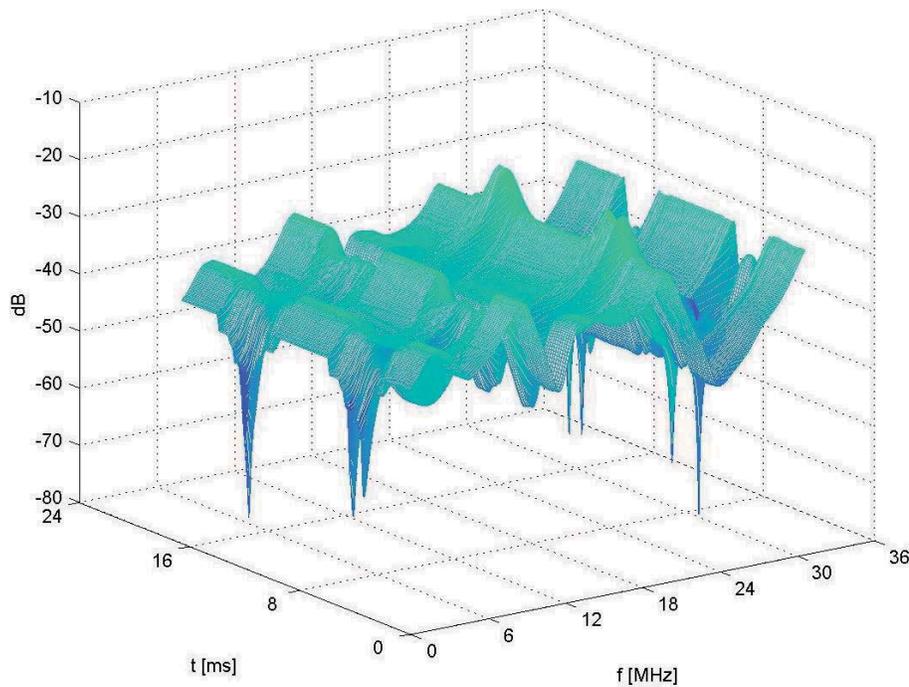


Figura 2.13: Respuesta del canal simulado en el tiempo y la frecuencia

obtenida de estas mediciones en el laboratorio se empleó fundamentalmente para validar el modelo del canal y el comportamiento simulado en Matlab.

En el simulador (ver Apéndice E.3.3), el dispositivo transmisor fue conectado al punto A, mientras que el receptor se conectó a C, ambos con impedancias acopladas de 50Ω . Finalmente, en el punto D se conectó la carga con la impedancia ciclo-estacionaria $Z(t, f)$ que aparece en la Figura 2.12. La función de transferencia resultante, variante en el tiempo y selectiva en frecuencia, se muestra en la Figura 2.13. El rango de las respuestas en frecuencia se resalta en 2.14, mostrando las posibles funciones de transferencia durante el ciclo de 60Hz de la línea si se conecta la carga con impedancia ciclo-estacionaria $Z(t, f)$.

El entorno de simulación fue programado en Matlab (ver Apéndice E) y permite habilitar y deshabilitar el comportamiento ciclo-estacionario. Cuando el comportamiento ciclo-estacionario es deshabilitado, sólo una de las posibles funciones de transferencia es empleada durante toda la simulación Monte Carlo. Por el contrario, en un escenario ciclo-estacionario la simulación garantiza que el canal es estacionario durante toda la transmisión de un mismo símbolo OFDM, pero cambia en forma continua entre dos símbolos OFDM sucesivos. Así, para cada símbolo OFDM se genera una nueva respuesta del canal, de tal manera que, después de un periodo completo de 60Hz, la respuesta del canal sufre todas las posibles variaciones debidas a la carga no lineal $Z(t, f)$. Finalmente, el ruido de fondo se simula para diferentes valores de la relación señal a ruido (SNR, por *Signal to Noise Ratio*) con una densidad para el ruido impulsivo que se mantiene constante durante toda la simulación (sin importar si se trata de un escenario estacionario o ciclo-estacionario).

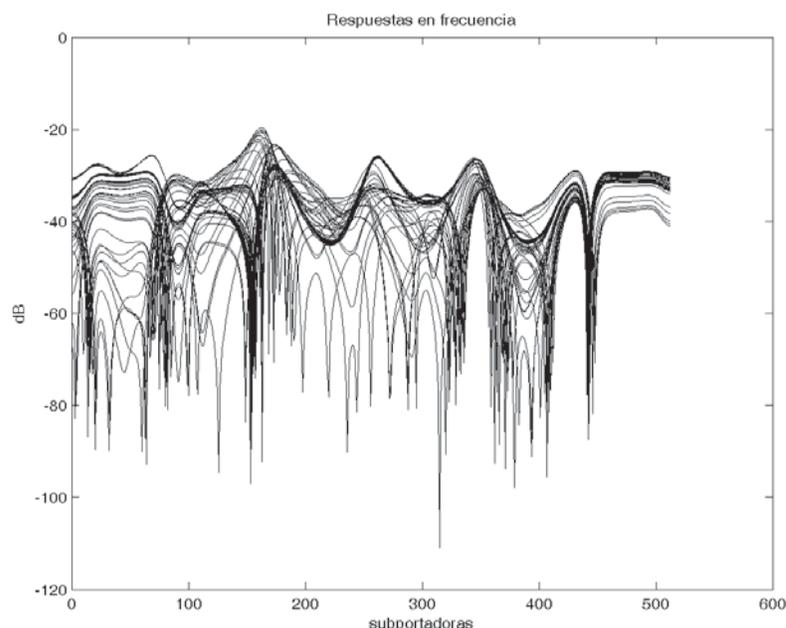


Figura 2.14: Respuestas en frecuencia del canal simulado

2.7. Simulación de transreceptor OFDM

Dentro de la tecnología PLC, a fin de minimizar los efectos de la atenuación y del ruido, maximizando al mismo tiempo el aprovechamiento del canal, se emplean técnicas avanzadas de procesamiento de señales y comunicaciones digitales, tales como el multiplexado por repartición en frecuencias ortogonales (Orthogonal Frequency-Division Multiplexing, OFDM) y la técnica de asignación de bits conocida como esquema multi-tono discreto (Discrete Multi-Tone, DMT), que han sido adoptadas como técnicas básicas en el ámbito PLC [9].

En lo que resta de este capítulo, se presenta una breve exposición sobre los principios de la técnica OFDM, útil para entender la operación del transreceptor OFDM que se simula en este trabajo (ver Apéndices E.3.2 y E.3.5), sin embargo, el lector puede remitirse a las referencias para tener más detalles.

2.7.1. Principio de operación de la técnica OFDM

El principio de modulación multiportadora descansa en la idea de paralelizar la información transmitida en el dominio de la frecuencia (FDM- *Frequency Division Multiplexing*). Los datos originales, de mayor velocidad $\frac{1}{T_d}$, son repartidos entre varios módulos que los transmiten simultáneamente, aunque a menor velocidad, sobre igual cantidad de subcanales espectrales centrados en ciertas frecuencias subportadoras.

Si N es el número de subportadoras utilizadas, los símbolos transmitidos por cada una de ellas tienen una duración $T_s = NT_d$. En el dominio temporal, la señal obtenida se descompone en símbolos de duración T_s que resultan de la superposición de N señales senoidales a diferentes frecuencias. Aumentando suficientemente el número de subportadoras, la duración de los símbolos puede hacerse mayor al retraso en la dispersión de la respuesta al impulso, lo que permite minimizar los efectos de interferencia entre símbolos (ISI- *Intersymbol Interferente*). De esta manera, las distorsiones introducidas en la señal por el canal son limitadas en el dominio en frecuencia ya que cada sub-banda se hace suficientemente estrecha para ser afectada únicamente por una respuesta local del canal.

Sin embargo, hay que mencionar que este sistema sólo puede funcionar apropiadamente si se impide la aparición de toda interferencia entre portadoras (ICI - *Intercarrier Interferente*). Una alternativa eficiente consiste en hacer el multiplexado en frecuencia, pero adecuando los espectros de las subportadoras para asegurar que éstas constituyen una base de funciones ortogonales. Es entonces que aparece la denominación OFDM (*Orthogonal FDM*) [31].

Las propiedades de ausencia de ISI en el dominio temporal y de ICI en el dominio en frecuencia, que constituyen una extensión del primer criterio de Nyquist, condujeron a la búsqueda de funciones base con el fin de transmitir los símbolos cumpliendo con estos requisitos, en lo que se conoce como formateo de pulso (*pulse shaping*). Para esto, la función de pulso rectangular $\Pi(t)$ es la más empleada por su sencillez. En adelante, se asumirá que la función de pulso rectangular será utilizada como función de formateo de pulso:

$$\Pi(t) = \begin{cases} 1, & |t| < \frac{1}{2}T_s \\ \frac{1}{2}, & t = \pm\frac{1}{2}T_s \\ 0, & \text{otro} \end{cases} \quad (2.40)$$

lo cual consiste en un ventaneo rectangular de los símbolos con una duración T_s y, para cada una de las subportadoras de la señal, genera un espectro con forma similar al de una función *sinc()* [120].

En principio, el espaciamiento mínimo entre dos portadoras adyacentes que permite conservar la ortogonalidad entre éstas es $d_f = \frac{1}{T_s}$. Entonces al paralelizar el flujo se espera tener una banda aproximadamente rectangular de ancho ideal $(B) = \frac{N}{T_s} = \frac{1}{T_d}$. De manera general, la señal OFDM implica un ancho de banda real $B = \frac{(N+1)}{T_s}$ con una profunda disminución del espectro en sus bordes pero sin desaparecer la banda infinita. Por ello es necesario limitar la contribución fuera de banda de los lóbulos secundarios mediante un filtro. Según la severidad del filtro, tal operación introduce una distorsión más o menos importante para las subportadoras localizadas en los bordes del espectro, que en el dominio temporal resulta en la aparición de ISI. Mientras estas perturbaciones sean menores a las engendradas por el canal de propagación, una solución es no transmitir nada sobre las subportadoras de los costados del espectro, llamadas subportadoras de guardia, obteniendo exactamente el espectro deseado pero con el costo de una ligera pérdida en la eficiencia espectral.

De manera más general, es posible asignar a cada subportadoras los símbolos provenientes de

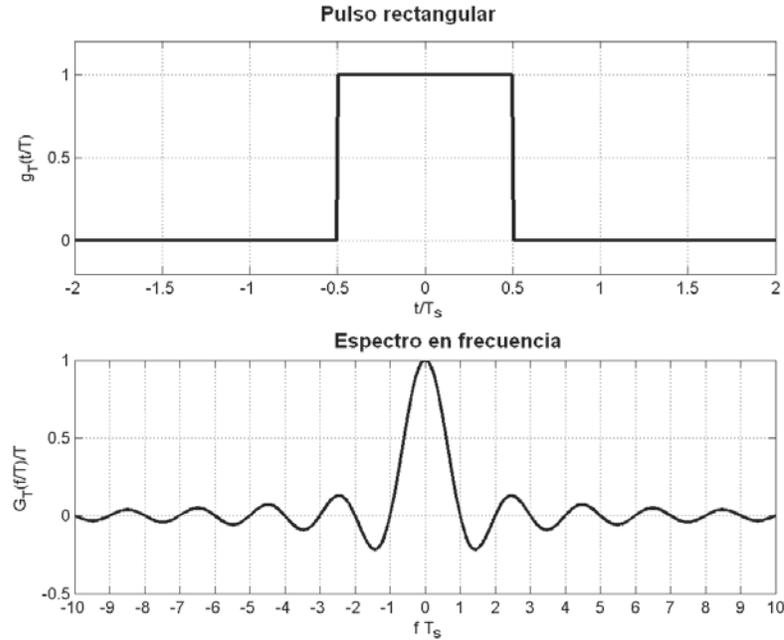


Figura 2.15: La función pulso rectangular $\Pi(t)$ y su espectro en frecuencia

diferentes esquemas de modulación y transmitirlos con potencias distintas. Esto representa la ventaja de poder adaptar la señal emitida al canal de propagación, siempre que se tenga el conocimiento del estado del canal en el transmisor. Este principio, denominado de modulación adaptativa, es el principal beneficiario del realizar una buena estimación de canal, que es justamente el propósito de nuestro trabajo de investigación.

2.7.2. La señal transmitida con OFDM

La señal OFDM se constituye de N subportadoras de frecuencia $f_k = f_0 + k\Delta_f, k \in [0 : N - 1]$, utilizadas para la transmisión paralela de N símbolos, denotados x_k . Los símbolos x_k son elementos complejos que toman su valor de un alfabeto conforme a la modulación digital empleada (por ejemplo, BPSK o QAM). Entonces, al utilizar la función rectangular $\Pi(t)$ como función de formateo de pulso, la cual cumple con los criterios de ortogonalidad, aseguramos que $\Delta_f = 1/T_s$. Entonces, la expresión normalizada de la señal OFDM generada durante el intervalo $[0 : T_s]$ está dada por

$$s(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \Re \left\{ x_k \Pi(t) e^{j2\pi(f_0 + \frac{k}{T_s})t} \right\} \quad (2.41)$$

Siendo f_c la frecuencia central de la banda B para la señal, $f_c = f_0 + \frac{N}{2T_s}$, tenemos

$$s(t) = \Re \left\{ \Pi(t) e^{j2\pi f_c t} \sum_{k=0}^{N-1} \frac{x_k}{\sqrt{N}} e^{j2\pi(k - \frac{N}{2})\frac{t}{T_s}} \right\} \quad (2.42)$$

que puede expresarse

$$s(t) = \Re\{\Pi(t)e^{j2\pi f_c t \tilde{s}(t)}\} \quad (2.43)$$

donde $\tilde{s}(t)$ es la parte compleja de la señal $s(t)$ antes del ventaneo por la función de pulso rectangular. Estando limitado el espectro al intervalo $[-\frac{N}{2T_s} : \frac{N}{2T_s}]$, la señal $\tilde{s}(t)$ puede ser muestreada a una frecuencia $f_m = N/T_s$ sin que haya superposición espectral. Las muestras obtenidas se expresan

$$\begin{aligned} s_n &= \Re\left\{ \Pi\left(n\frac{T_s}{N}\right) e^{j2\pi f_c n \frac{T_s}{N}} \sum_{k=0}^{N-1} \frac{x_k}{\sqrt{N}} e^{j2\pi\left(k-\frac{N}{2}\right)\frac{n}{N}} \right\} \\ &= \Re\left\{ (-1)^n \Pi\left(n\frac{T_s}{N}\right) e^{j2\pi f_c n \frac{T_s}{N}} \underbrace{\sum_{k=0}^{N-1} \frac{x_k}{\sqrt{N}} e^{j2\pi\frac{kn}{N}}}_{\text{DFT}^{-1}} \right\} \end{aligned} \quad (2.44)$$

Lo cual pone en evidencia que la señal puede ser generada fácilmente utilizando una transformada inversa de Fourier discreta (IDFT, *Inverse Discrete Fourier Transform*). Complementariamente, en la recepción se realiza una transformada de Fourier discreta (DFT) sobre las muestras recibidas, para recuperar los símbolos transmitidos. Una escritura matricial de esta señal conduce finalmente a

$$\mathbf{s} = \mathbf{F}^{-1}\mathbf{x} \quad (2.45)$$

donde $\mathbf{s} = [s_0 \ s_1 \ \cdots \ s_{N-1}]^T$ es el vector de muestras temporales del símbolo OFDM, $\mathbf{x} = [x_0 \ x_1 \ \cdots \ x_{N-1}]^T$ es el vector de símbolos de modulación (por ejemplo, símbolos BPSK o QAM) transmitidos sobre las subportadoras y, finalmente, \mathbf{F} es la matriz de Fourier de dimensiones $N \times N$ definida por

$$\mathbf{F} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{-j\frac{2\pi}{N}} & \ddots & e^{-j\frac{2\pi(N-1)}{N}} \\ \vdots & & & \vdots \\ 1 & e^{-j\frac{2\pi(N-1)}{N}} & \cdots & e^{-j\frac{2\pi(N-1)^2}{N}} \end{bmatrix} \quad (2.46)$$

Recordemos que esta matriz es unitaria y su inversa es igual a su transpuesta Hermitiana (conjugada) $\mathbf{F}^{-1} = \mathbf{F}^H$.

2.7.3. Intervalo de guardia

A pesar de que la ISI podría ser limitada aumentando indefinidamente la duración T_s de los símbolos OFDM, en la práctica esto no es posible debido a que el tiempo de coherencia del canal es finito. Una solución ingeniosa consiste en insertar un intervalo de guardia (GI, por sus siglas en inglés). El intervalo de guardia es un lapso del tiempo durante el cual no es transmitido ningún dato útil y, al ser insertado como prefijo de cada símbolo OFDM, sirve para absorber la ISI, siempre que su duración T_g sea mayor o igual al retraso máximo de la dispersión τ_{max} de la respuesta al impulso. Después de su inserción, el espaciamiento entre las subportadoras se mantiene en $1/T_s$, a pesar de que la duración de los símbolos OFDM se incrementa a $T_s + T_g$, arrastrando con ello cierta pérdida de ortogonalidad entre las subportadoras y una menor eficiencia espectral dada por

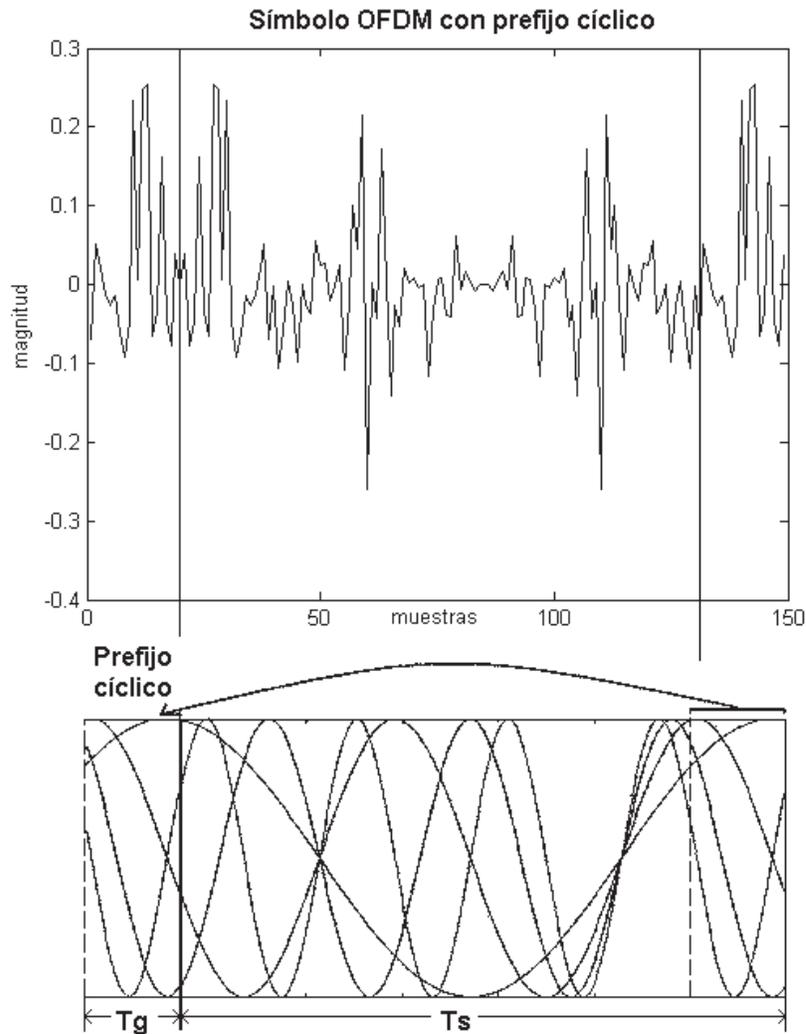


Figura 2.16: Símbolo OFDM con intervalo de guardia

$T_s/(T_s + T_g)$. A fin de recuperar la ortogonalidad en la recepción y, con ello, recuperar los símbolos transmitidos sin afectación de ICI alguna, durante la ventana rectangular de duración T_s , cada una de las señales senoidales que componen el símbolo OFDM debe tener un número entero de períodos, lo cual se logra si el intervalo de guardia es una copia de la parte final del mismo símbolo OFDM (ver Figura 2.16), que entonces se conoce como prefijo cíclico (CP, por sus siglas en inglés).

2.7.4. Especificación para el transreceptor OFDM simulado

La especificación base del sistema OFDM simulado (ver Apéndices E.3.2 y E.3.5), en lo referente a la estructura de la trama de datos y el procedimiento de sincronización, es la del transreceptor lineal asistido por pilotos descrita en [40]. Se trata de un transreceptor básico con un sistema DMT,

Cuadro 2.1: Parámetros del transreceptor convencional simulado, referidos a la señal analítica

Frecuencia del sistema – $f_s = B$	30 MHz
Duración de la muestra – $t_s = 1/f_s$	33,3 ns
Subportadoras – N	512
Ancho de las sub-bandas – $\Delta_f = f_s/N$	58.59 kHz
Duración del GI – D	75 muestras
Tiempo del GI – $T_g = D \times t_s$	$2,5\mu$ s
Duración del símbolo OFDM – $T_s = 1/\Delta_f = N \times t_s$ sin GI	17μ s
Duración total del símbolo OFDM – $T_s + T_g$	19.5μ s
Modulación de portadora	QPSK
FFT	1024

mono-usuario, con 512 subportadoras, banda total de 30Mhz, símbolos de modulación QPKS y símbolos OFDM con *intervalo de guardia* (GI). El transreceptor simulado envía los datos de información repartidos en tramas de 256 símbolos OFDM, donde el primer símbolo de cualquier trama de un enlace descendente es siempre un piloto de entrenamiento, empleado tanto para sincronización fina del sistema como para estimación del canal. Las estrategias para la estimación del canal disponibles en el entorno de simulación programado en Matlab incluyen tanto a la estrategia asistida por pilotos (para el caso estacionario) como la estrategia regida por el proceso de decisión (para el caso ciclo-estacionario). La tabla 2.1 resume las características del transreceptor simulado. Observe que los valores son especificados para la señal analítica, ya que la frecuencia real de muestreo debe ser 60MHz para cumplir el principio de Nyquist. Finalmente, el número real de subportadoras empleadas está siempre determinado por la atenuación del canal estimada en la trama anterior. El lector puede remitirse a [85] para tener un panorama general del sistema de referencia o bien ir a la propuesta original en [40] para tener más detalles.

En la literatura se ha reportado que los sistemas PLC comerciales han alcanzado velocidades de hasta 200 Mbps con un ancho de banda de 30 MHz [1, 10]. Esto es perfectamente posible con un sistema OFDM debido al multiplexado en frecuencia. Por ejemplo, para nuestro sistema con 512 subportadoras y un ancho de bada de 30MHz, hablando en términos de la señal analítica y sin considerar al intervalo de guardia, resulta que se toma una muestra cada 33.3 ns (ver Cuadro 2.1) y, por lo tanto, las 512 muestras que corresponden a un símbolo OFDM consume un total de 17μ s. Un símbolo OFDM cada 17μ s implica un rendimiento aproximado de 58823 símbolos OFDM por segundo, cada uno con codificación de 512 símbolos de modulación. Entonces el rendimiento total depende del sistema de señalización empleado para los símbolos de modulación. Si la señalización de los símbolos de modulación se realiza con un esquema BPSK (dos posibles señales por cada símbolo de modulación codificados con un bit), entonces hablamos de que un símbolo OFDM puede transportar hasta 512 bits, lo que resulta en un rendimiento total aproximado de hasta $512 \times 1 \times 58823 \approx 30$ Mbps. Si cambiamos a un esquema de señalización QPSK (cuatro posibles señales por cada símbolo de modulación codificados con 2 bits) entonces el mismo sistema

puede tener un rendimiento total de hasta $512 \times 2 \times 58823 \approx 60$ Mbps. Finalmente, si empleamos un esquema 256-QAM (256 posibles señales por cada símbolo de modulación codificados con 8 bits) entonces se puede alcanzar un rendimiento total de hasta $512 \times 8 \times 58823 \approx 240$ Mbps.

Modelo y dimensiones de la señal transmitida en el sistema OFDM simulado

Las dimensiones del sistema propuesto son resumidas en la Tabla 2.1. Nótese que los valores dados en la tabla 2.1 caracterizan a la señal analítica ya que, en la práctica, se transmite una señal real muestreada a una frecuencia $2f_m = 60$ MHz. En la práctica, dicha señal real se obtiene mediante una IFFT (*Inverse Fast Fourier Transform*) de orden 4096 pero, a cambio, el transmisor emplea sólo un convertidor digital-analógico en vez de dos.

La señal real transmitida se puede generar directamente sobre la frecuencia intermedia de la banda, en este caso entre 0 y 30 MHz, mediante una IFFT de tamaño doble. La idea es obtener directamente una señal real, en la salida de modulador, aplicando una señal con simetría Hermítica en la entrada de la IFFT. Para reconstituir la simetría a partir de la señal analítica es necesario duplicar las muestras y arreglarlas correctamente.

En el transmisor las muestras de la señal OFDM obtenidas en salida de modulador se escriben

$$s_n = \sum_{k=0}^{N-1} \Re \left\{ x_k e^{\frac{j2\pi k}{N}} \right\} \quad (2.47)$$

con N es número de puntos de IFFT. Utilizando las propiedades de los números complejos, podemos reescribir que

$$s_n = \frac{1}{2} \left\{ \sum_{k=0}^{N-1} x_k e^{\frac{j2\pi k}{N}} + \sum_{k=0}^{N-1} x_k^* e^{-\frac{j2\pi k}{N}} \right\} \quad (2.48)$$

donde $(\cdot)^*$ representa la operación de conjugación compleja. Manipulando un poco esta expresión, tenemos

$$\begin{aligned} s_n &= \frac{1}{2} \left\{ \sum_{k=0}^{N-1} x_k e^{\frac{j2\pi k}{N}} + \sum_{k=N+1}^{2N} x_{2N-k}^* e^{-\frac{j2\pi k}{N}(2N-k)} \right\} \\ &= \frac{1}{2} \left\{ \sum_{k=0}^{2N-1} x'_k e^{\frac{j2\pi k}{N}} \right\} \end{aligned} \quad (2.49)$$

con las nuevas muestras x'_k definidas por

$$x'_k = \begin{cases} 0 & k = 0 \\ x_k & k \in [1 : N - 1] \\ 0 & k = N \\ x_{2N-k}^* & k \in [N + 1 : 2N - 1] \end{cases} \quad (2.50)$$

En la recepción, las operaciones son invertidas: la señal real es muestreada y alimentada a la entrada de una doble FFT. En la salida de la FFT, únicamente N muestras corresponden a las frecuencias positivas, es decir, a la señal analítica.

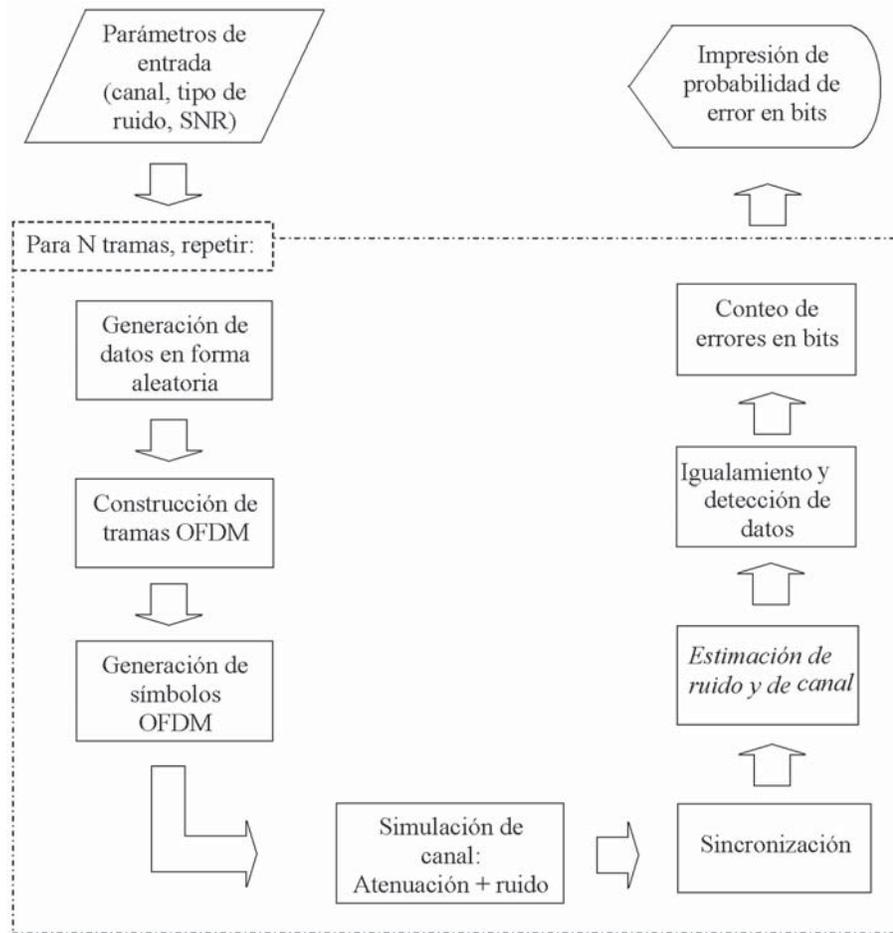


Figura 2.17: Simulación Monte Carlo de enlace OFDM para canal PLC

2.7.5. Metodología de simulación

La metodología para los experimentos tiene su fundamento en la repetición de simulaciones Monte Carlo (ver Apéndice E.3.1), como se puede observar en el esquema de la Figura 2.17. Durante cada experimento se mantiene constante la densidad de ruido impulsivo en las condiciones del escenario elegido (estacionario o ciclo-estacionario). Se realiza una simulación Monte Carlo para 1024 símbolos OFDM (cuatro tramas OFDM) por cada valor de la relación señal a ruido (SNR) a probar. Finalmente, cada experimento se repite diez veces a fin de promediar la probabilidad del error en los bits (BER, por *Bit Error Rate*) obtenida y obtener gráficas más representativas [121].

2.8. Conclusiones del capítulo

El modelo de canal más apropiado para nuestros propósitos de simulación es uno con enfoque determinístico, por ello en el código de simulación del entorno PLC se ha optado por el formalismo

de las matrices de transmisión para el canal de propagación. Así mismo, dado que en la literatura especializada se considera al modelo de ruido de clase A de Middleton como el más apropiado para representar a la mezcla de ruido Gaussiano con ruido impulsivo, conforme a las campañas de mediciones en redes PLC reales, aquí también se ha elegido a este modelo de ruido para propósitos de simulación. Finalmente, se realiza la aportación del modelado para una carga con comportamiento ciclo-estacionario a fin de incorporarla al entorno de simulación y tener la opción de habilitar escenarios con variaciones de corta duración y periódicas en el tiempo conforme a la señal de potencia de 60Hz.

Capítulo 3

Estimación de canal en sistemas OFDM

En este capítulo se presenta una breve revisión de las técnicas de estimación de canal empleadas en los sistemas de comunicación con multiplexado por repartición de frecuencias ortogonales (OFDM, por sus siglas en inglés). Muchas de estas técnicas fueron originalmente diseñadas para canales inalámbricos pero usualmente han sido adaptadas con éxito para su aplicación en los canales de comunicación por la red eléctrica (PLC, por sus siglas en inglés) debido a que comparten las características de los canales atenuados.

3.1. Breve reseña sobre OFDM y DMT

Los sistemas con multiplexado por repartición de frecuencias ortogonales (OFDM) han sido objeto de un gran interés en el ámbito de las comunicaciones PLC debido a sus ventajas para transmitir datos a gran velocidad a través de canales selectivos en frecuencia y variantes en el tiempo. En particular, dentro del ámbito de los canales cableados como lo es la línea eléctrica, la aplicación de OFDM con asignación adaptativa de los bits de datos es ampliamente conocida como técnica multi-tono discreto (DMT, por sus siglas en inglés). La técnica DMT es una variante de la modulación multiportadora que reparte eficientemente una secuencia de bits, con datos de entrada, en varias secuencias paralelas de menor velocidad asignadas a subportadoras/subcanales ortogonales. Las ventajas adquiridas incluyen la adaptabilidad de la velocidad de transmisión en función de la relación señal a ruido (SNR), eficiencia espectral, robustez ante la interferencia inter-símbolo (ISI) y una inherente inmunidad tanto al ruido impulsivo como a la radio interferencia (RFI).

En la técnica DMT la SNR de cada subcanal es estimada en una sesión inicial de entrenamiento, de tal manera que el número total de bits es repartido en los subcanales de acuerdo con su respectiva SNR. Un subcanal con una SNR grande transportará más bits que uno con una SNR baja.

3.2. Estimación del canal

Como los sistemas OFDM emplean detección coherente, se requiere un conocimiento perfecto de la atenuación del canal y de la potencia del ruido en cada subportadora. De manera más general,

incluso es posible asignar a cada subportadora los símbolos provenientes de diferentes esquemas de modulación y transmitirlos con potencias distintas. Esto representa la ventaja de poder adaptar la señal emitida al canal de propagación, siempre que se tenga el conocimiento del estado del canal en el transmisor. Con ello, cada subportadora debería transportar, en un solo símbolo de modulación, tantos bits como lo permita la atenuación y nivel de ruido observado en la frecuencia de dicha subportadora [9]. Este principio, denominado de modulación adaptativa, es el principal beneficiario del realizar una buena estimación de canal, que es justamente el propósito de nuestro trabajo de investigación.

Así, cada transreceptor debe disponer del conocimiento del estado del canal sobre el cual va a transmitir la información. Este conocimiento comprende no solamente la estimación de la respuesta del canal, sino también la estimación del ruido y perturbaciones presentes en éste, incluyendo la interferencia producida por los transreceptores vecinos.

3.2.1. Procesos básicos de cualquier receptor OFDM

Tres son los procesos que trabajan en forma continua y paralela en cada uno de los receptores OFDM:

- Sincronización,
- estimación del canal y
- estimación del ruido.

Se dice que dos señales a portadora múltiple están sincronizadas si la ortogonalidad entre las subportadoras se preserva una vez recibidas las señales, lo que es necesario para el correcto procesamiento de la señal recibida. La estimación del canal es necesaria no sólo para realizar el igualamiento de los datos, sino también para los procedimientos de asignación dinámica del espectro entre los usuarios. Por esta razón, es muy importante que cada receptor estime la respuesta del canal sobre toda la banda de la señal y no únicamente sobre las subportadoras asignadas al usuario. Por su parte, la estimación del ruido permite calcular la relación señal a ruido (SNR), útil para la adaptación de la modulación del canal, además de ser necesaria en el marco de un detector de mínimo error cuadrático medio.

En este trabajo únicamente nos vamos a concentrar en el problema de la estimación del canal, asumiendo que la sincronización es perfecta y que funciona correctamente la estimación del ruido.

3.2.2. La señal OFDM recibida en un canal multi-trayectoria con sincronización perfecta

Considerando únicamente las señales en la salida de la operación FFT del receptor, tenemos una señal OFDM clásica (Ver capítulo 2). A partir del modelo multi-trayectoria, expresado por la

ecuación

$$h(t) = \sum_{i=1}^N C_i \cdot \delta(t - \tau_i) \quad (3.1)$$

es posible obtener un muestreo de la respuesta al impulso del canal de propagación. Así, se obtiene, por interpolación, un vector $\mathbf{h}_t = [h_0 \dots h_{L-1}]^T$ compuesto de L coeficientes complejos correspondientes a L muestras. Entonces, la señal analítica recibida se expresa

$$r_n = h_t * s_n + w_n = \sum_{l=0}^{L-1} h_l s_{n-l}^* + w_n \quad (3.2)$$

donde s_n representa las muestras de la señal transmitida con portadoras múltiples, w_n las muestras de ruido blanco y $(\cdot)^*$ representa la operación de complejo conjugado. Considerando un intervalo de guardia de D muestras, los símbolos transmitidos se escriben en la forma de bloques temporales \mathbf{t}_m compuestos de $N+D$ muestras, lo que vectorialmente se expresa $\mathbf{t}_m = [s_{m,N-D} \dots s_{m,N-1} s_{m,0} \dots s_{m,N-1}]^T$, con m el índice del bloque temporal. Si el receptor tiene un conocimiento perfecto del ritmo de las muestras y de los símbolos, entonces son suprimidas las primeras D muestras recibidas de cada bloque temporal. Asumiendo que $D > L$, entonces el vector de muestras obtenidas no sufre ninguna interferencia por parte del bloque temporal precedente. Empleando el hecho de que las D muestras del intervalo de guardia constituyen una repetición de las últimas D muestras de cada símbolo $\mathbf{s}_m = [s_{m,0} \dots s_{m,N-1}]^T$, la señal recibida se escribe:

$$\mathbf{r}_m = \mathbf{T}_h \cdot \mathbf{s}_m + \mathbf{w}_m \quad (3.3)$$

con,

$$\mathbf{T}_h = \begin{bmatrix} h_0 & 0 & \dots & \dots & h_{L-1} & \dots & \dots & h_1 \\ h_1 & \ddots & & & & \ddots & & h_2 \\ \vdots & & \ddots & & & & \ddots & \vdots \\ \vdots & & & \ddots & & & & h_{L-1} \\ h_{L-1} & & & & \ddots & & & \vdots \\ 0 & \ddots & & & & \ddots & & \vdots \\ \vdots & & \ddots & & & & \ddots & 0 \\ 0 & \dots & 0 & h_{L-1} & \dots & \dots & h_1 & h_0 \end{bmatrix} \quad (3.4)$$

siendo $\mathbf{r}_m = [r_0 \dots r_{N-1}]^T$ el vector de muestras obtenido después de la supresión del intervalo de guardia. La matriz del canal \mathbf{T}_h utilizada es una matriz particular de Toeplitz debido a que esta es cíclica por construcción. Así, el resultado del producto matricial es el producto de convolución cíclica de los coeficientes del canal para las muestras del símbolo \mathbf{s}_m . La matriz \mathbf{T}_h es de dimensión $N \times N$ y su primer columna $\mathbf{v}_1 = [h_t^T 0 \dots 0]^T$. Por propiedad, se sabe que toda matriz cíclica es diagonalizable en una base de vectores propios de Fourier, y sus valores propios no son otros que el resultado de la transformada de Fourier de su primera columna. Dicho de otra forma, se puede escribir $\mathbf{T}_h = \mathbf{F}^H \mathbf{H} \mathbf{F}$, con $\mathbf{H} = \text{diag}(\mathbf{F} \mathbf{v}_1)$. Recordando que la escritura de una señal

multiportadora se expresa,

$$\mathbf{s}_m = \mathbf{F}^{-1} \mathbf{x}_m = \mathbf{F}^H \mathbf{x}_m \quad (3.5)$$

con $\mathbf{s} = [s_0 \ s_1 \ \dots \ s_{N-1}]^T$ el vector de muestras en el tiempo del símbolo OFDM, $\mathbf{x} = [x_0 \ x_1 \ \dots \ x_{N-1}]^T$ el vector de símbolos de modulación (por ejemplo, QAM o BPSK) transmitidos en cada subportadora y \mathbf{F} es la matriz de Fourier de dimensiones $N \times N$.

Por lo que podemos describir a la señal recibida, después del muestreo, como

$$\mathbf{r}_m = \mathbf{F}^H \mathbf{H} \mathbf{F} \cdot \mathbf{F}^H \mathbf{x}_m + \mathbf{w}_m \quad (3.6)$$

Entonces, el vector de símbolos $\mathbf{y}_m = [y_{m,0} \ \dots \ y_{m,N-1}]^T$, obtenido después de la operación de FFT en la recepción, es

$$\mathbf{y}_m = \mathbf{F} \cdot (\mathbf{F}^H \mathbf{H} \mathbf{F}) \cdot \mathbf{F}^H \mathbf{x}_m + \mathbf{F} \mathbf{w}_m = \mathbf{H} \mathbf{x}_m + \mathbf{w}'_m \quad (3.7)$$

Así, después de la demodulación OFDM, se recuperan los símbolos $x_{m,k}$, transmitidos en cada subportadora, afectados por un factor multiplicativo correspondiente al valor que toma la función de transferencia del canal en la frecuencia f_k , $k = [0 \ \dots \ N-1]$. El vector \mathbf{w}'_m se mantiene como un vector de muestras independientes de ruido blanco gaussiano con la misma varianza debido a que la transformada de Fourier de un vector gaussiano no cambia sus estadísticas. Sabiendo que \mathbf{H} puede escribirse también $\mathbf{H} = \text{diag}(\mathbf{F}_{N,L} \mathbf{h}_t)$, con $\mathbf{F}_{N,L}$ la matriz de Fourier de dimensión $N \times L$, se puede obtener una segunda expresión para la señal con portadoras múltiples en la recepción:

$$\mathbf{y}_m = \mathbf{X}_m \mathbf{F}_{N,L} \mathbf{h}_t + \mathbf{w}'_m = \mathbf{X}_m \mathbf{h}_f + \mathbf{w}'_m \quad (3.8)$$

con $\mathbf{X}_m = \text{diag}(\mathbf{x}_m)$ y $\mathbf{h}_f = [H_0 \ \dots \ H_{N-1}]^T$.

3.2.3. Igualamiento y estimación del canal

Deducimos de las expresiones precedentes que la distorsión introducida por el canal en cada subportadora k puede ser compensada simplemente mediante la multiplicación de cada símbolo $y_{m,k}$ por un coeficiente complejo g_k apropiado. Nótese que escribimos estos coeficientes como g_k y no $g_{m,k}$, debido a que suponemos un canal invariante a escala de los símbolos. Entonces, se aplica una matriz de igualamiento \mathbf{G} a la señal espectral recibida, o sea

$$\tilde{\mathbf{y}}_m = \mathbf{G} \mathbf{H} \mathbf{x}_m + \mathbf{G} \mathbf{w}'_m \quad (3.9)$$

con $\mathbf{G} = \text{diag}(\{g_k\}_{0 \leq k \leq N-1})$. La elección de los coeficientes g_k se deriva de algún criterio de recepción dado, tal como el criterio de mínimo error cuadrático medio (MMSE, por *Minimum Mean Square Error*) o incluso el criterio de decorrelación (ZF, por *Zero-forcing*) [121]. Así, la operación de convolución del filtro adaptativo utilizado para el igualamiento del canal en los sistemas mono portadora se reduce a una simple operación de multiplicación compleja. Sin embargo, debemos aclarar que la matriz \mathbf{G} es diagonal sólo si el vector \mathbf{x}_m tiene una estructura desconocida, en el sentido algebraico del término. Este es efectivamente el caso de OFDM, ya que precisamente \mathbf{x}_m

no tiene una estructura particular debido a que es un vector de símbolos aleatorios. Por el contrario, si se emplea la función de dispersión, donde se da a \mathbf{x}_m una estructura conocida en el receptor, ésta debería ser tomada en cuenta, en teoría, durante la aplicación del criterio de recepción y, en consecuencia, para la obtención de la matriz \mathbf{G} resultante. En la práctica, frecuentemente se omite la estructura de \mathbf{x}_m a fin de obtener receptores simples. No obstante, cualquiera que sea el receptor en operación, ya sea en términos de complejidad o del criterio de recepción, es necesario conocer las atenuaciones del canal, para cada subportadora del espectro, a fin de construir la matriz \mathbf{G} . Por lo tanto, en la aplicación es determinante el módulo de *estimación de canal*.

3.2.4. Tipos de estimadores de canal

En sistemas OFDM reales, que operan en un canal con dispersión de la respuesta al impulso (*time-dispersive channel*), en la secuencia transmitida típicamente se inserta un prefijo cíclico (CP o intervalo de guardia) de mayor duración que la susodicha dispersión de la respuesta al impulso. Como resultado, el efecto filtro del canal se traduce en una distorsión multiplicativa en cada subcanal OFDM. Entonces la estimación del canal se expresa como sigue,

$$\hat{h}_{m,k} = \frac{y_{m,k}}{x_{m,k}}$$

donde $\hat{h}_{m,k}$ es el estimado para el coeficiente de atenuación complejo para la subportadora k del m ésimo símbolo. Entonces, dependiendo del origen del patrón de referencia $x_{m,k}$ para el denominador, podemos clasificar a las técnicas de estimación del canal en cuatro:

- Técnicas asistidas por pilotos (*Pilot-aided*). La fuente del símbolo transmitido $x_{m,k}$ es un patrón conocido o piloto. Se emplean algunos símbolos específicos, dedicados a estas operaciones de estimación, que son insertados en el flujo de datos como símbolos de reconocimiento y que también pueden servir para propósitos de sincronización. En el caso de los canales variantes en el tiempo, el símbolo piloto debe ser repetido periódicamente a fin de estimar dinámicamente al canal.
- Técnicas regidas por el proceso de decisión (*Decision-directed* o DD) de la detección de los símbolos. La fuente del símbolo transmitido $x_{m,k}$ es el símbolo detectado. Se asume que la última estimación del canal proporciona un coeficiente de igualamiento tal que permite estimar aproximadamente al símbolo transmitido $\hat{x}_{m,k} = y_{m,k}g_k$ y obtener un símbolo $\tilde{x}_{m,k}$ a la salida del proceso de detección. Así, en las técnicas DD la última estimación del canal se calcula con base en el símbolo detectado, a saber:

$$\hat{h}_{m,k} = \frac{y_{m,k}}{\tilde{x}_{m,k}}$$

- Técnicas *semi-ciegas* (*semi-blind channel estimation*). Se parte de una estimación inicial con base en símbolos pilotos y, luego, se va siguiendo (*Tracking*) la evolución del canal sin emplear símbolos de entrenamiento (“a ciegas”). La técnica DD puede ser catalogada como *semi-ciega*, ya que emplea símbolos pilotos de entrenamiento únicamente al inicio de cada trama.

- Técnicas *ciegas* (*blind channel estimation*). Los métodos ciegos tienen su fundamento en la estadística de los símbolos de datos desconocidos y las propiedades estadísticas del canal, por lo que no requieren de ningún símbolo piloto.

Finalmente, debemos recordar que las técnicas estadísticas con propósitos de estimación pueden emplear alguno de los siguientes enfoques [76]:

- **Técnicas clásicas.** Cuando el parámetro a estimar se considera desconocido pero determinista. Destacan las técnicas de maximización de la verosimilitud (*Maximum Likelihood Estimator, ML*) y el método de mínimos cuadrados (*Least Squares Estimator, LS*).
- **Técnicas Bayesianas.** Cuando el parámetro a estimar se considera desconocido y además estocástico, regido por una ley de probabilidad. Destacan la técnica de minimización del error cuadrático promedio (*Minimum Mean Square Error Estimator, MMSE*), el método de maximización *a posteriori* (*Maximum a Posteriori Estimator, MAP*) y el método de minimización del error lineal cuadrático promedio (*Linear Minimum Mean Square Error Estimator, LMMSE*).

Ya que las técnicas de estimación de canal para canales PLC con modulación OFDM coherente son, típicamente, adaptaciones de las técnicas desarrolladas originalmente para los canales inalámbricos [39, 87], a continuación se presenta un breve estudio de los últimos avances en este campo.

3.3. Evolución de la estimación del canal para sistemas OFDM

Es conocido que en cualquier sistema de comunicación, la detección coherente de datos no es confiable si no se cuenta en el receptor con una estimación fiel del canal. Aunque puede ser usada la modulación diferencial (por ejemplo, DPSK) para detectar los datos transmitidos sin ningún conocimiento del canal, esta clase de modulación implica restricciones en la elección de la constelación y produce un menor desempeño en comparación con las modulaciones coherentes. Engels y su equipo propusieron en [55], como una alternativa a DPSK, la modulación mediante codificación con cambios diferenciales de amplitud y fase (DAPSK), en la que la amplitud de los símbolos transmitidos también es codificada diferencialmente. Por su parte, una modulación coherente permite el uso de constelaciones arbitrarias y constituye una opción buena y flexible cuando las variaciones temporales del canal son relativamente lentas. Además, usando constelaciones de alta eficiencia espectral, la modulación coherente puede alcanzar altas velocidades de transferencia de datos, como en la norma de transmisión de video digital.

Existe una literatura muy rica en técnicas de estimación de canal para sistemas OFDM. En lo que sigue, presentamos las principales contribuciones para cada una de las técnicas de estimación del canal, muchas de las cuales fueron desarrolladas originalmente para los canales inalámbricos.

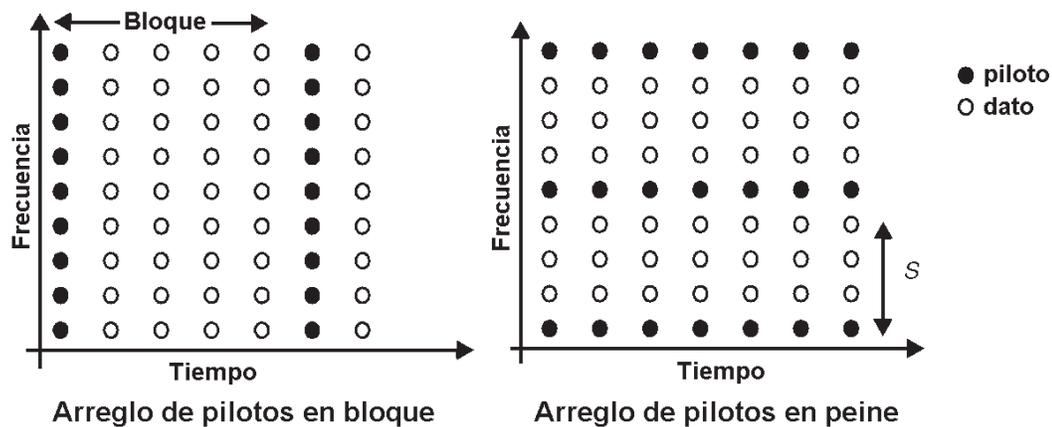


Figura 3.1: Arreglos básicos para distribuir los símbolos pilotos

3.3.1. Técnicas de estimación del canal únicamente con base en pilotos

Aquí describimos diferentes técnicas de estimación de canal con base en símbolos pilotos. Estas técnicas que también son llamadas de modulación asistida por símbolos pilotos (PSAM), fueron introducidas para sistemas de portadora única por Moher y Lodge [102] y más tarde analizadas por Cavers [29].

Dos formas básicas de colocar los símbolos pilotos en sistemas OFDM se ilustran en la Figura 3.1, que generalmente se consideran para estimaciones del canal con fundamento en filtros mono-dimensionales. La primera, es la estimación asistida por pilotos con arreglo en bloque, que se utiliza cuando se asume que el canal se atenúa en forma lenta. Se pone en práctica insertando tonos pilotos en todas las subportadoras en un instante determinado. La segunda, es la estimación asistida por pilotos con arreglo en peine, que se utiliza cuando se sabe que el canal cambia de un símbolo OFDM a otro. En este caso, se insertan tonos piloto en ciertas subportadoras de cada símbolo OFDM, con lo que se hace necesaria cierta interpolación para determinar el estado de las subportadoras restantes.

En la estimación bidimensional, los pilotos son insertados tanto en el dominio del tiempo como en el dominio de la frecuencia y los estimadores tienen generalmente su fundamento en filtros bidimensionales. Aunque la estimación bidimensional produce un mejor desempeño que los esquemas mono-dimensionales, esto tiene una mayor complejidad computacional y mayor consumo de tiempo. Debido a lo anterior es que se han desarrollado estimadores bidimensionales simplificados empleando, por ejemplo, filtrado mono-dimensional por partes. Incluso existen estimadores de canal iterativos, en los que la estimación se realiza bloque por bloque, filtrando por partes y decodificando en forma iterativa. También existen diversos métodos iterativos con fundamento en algoritmos de probabilidad *a posteriori* (AP), en los que subyacen dos estimadores AP, uno para el dominio de la frecuencia y otro para el dominio del tiempo, trabajando en un lazo iterativo similar al principio de la decodificación *turbo* [73]. Los estimadores iterativos producen un desempeño robusto, incluso con escenarios de baja relación señal a ruido (SNR), pero con el costo de una gran

complejidad computacional y cierto tiempo de retraso por iteración.

Técnicas con base en el criterio de mínimos cuadrados

Se sabe que en OFDM, gracias a la introducción del intervalo de guardia con prefijo cíclico (CP), cada subportadora ve un canal de atenuación llana. Consideremos una trama de M símbolos OFDM consecutivos con K subportadoras por símbolo. La trama transmitida puede ser vista como una rejilla de frecuencia-tiempo de tamaño $(M \times K)$. El símbolo recibido $y_{m,k}$, en cada posición de la rejilla se escribe como

$$y_{m,k} = h_{m,k}x_{m,k} + w_{m,k} \quad (3.10)$$

donde $h_{m,k}$ y $x_{m,k}$ son, respectivamente, la respuesta en frecuencia del canal (CFR) y el símbolo transmitido en la (m, k) posición de la rejilla, y $w_{m,k}$ es el AWGN que afecta a la transmisión de datos. Un método simple es estimar $h_{m,k}$ y minimizar $|y_{m,k} - h_{m,k}x_{m,k}|^2$. Así, en presencia de ruido Gaussiano independiente e idénticamente distribuido (i.i.d.), la estimación de mínimos cuadrados (LS) está dada por

$$\hat{h}_{m,k}^{\text{LS}} = \frac{y_{m,k}}{x_{m,k}} \quad (3.11)$$

donde se asume que $x_{m,k}$ es conocido en el receptor. En la técnica PSAM, algunos símbolos de entrenamiento son insertados tanto en tiempo como en frecuencia en la rejilla bidimensional. El intervalo de repetición de pilotos depende del tiempo de coherencia del canal y en el ancho de banda coherente. Obviamente, en PSAM una estimación del canal es obtenida sólo en las posiciones de los pilotos. A fin de estimar el canal en todas las posiciones de tiempo y frecuencia, en la literatura se han propuesto diferentes técnicas de interpolación, que describimos brevemente a continuación.

Interpolación polinomial. La Interpolación se puede realizar en el dominio de la frecuencia con un único símbolo OFDM [122] o en ambos dominios (tiempo-frecuencia) [30]. Por ejemplo, en [122], Rinne y Renfors proponen a dos estimadores LS de canal. La primera técnica estima el canal en las frecuencias de los pilotos y asume que el canal es constante en el ancho de banda igual al espacio en frecuencia entre dos tonos pilotos (centrados en la frecuencia del tono piloto). La segunda técnica realiza una interpolación en el dominio de la frecuencia considerando que el canal cambia en línea recta entre dos tonos pilotos consecutivos. En [30], Chang y Su propusieron una interpolación bidimensional en el sentido de LS considerando un modelo parabólico para el canal tal que

$$h_{m,k} = a_1m^2 + a_2mk + a_3k^2 + a_4m + a_5k + a_6 \quad (3.12)$$

donde los coeficientes a_1, \dots, a_6 son determinados minimizando

$$\sum_{m=0}^{M-1} \sum_{k=0}^{K-1} |\hat{h}_{m,k}^{\text{LS}} - h_{m,k}|^2. \quad (3.13)$$

Interpolaciones polinomiales por partes (*piecewise*) de orden más alto son propuestas en [71] donde la amplitud de banda de un símbolo OFDM es dividida en bandas de frecuencia diferentes. Sobre cada banda de frecuencia, el canal es modelado como un polinomio y el símbolo OFDM entero es

recuperado considerando cada polinomio elemental.

Interpolación usando filtros FIR. Como una alternativa a la interpolación polinomial, Moon y Choi propusieron una interpolación bidimensional diferente en [103]. Esta técnica usa filtros Gaussianos y filtros de plantillas (*spline*) cúbicas para la interpolación. Además, los filtros de interpolación adoptados tienen una respuesta de impulso finita (FIR) con tres coeficientes y usan los coeficientes estimados del canal que proporcionan tres símbolos pilotos. Sin embargo, al contrario del enfoque propuesto en [122], la interpolación bidimensional se realiza por dos filtros FIR unidimensionales, uno realiza la interpolación en el eje del tiempo y el otro en el eje de la frecuencia. En [37], el canal es estimado en las frecuencias de los pilotos y luego convertido al dominio temporal por una IFFT. Entonces la señal es interpolada usando un filtro de platilla cúbica antes de ser convertido al dominio de la frecuencia. Ozinawa *et al.* proponen en [113] un método para seleccionar un filtro FIR de interpolación entre un conjunto predeterminado de filtros.

Técnicas con base en la minimización del error cuadrático medio

El criterio del mínimo error cuadrático medio (MMSE) ha sido usado ampliamente en OFDM para la estimación del canal. La estimación bidimensional óptima del canal, en el sentido del MMSE, fue tratada en [67]. Debido a la alta complejidad de este estimador, se han propuesto diferentes estimadores sub-óptimos con complejidades inferiores, como en [66]. Otros trabajos en esta área han sugerido el uso de dos filtros unidimensionales en cascada que realizan las respectivas estimaciones MMSE en el dominio del tiempo y de la frecuencia.

En [135], Van De Beek *et al.* propusieron modificaciones al estimador MMSE de la respuesta en frecuencia asumiendo una respuesta de impulso de longitud finita. Este trabajo usa la teoría de aproximaciones de bajo rango (*low-rank*), con base en el truncamiento de las matrices de Fourier discreto (DFT). El principio de aproximación de bajo rango consiste en asumir que la mayoría de la energía de la respuesta al impulso, correspondiente a cierta respuesta en frecuencia, se encuentra en las primeras $L + 1$ muestras (*taps*), donde L son las muestras que corresponden al intervalo de guardia. Así, se pueden considerar únicamente las muestras con energía significativa. Inspirado por las observaciones hechas en [135], Edfors *et al.* proponen en [53] un estimador MMSE lineal (LMMSE) usando la descomposición en valores simples (SVD, por sus siglas en inglés) de la matriz de covarianza de la respuesta en frecuencia del canal. La complejidad de este estimador es menor al usar la reducción de rango óptimo. La técnica de reducción de rango es adoptada por Hsieh en [70] donde la función de transferencia del canal de los tonos pilotos es estimada usando un estimador MMSE de rango bajo, mientras que los valores de la función de transferencia del canal son interpolados mediante una interpolación lineal por partes.

En [90], Li *et al.* amplían el estimador de [53] y obtienen un estimador MMSE que hace uso pleno de la correlación de la respuesta de frecuencia de canal en diferentes tiempos y frecuencias. En este método, la correlación de la respuesta en frecuencia del canal se separa en una multiplicación de las funciones de correlación en el dominio del tiempo y la frecuencia. Se propuso un método similar en [126] donde se realiza un filtrado MMSE, por separado, en los dominios del

tiempo y la frecuencia. El algoritmo propuesto calcula el retardo de dispersión del canal y usa este resultado para mejorar el filtrado en el dominio de la frecuencia que es realizado por un filtro adaptativo. En [91], los autores desarrollaron la estimación de canal OFDM para el caso de una codificación espacio-tiempo y múltiples antenas, obteniendo algunos límites para el error cuadrático medio (MSE).

Algunas técnicas encontradas en la literatura tienen su fundamento en los filtros de Kalman. Por ejemplo, en [134], Tufvesson y Maseng usaron un filtro de Kalman para estimar los pasos del filtro, para un canal variante en el tiempo, que son modelados por un proceso autoregresivo (AR) de primer orden.

3.3.2. Técnicas regidas por el proceso de decisión

El inconveniente principal de técnicas de estimación de canal con base únicamente en pilotos es la pérdida de la eficiencia espectral debida a la sobrecarga (*overhead*) de los pilotos. El tamaño de la secuencia de entrenamiento transmitida puede ser reducido adoptando los métodos regidos por la decisión (DD) del proceso de detección, que requieren únicamente de la transmisión de un piloto OFDM al inicio de la trama. La idea básica de este método es que, en ausencia de errores de transmisión, se pueden usar los símbolos detectados como señales de referencia *a posteriori*, en vez de símbolos pilotos, para la estimación del canal. Un trabajo pionero en esta área es el iniciado por Frenger y Svensson [58, 57] donde proponen un detector DD coherente para sistemas de multipor-tadora y mono-portadora con base en una estimación de canal MMSE. Sin embargo, esta técnica asume que todas las decisiones anteriores de detección son correctas en cada instante. Obviamente, esto no es una suposición realista en situaciones prácticas.

A fin de mitigar el problema de la propagación del error, Mignone y Morello [101] propusieron un estimador de canal DD que aprovechaba los códigos de corrección de errores (FEC). En esta técnica, los símbolos decodificados confiables son usados para la estimación/actualización de canal y el problema de la propagación de error es disminuido. Sin embargo, el lazo de estimación de canal de este método requiere largos retrasos que pueden volverse restrictivos para su aplicación sobre todo en canales de ámbitos móviles. En [19], Bulumulla *et al.* consideraron un receptor de MAP con base en la retroalimentación de la decisión, donde la probabilidad *a posteriori* de los símbolos es calculada usando un filtro de Kalman.

Recientemente, Akhtman y Hanzo [3] propusieron un estimador DD de canal para sistemas OFDM que emplea un modelo parcialmente espaciado (*fractionally-spaced*) para la respuesta al impulso. La adopción del modelo parcialmente espaciado es motivada por el hecho que en ambientes realistas de canal, no se tiene ningún control sobre el retraso en la dispersión de la respuesta al impulso.

3.3.3. Técnicas semi-ciegas de estimación de canal

Los métodos ciegos también pueden ser usados en combinación con los datos de entrenamiento, a fin de rastrear mejor las variaciones del canal y permitir una convergencia más rápida. Estos métodos de estimación del canal se denominan como métodos “semi-ciegos” [44]. Por lo general, uno o dos símbolos pilotos son transmitidos al principio de cada trama para la sincronización y estimación inicial del canal. La mayor parte de algoritmos ciegos descritos arriba pueden ser extendidos al caso semi-ciego. Por ejemplo, en [107] se usa un símbolo piloto para eliminar el tiempo de convergencia del algoritmo ciego. Además, las subportadoras pilotos son usadas para superar la indeterminación escalar inherente que es común en la mayoría de los algoritmos ciegos (ver [92], por ejemplo). En [105] los autores presentan un sistema OFDM con relleno de ceros (*zero-padding* o ZP) en el que las muestras de valor cero que normalmente son insertadas entre cada bloque OFDM son sustituidos por una secuencia pseudo-aleatoria de escalares. Las observaciones obtenidas de esta manera son usadas para realizar una estimación semi-ciega del canal y el rastreo del mismo.

También se han sugerido técnicas de estimación semi-ciega del canal con fundamento en el eficiente y ampliamente usado algoritmo de Maximización de Esperanza (EM), como en el caso de [124]. El algoritmo EM es un algoritmo iterativo que puede ser usado para obtener una solución aproximada a los criterios ML o MAP, para un canal desconocido y donde los símbolos transmitidos también son desconocidos por el estimador del canal (caso ciego). Para una exposición más detallada del algoritmo EM, el lector puede remitirse a [45].

3.3.4. Técnicas de estimación del canal totalmente ciegas

La creciente necesidad de mayores velocidades de transferencia de datos motivó la búsqueda de métodos ciegos, tanto de identificación de canal como de igualamiento, con un ahorro en el ancho de banda al evitar el uso de secuencias de entrenamiento.

Numerosos algoritmos ciegos han sido desarrollados en la literatura [93], de los cuales múltiples trabajos se han concentrado expresamente en sistemas multiportadora. Los métodos existentes de estimación ciega del canal para sistemas OFDM, por lo general, se clasifican como estadísticos o como determinísticos.

Por ejemplo, entre los métodos ciegos del tipo estadístico, está el de [65, 106] donde se aprovecha la ciclo-estacionariedad inherente inducida por el intervalo de guardia. Expresamente, Heath y Giannakis [65] proponen un método ciego con base en la propiedad de ciclo-estacionariedad de la correlación de los datos recibidos debida a la inserción del intervalo de guardia en el transmisor, sin embargo, este enfoque sufre de una convergencia lenta para el estimador. Otros métodos en esta categoría tienen su fundamento en la descomposición en subespacios [104]. Estos algoritmos explotan la ortogonalidad entre el subespacio del ruido y el subespacio de la señal y tienen su fundamento en la matriz de autocorrelación de los datos recibidos antes de la DFT [107, 26, 18]. Una técnica de estimación ciega del canal con base en sub-espacios es presentada por Zhou *et al.* [143] para sistemas OFDM con codificación espacio-temporal que emplea precodificación lineal

[138]. Este último algoritmo, al usar precodificación lineal, garantiza la identificación del canal, independientemente de las posiciones de los ceros subyacentes. Recientemente, Lin y Petropulu [92] propusieron un sistema OFDM lineal, precodificado, no-unitario y no-redundante, usando la correlación introducida por la matriz de precodificación para estimar el canal en el receptor con base en operaciones de correlación cruzada.

Para propósitos de estimación ciega de canal, además del intervalo de guardia, también han sido aprovechadas las subportadoras virtuales. En [88] se presenta un estimador conjunto de máxima verosimilitud (ML), tanto para la estimación ciega del canal como para la detección de datos, que aprovecha la propiedad finita del alfabeto de modulación y la presencia de portadoras virtuales.

Sin embargo, cada uno de los algoritmos anteriores tiene sus propias limitaciones. Por ejemplo, los métodos con base en subespacios típicamente requieren un número grande de símbolos OFDM introduciendo así un retraso considerable en el sistema en su conjunto. Un algoritmo con base en un alfabeto finito puede ser aplicado sólo a una señal de magnitud (módulo) constante. Así, son preferidos los estimadores que requieren pocos símbolos OFDM, ya que pueden funcionar sin introducir retardos apreciables.

Los métodos deterministas son, por lo general, aplicados en símbolos OFDM recibidos después de la demodulación FFT y usan la propiedad de alfabeto finito de los símbolos recibidos. Los métodos deterministas por lo general convergen mucho más rápido que los estadísticos, pero implican una mayor complejidad computacional en el receptor. Esta categoría incluye enfoques con criterio ML [34] y enfoques de búsqueda exhaustiva [142]. El método en [34] tiene la ventaja de producir una estimación de canal a partir de un sólo símbolo OFDM recibido. Su inconveniente principal es la enorme complejidad computacional requerida para ejecutar la operación de maximización implícita en el algoritmo. Sin embargo, la estimación del canal arrastra cierta ambigüedad de la fase. En [110], Necker y Stuber modificaron el método ML básico de [34] para el caso de señales PSK y desarrollaron una versión iterativa de baja complejidad que aprovecha la correlación en el dominio temporal de la función de transferencia.

3.4. Ejemplos de estimadores de canal asistidos por símbolos pilotos

A modo de ejemplo, en los siguientes apartados damos una breve descripción de los métodos de estimación considerados como clásicos [73].

3.4.1. Estimación del canal en el sentido de mínimos cuadrados (LS)

El método de estimación del canal más simple es aquel donde los símbolos transmitidos son conocidos y es, justamente, la estimación en el sentido de mínimos cuadrados (*Least Squares, LS*).

En ese caso, un coeficiente de atenuación del canal se expresa simplemente como

$$c_{k,l}^{\text{LS}} = \frac{y_{k,l}}{a_{k,l}} \quad (3.14)$$

donde $y_{k,l}$ es la muestra del $l^{\text{ésimo}}$ símbolo OFDM recibido sobre la $k^{\text{ésima}}$ portadora, y $a_{k,l}$ el símbolo transmitido correspondiente.

En la literatura encontramos dos grandes familias de estimadores en el sentido de LS. La primera se compone de los estimadores regidos por decisión (DD), mientras que la segunda incluye a los sistemas para los que la información piloto es repartida en la totalidad del espacio tiempo-frecuencia de la señal OFDM. Este tipo de estimadores realiza una interpolación, ya sea en la frecuencia únicamente [71, 103] o en forma bidimensional [30], a partir de la estimación LS realizada a nivel de los símbolos pilotos y para la totalidad de los símbolos de datos.

3.4.2. Estimación del canal por el criterio del mínimo error cuadrático medio (MMSE)

El criterio del mínimo error cuadrático medio se encuentra muy frecuentemente en la literatura concerniente a la estimación del canal para sistemas OFDM. Este criterio se utiliza en dos grandes familias de estimadores. En la primera se incluye al filtro de Wiener, la aplicación más conocida del criterio MMSE. Los estimadores de la segunda familia realizan un filtrado pasa bajos, en el sentido de MMSE, en el dominio de transformación obtenido por la transformada de Fourier. En esta subsección presentamos solamente las técnicas MMSE *directas*. Las técnicas que utilizan el criterio de MMSE en el dominio de transformación son referidas más adelante.

El estimador lineal óptimo para los sistemas OFDM en el sentido del error cuadrático medio (MSE) es el filtro de Wiener 2-D (tiempo-frecuencia [67]). La complejidad de este estimador es a menudo demasiado grande para una utilización práctica. Por consiguiente, en la literatura se han propuesto estimadores sub-óptimos de menor complejidad [66]. Existen dos clases para estos estimadores: los estimadores bidimensionales y los estimadores separables. La utilización de filtros separables es un método corriente para reducir la complejidad en relación con el procesamiento de la señal multi-dimensional.

Por ejemplo, los coeficientes de atenuación del canal estimados en el sentido de los mínimos cuadrados (LS) se denotan:

$$\hat{c}^{\text{LS}} = \begin{bmatrix} y_0 & y_1 & \dots & y_{N-1} \\ x_0 & x_1 & & x_{N-1} \end{bmatrix}, \quad (3.15)$$

donde y_n son los elementos del vector de los datos recibidos y x_n son los elementos del vector de los símbolos transmitidos. La estimación final de los coeficientes de atenuación del canal se realiza por combinación lineal de los $\hat{c}_{k,l}^{\text{LS}}$ (por ejemplo, mediante un filtro de promedio), donde los coeficientes de la combinación lineal dependen de la estructura del estimador. El estimador en el sentido del mínimo error cuadrático medio lineal \hat{c}^{LMMSE} está dado por:

$$\hat{c}^{\text{LMMSE}} = \mathbf{R}_{cc} (\mathbf{R}_{cc} + \sigma_n^2 (\mathbf{X}\mathbf{X}^H)^{-1})^{-1} \hat{c}^{\text{LS}} \quad (3.16)$$

donde \mathbf{R}_{cc} es la matriz de autocovarianza del vector del canal \mathbf{c} . Ya que el cálculo de $(\mathbf{X}\mathbf{X}^H)^{-1}$ es especialmente demandante de tiempo de cómputo, se puede realizar una simplificación que produce poca degradación [53] si tomamos la media estocástica de este estimador. Entonces, la expresión del estimador simplificado es:

$$\hat{\mathbf{c}} = \left(\mathbf{R}_{cc} + \frac{\beta}{\text{SNR}} \mathbf{I} \right)^{-1} \hat{\mathbf{c}}^{\text{LS}} \quad (3.17)$$

donde $\beta = E[|x_k|^2]E[|1/x_k|^2]$ y $\text{SNR} = \frac{E[|x_k|^2]}{\sigma_n^2}$.

Variante de estimación MMSE por descomposición en valores simples

En [53], Edfors *et al.* presentan un método de estimación lineal en el sentido de MMSE cuya complejidad se reduce gracias a una reducción de rango óptima. Este método es obtenido por descomposición en valores simples de la matriz de autocorrelación del canal en frecuencia

$$\mathbf{R}_{cc} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H \quad (3.18)$$

donde \mathbf{U} es una matriz unitaria que contiene los vectores singulares de \mathbf{R}_{cc} y $\mathbf{\Lambda}$ es una matriz diagonal que contiene los valores singulares (simples) $\lambda_1 \geq \lambda_2 \geq \dots \lambda_N$ de \mathbf{R}_{cc} sobre su diagonal. Entonces, el estimador óptimo de rango p es:

$$\hat{\mathbf{c}}_p = \mathbf{U}\mathbf{\Delta}_p\mathbf{U}^H\hat{\mathbf{c}}^{\text{LS}} \quad (3.19)$$

donde $\mathbf{\Delta}_p = \text{diag}(\delta_1, \dots, \delta_N)$ es una matriz diagonal definida por:

$$\delta_k = \begin{cases} \frac{\lambda_k}{\lambda_k + \frac{\beta}{\text{SNR}}} & k = 1, 2, \dots, p \\ 0 & k = p + 1, \dots, N \end{cases} \quad (3.20)$$

3.4.3. Estimación por filtrado en el dominio de transformación

Estas técnicas transforman la señal recibida, es decir la estimación del canal LS o MMSE, hacia un dominio intermedio en el que se realiza cierto filtrado. La mayoría de los artículos que tratan de este tema utilizan la transformada inversa de Fourier para crear su dominio de transformación y emplean un filtro dependiendo del criterio elegido.

- **Filtrado rectangular - (también conocido como zero padding)** Se trata de una técnica simple utilizada para eliminar la contribución del ruido en la estimación del canal que se tiene. En este caso, el filtro es simplemente una ventana rectangular en el dominio de transformación. Por ejemplo, una manera práctica de mejorar la estimación de canal fue introducida en [33] y consiste en transformar la respuesta en frecuencia al dominio temporal, mediante una IFFT, para luego ventanear la respuesta al impulso, preservando únicamente las muestras que se encuentran dentro del intervalo correspondiente al máximo de dispersión prevista y asumiendo que las muestras restantes son exclusivamente ruido. Si procedemos de esta forma y retornamos al dominio de la frecuencia, mediante una FFT, obtenemos una respuesta en frecuencia suavizada, con lo que se mejora la estimación del canal en una forma simple (Ver Figura 3.2).

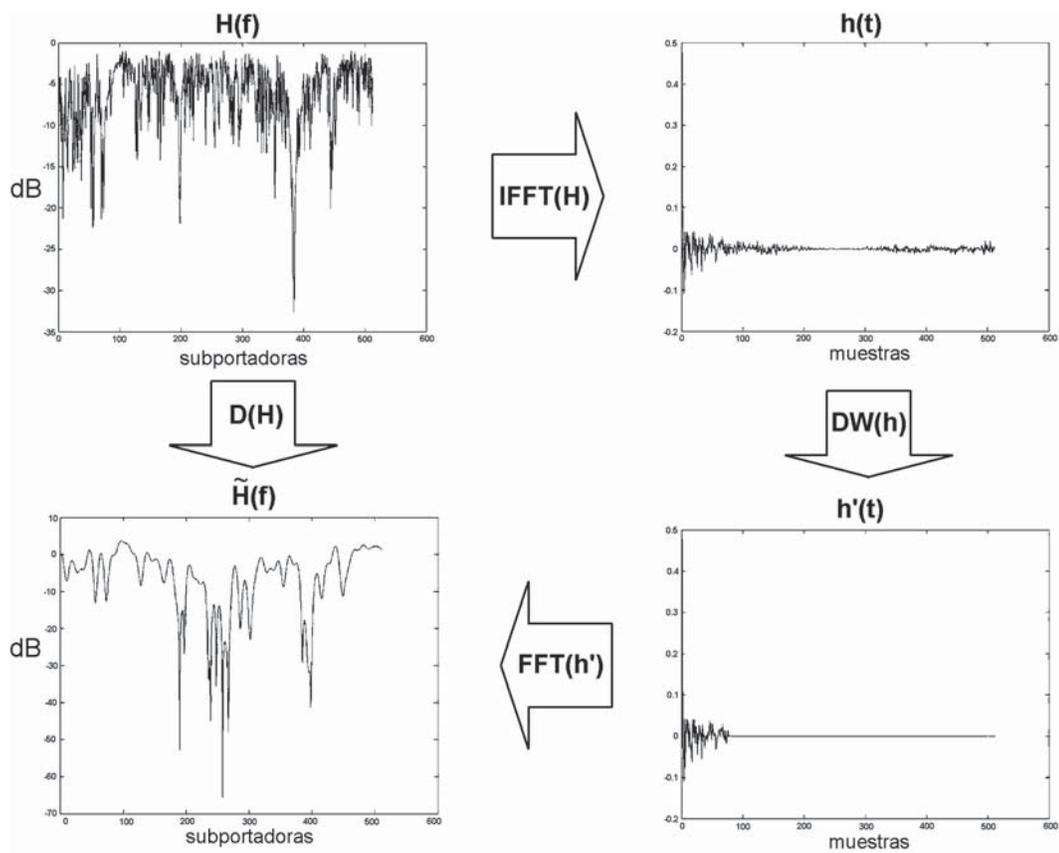


Figura 3.2: Mejora de la estimación de canal mediante ventaneo de la respuesta al impulso

- **Filtrado MMSE** Los métodos propuestos en [135] y [33] utilizan un filtro MMSE.
- **Otros filtros y otros dominios de transformación** En la literatura también encontramos otros tipos de filtrado como el filtro Kalman [134] y otros tipos de dominio de transformación, por ejemplo, el de la transformada de ondeleta. Específicamente, en el dominio resultante de la descomposición por ondeleta el proceso de filtrado se puede remplazar por un proceso de umbralado, como se propone en [140] y en esta tesis.

3.5. Estimación del espectro en potencia

El espectro en potencia está íntimamente relacionado con la respuesta en frecuencia de un canal si se considera al proceso de comunicación como un proceso estocástico, al canal como un filtro y a las muestras de la señal recibida como una serie de tiempo. Si la función de auto-correlación es una *descripción en el dominio temporal* de las estadísticas de segundo orden de un proceso estocástico, entonces el espectro en potencia es la *descripción en el dominio de la frecuencia* de las estadísticas de segundo orden del mismo proceso. En resumen, la *densidad espectral en potencia* se define como la transformada de Fourier discreta de la función de correlación de una serie de tiempo:

$$S(\omega) = \sum_{n=-\infty}^{\infty} r(n)e^{-j\omega n}, \quad -\pi < \omega \leq \pi \quad (3.21)$$

es decir,

$$r(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\omega)e^{j\omega n} d\omega, \quad n = 0, \pm 1, \pm 2, \dots \quad (3.22)$$

La densidad espectral en potencia es comúnmente conocida como *espectro en potencia* o simplemente *espectro*. En diversos campos de la ingeniería y de la ciencia física se considera al espectro en potencia de un proceso estocástico como la descripción más útil de las series de tiempo observadas y, por lo mismo, se hace importante su estimación [62]. Existen muchos métodos para estimar el espectro en potencia de un proceso estacionario del tipo denominado como de “sentido amplio” (*wide-sense*), pero pueden ser identificadas dos grandes familias:

- **Métodos paramétricos.** En estos métodos se propone un modelo estocástico para el proceso, de tal manera que el problema de estimación implica estimar los parámetros del modelo propuesto para representar aceptablemente, en un sentido estadístico, al proceso bajo estudio.
- **Métodos no paramétricos.** En estos métodos no se hace ninguna suposición relativa al proceso estocástico subyacente y, simplemente, se calcula el espectro con base en los datos disponibles en la serie de tiempo observada.

Los métodos paramétricos, por ejemplo, pueden emplear procedimientos para la identificación del sistema y proponer un modelo estocástico *Auto-regresivo* (AR), de *promedio móvil* (MA, por sus siglas en inglés) o combinación de ambos (ARMA), como base para la función de transferencia del proceso modelado. De los modelos estocásticos para la identificación del sistema, el más empleado es el AR debido a la forma lineal del sistema de ecuaciones que incluye a las incógnitas del modelo

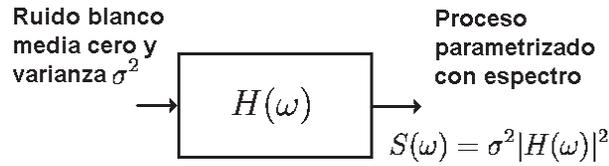


Figura 3.3: Principio de estimación paramétrica del espectro con modelo estocástico para $H(\omega)$

y a la existencia de algoritmos eficientes que calculan su solución. En estos modelos se emplea como entrada al proceso una fuente de ruido blanco con media cero y varianza σ^2 (ver Figura 3.3), de tal manera que el espectro en potencia $S(\omega)$ a la salida del modelo es igual a

$$S(\omega) = \sigma^2 |H(\omega)|^2 \quad (3.23)$$

donde $|H(\omega)|$ es la magnitud de la respuesta en frecuencia del modelo. En este enfoque, la función de autocorrelación se hace Hermitiana, lo que produce una potencia espectral $S(\omega)$ con valores reales, pero tiene como desventaja que se elimina la información de la fase. No obstante, la potencia espectral proporciona una aceptable descripción cualitativa de la respuesta en frecuencia del proceso estacionario. Un método paramétrico para la estimación del espectro en potencia es el bien conocido método de *máxima entropía* (MEM, por sus siglas en inglés) que se presenta en el apéndice F.

3.6. Estimación del canal PLC

En la literatura especializada se encuentran trabajos donde se estudia la aplicación de diversos esquemas de estimación de canal asistidos por pilotos para los sistemas OFDM empleados en la línea eléctrica, tal y como está reportado en [39], mientras que otros investigadores han estudiado la aplicación de técnicas de estimación del canal regidas por decisión (DD) o de estimación ciega a los sistemas PLC, como en [133]. Usualmente, los transreceptores PLC emplean símbolos piloto en la primera estimación del canal para, después, continuar con esta técnica o emplear otra. De hecho, debido a que el canal de la línea eléctrica muestra variaciones significativas, tanto en el tiempo como en la frecuencia, es conveniente un procedimiento de seguimiento del canal soportado por una estrategia regida por decisión o, incluso, estimación ciega a fin de mejorar el desempeño de estas aplicaciones OFDM.

En algunos casos, relevantes por su enfoque práctico, se proponen técnicas de estimación del canal PLC relativamente simples pero funcionales. Por ejemplo, en [40] se emplea el método de estimación del canal PLC mediante una técnica asistida por pilotos y se mejora por eliminación de ruido mediante la técnica de ventaneo de la respuesta al impulso descrita en la sección 3.4.3.

3.7. Conclusiones del capítulo

De todos los métodos de estimación del canal para sistemas OFDM, los más populares son los estimadores lineales que emplean la información proporcionada por símbolos pilotos a fin de aproximar a la función de transferencia del canal. Entre ellos destacan el método de mínimos cuadrados (*Least Squares Estimator, LS*) y el método de minimización del error lineal cuadrático promedio (*Linear Minimum Mean Square Error Estimator, LMMSE*). Específicamente, estas técnicas también han sido ampliamente usadas y adaptadas al ámbito PLC.

Capítulo 4

Conceptos básicos sobre la transformada de ondeleta y su aplicación a la estimación

En este capítulo se presenta una breve revisión de la teoría de la transformada de ondeleta discreta y su aplicación a la estimación. El lector podrá referirse a las obras de I. Daubechies [43], M. Holschneider [68], S. Mallat [97], Y. Meyer [99] y Vidakovic [136] para una descripción más detallada de estas herramientas de análisis.

4.1. Evolución de las ondeletas

Desde poco más de veinte años las ondeletas han tenido gran éxito en el procesamiento digital. Estas herramientas permiten descomponer las señales y, a partir de los coeficientes obtenidos, efectuar análisis no lineales. También se puede extender esta transformada al caso multidimensional donde, usualmente, las ondeletas n -dimensionales resultantes son el *producto tensorial* de ondeletas unidimensionales. La transformada así obtenida es separable. Por otro lado, se puede llevar a la práctica en forma simple y eficaz, utilizando parejas de filtros (banco de filtros diádicos). Sin embargo, también han sido utilizados otros bancos de filtros, particularmente los de M -bandas. Finalmente, al iterar recursivamente estos bancos de filtros, podemos llegar a un análisis de multi-resolución que permite examinar la señal a diferentes escalas. Todo esto ha derivado en métodos que actualmente están entre los más competitivos para diferentes procesamientos tales como eliminación de ruido, la restauración o la compresión.

Sin embargo, un punto débil de las descomposiciones con bases de ondeletas es que no son invariantes a traslados. Este problema puede ser resuelto utilizando los denominados *árboles duales de ondeletas complejas*, pero con el costo de una mayor redundancia en la descomposición. También se han desarrollado ondeletas específicas para el procesamiento de imágenes, pero aunque algunas de ellas manejan bien las singularidades, tales como los contornos, no necesariamente trabajan bien con los datos texturizados. En el caso de que estén presentes simultáneamente tanto contornos como texturas, son utilizados comúnmente métodos no separables o direccionales. Generalmente, estas novedosas transformadas son redundantes y de compleja realización, pero los

volúmenes de datos analizados son a menudo grandes y puede resultar evidente la necesidad de desarrollar transformadas menos costosas.

Lo ideal es establecer un equilibrio entre la complejidad del método, su resultado, y su robustez. Es en este contexto que fue desarrollada la transformada diádica de árbol dual, a finales de los años 1990, y desde entonces ha adquirido una popularidad creciente. Las realizaciones de esta transformada tienen fundamento en bancos de filtros digitales, lo que le permite conservar todas las ventajas de las descomposiciones de las ondeletas convencionales pero con una realización sencilla. Esta nueva herramienta de análisis, introducida por N. Kingsbury, luego fue objeto de contribuciones posteriores por parte de otros investigadores tales como I. Selesnick o R. Baraniuk [128]. Esta transformada, que hereda de las propiedades de las ondeletas, constituye lo que en matemáticas se denomina *marco (frame)* y constituye un ejemplo de los más recientes avances en el campo de las ondeletas.

Las transformaciones tienen como objetivo modificar la descripción de una señal descomponiéndola en “elementos de información” que, en general, son más fáciles de interpretar. Estos elementos que representan la señal en la salida de la transformación (o transformada) están directamente relacionados a coeficientes que normalmente constituyen el objeto de estudio en un nuevo dominio (dominio de las ondeletas).

El empleo de bases de ondeletas es particularmente apropiado para la descomposición de señales unidimensionales. En efecto, la capacidad de poder representar una señal a diferentes resoluciones es un aspecto importante en diversas tareas de compresión, de eliminación de ruido y de análisis. En este trabajo, nos enfocaremos en el problema de la descomposición de señales unidimensionales dado que la estimación que nos aqueja comprende únicamente series de datos.

En lo que resta de esta sección 4.1, donde se revisa la evolución de las ondeletas, emplearemos la notación referida en el Cuadro 4.1, tal y como lo hace Caroline Chaux en [32]. Esto lo hacemos por comodidad, debido a que los últimos avances en el campo de las ondeletas incluyen la transformada de ondeleta compleja de árbol dual (DT-CWT, por sus siglas en inglés), que emplea un par de transformadas de ondeleta discreta (DWT) de doble banda [128], y considerando que estas transformadas fueron extendidas recientemente a M-bandas por Caroline Chaux. Entonces, en esta revisión emplearemos la notación de Chaux en vista de la dificultad de adaptar la notación tradicional para el caso de M-bandas. Observe cómo en la notación de Chaux la función de escalamiento se distingue de las funciones de ondeleta por el sub-índice (0 para la función de escalamiento y cualquier otro para las ondeletas) y no por el empleo de la letra φ o ψ , respectivamente. Lo mismo aplica para los coeficientes de descomposición. No obstante, en las secciones restantes retomaremos la notación típicamente empleada en la literatura de las ondeletas.

Cuadro 4.1: Notación empleada en la sección 4.1

Notación	significado
M :	Numero de bandas de los bancos de filtros.
$\mathbb{N}_M =$	$\{0, \dots, M - 1\}$.
$\mathbb{N}_M^* =$	$\{1, \dots, M - 1\}$.
$m \in \mathbb{N}_M, h_m[k]$:	Respuesta al impulso de un filtro.
$m \in \mathbb{N}_M, H_m[\omega]$:	Respuesta en frecuencia de un filtro.
$m \in \mathbb{N}_M, H_m^*[\omega]$:	Respuesta en frecuencia conjugada de un filtro.
$m \in \mathbb{N}_M, \tilde{H}_m^*[\omega]$:	Filtro de análisis en una descomposición biortogonal.
ψ_0 :	Ondeleta padre o función de escalamiento.
$m \in \mathbb{N}_M^*, \psi_m$:	Ondeleta madre.
\hat{a} :	Transformada de Fourier de una función a .
$c_{l,m}[k]$:	Coficientes c resultantes de una transformación a un nivel l dentro de la sub-banda m , en la posición espacial k .
\mathbb{Z} :	Espacio de enteros.
\mathbb{N} :	Espacio de los naturales.
\mathbb{N}^* :	Espacio de los naturales no nulos.
\mathbb{R} :	Espacio de los reales.
\mathbb{R}^* :	Espacio de los reales no nulos.
\mathbb{R}_+ :	Espacio de los reales positivos.
\mathbb{R}_+^* :	Espacio de los reales positivos no nulos.
$\delta_{m,n}$:	Delta de Kronecker.
δ :	Delta de Dirac.

4.1.1. Transformada de ondeleta continua

La ondeleta ψ_1 es una función de $L^2(\mathbb{R})$ integrable de media cero (presuponiendo valores reales) que oscila localmente:

$$\int_{-\infty}^{+\infty} \psi_1(t) dt = 0, \quad (4.1)$$

que puede ser dilatada/contraída (mediante el factor de escala $e \in \mathbb{R}_+^*$) y desplazada (mediante el parámetro de localización $u \in \mathbb{R}$):

$$\psi_{u,e}(t) = \frac{1}{\sqrt{e}} \psi_1\left(\frac{t-u}{e}\right) \quad (4.2)$$

La transformada de ondeleta continua de la señal f (con una ondeleta dilatada/contraída en e y desplazada en u) está dada por:

$$W_f(e, u) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{e}} \psi_1^*\left(\frac{t-u}{e}\right) dt = \langle f, \psi_{u,e} \rangle \quad (4.3)$$

La señal f puede ser reconstruida a partir de $W_f(e, u)$ empleando la relación siguiente:

$$f(t) = \frac{1}{C_{\psi_1}} \int_0^{+\infty} \int_{-\infty}^{+\infty} W_f(e, u) \frac{1}{\sqrt{e}} \psi_1\left(\frac{t-u}{e}\right) du \frac{de}{e^2} \quad (4.4)$$

con

$$C_{\psi_1} = \int_0^{+\infty} \frac{|\psi_1(\omega)|^2}{\omega} d\omega. \quad (4.5)$$

donde la constante C_{ψ_1} debe ser finita. Esto es lo que se llama *condición de admisibilidad*, que implica que la ondeleta sea de media cero. La descomposición en integrales de (4.4) permite reconstruir f , empleando ondeletas correspondiente a todas las localizaciones temporales y a todos los factores de escala.

4.1.2. Análisis multi-resolución

Usualmente es más cómodo disponer del desarrollo de la señal sobre un conjunto finito de ondeletas. Esto puede ser realizado variando el factor de escalamiento de un modo diádico, escogiendo $e = 2^j$, $j \in \mathbb{Z}$ y $u = k2^j$, $k \in \mathbb{Z}$. En este contexto, un análisis multi-resolución se define como una extensión de los subespacios vectoriales cerrados $(V_j)_{j \in \mathbb{Z}}$ de $L^2(\mathbb{R})$ que cumple las siguientes 4 propiedades:

1.

$$\forall j \in \mathbb{Z}, \quad V_{j+1} \subset V_j,$$

2.

$$\forall j \in \mathbb{Z}, \quad f(t) \in V_j \Leftrightarrow f\left(\frac{t}{2}\right) \in V_{j+1},$$

3.

$$\lim_{j \rightarrow +\infty} V_j = \bigcap_{j \in \mathbb{Z}} V_j = 0,$$

y

$$\lim_{j \rightarrow +\infty} V_j = \text{cerradura} \left(\bigcup_{j \in \mathbb{Z}} V_j \right) = L^2(\mathbb{R}).$$

4. Existe una ψ_0 tal que $\{\psi_0(t-n), n \in \mathbb{Z}\}$ es una base ortonormal de V_0 ; ψ_0 es llamada *función de escalamiento*.

También se cumple que $\{2^{-j/2}\psi_0(t/2^j - k), k \in \mathbb{Z}\}$ forma una base ortonormal de V_j .

El análisis multi-resolución de una señal f consiste en realizar proyecciones ortogonales sucesivas de la señal sobre los espacios V_j , lo que deriva en aproximaciones más y más burdas de f a medida que j crece. La diferencia entre dos aproximaciones consecutivas representa la información del *detalle* que se pierde en el paso de una escala al otra; esta información se contiene en el subespacio W_j , ortogonal a V_j , tal que:

$$V_{j-1} = V_j \oplus W_j.$$

Se puede demostrar que existe una ondeleta $\psi_1 \in L^2(\mathbb{R})$ tal que $\{2^{-j/2}\psi_1(t/2^j - k), k \in \mathbb{Z}\}$ es una base ortonormal de W_j . La descomposición por ondeletas ortogonales de una señal f se puede realizar de manera muy eficiente [97] si determinamos en cada nivel de resolución $j \in \mathbb{Z}$ los coeficientes de aproximaciones $(c_{j,0}[k])_{k \in \mathbb{Z}}$ y de detalles $(c_{j,1}[k])_{k \in \mathbb{Z}}$ definidos por:

$$\forall k \in \mathbb{Z}, \quad c_{j,0}[k] = \left\langle f, \frac{1}{2^{j/2}}\psi_0\left(\frac{t}{2^j} - k\right) \right\rangle \quad (4.6)$$

$$c_{j,1}[k] = \left\langle f, \frac{1}{2^{j/2}}\psi_1\left(\frac{t}{2^j} - k\right) \right\rangle \quad (4.7)$$

donde $\langle \cdot, \cdot \rangle$ designa el producto escalar en $L^2(\mathbb{R})$.

4.1.3. Noción de marco

La noción de los marcos (*frames*) fue introducida en [52] con el fin de obtener representaciones (descomposiciones) mejor adaptadas a una señal, así como para minimizar los problemas invarianza por traslado y de repliegue de espectros, al proporcionar representaciones redundantes de las señales.

Consideremos un espacio de Hilbert \mathcal{H} e introduzcamos $\mathbb{K} \subset \mathbb{N}$. Una familia de vectores $(e_k)_{k \in \mathbb{K}}$ de \mathcal{H} es un marco si existen dos constantes μ y ν pertenecientes a $(0, +\infty)$ tales que:

$$\forall f \in \mathcal{H}, \quad \mu \|f\|^2 \leq \sum_{k \in \mathbb{K}} |\langle f, e_k \rangle|^2 \leq \nu \|f\|^2. \quad (4.8)$$

En el caso de $\mu = \nu$, decimos que el marco es del tipo “ajustado” de límite μ . Un ejemplo simple de marco ajustado es la unión de N bases ortonormales. Tenemos entonces $\mu = \nu = N$.

Ya que es posible construir marcos con ondeletas [42], en años recientes, muchos investigadores se han dedicado a la construcción de marcos que permiten el análisis de los datos en forma más adecuada que las transformadas de ondeletas originales.

4.1.4. Generalización a M-bandas

Se pueden extender los resultados del caso diádico (parejas de filtros para el análisis/síntesis) al caso de M-bandas (grupos de filtros para el análisis/síntesis) si hacemos a M un entero mayor o igual a 2 y definimos $\mathbb{N}_M = \{0, \dots, M-1\}$ y $\mathbb{N}_M^* = \{1, \dots, M-1\}$. En lo subsiguiente, \widehat{a} designa a la transformada de Fourier de una función a y $(\delta_{m,n})_{m \in \mathbb{Z}}$ es la secuencia de Kronecker (igual a 1 si $m = n$ y 0 en otro caso).

Deseamos que el factor de escala no varíe de modo diádico pero lo elegimos de tal forma que $e = M^j$. Notemos que en las ecuaciones siguientes obtenemos el caso diádico si hacemos $M = 2$. Partiendo de espacios V_j y $(W_j^m)_{m \in \mathbb{N}_M^*}$ definidos a un nivel de resolución j , tenemos que están ligados por la siguiente relación [32]:

$$V_{j-1} = V_j \oplus \bigoplus_{m=1}^{M-1} W_j^m. \quad (4.9)$$

Por otra parte, V_j (o bien W_j^m) admite una base ortonormal $\{M^{-j/2}\psi_0(t/M^j - k), k \in \mathbb{Z}\}$ (o bien $\{M^{-j/2}\psi_m(t/M^j - k), k \in \mathbb{Z}\}$). Se puede apreciar que en un análisis multi-resolución de M-bandas en $L^2(\mathbb{R})$ se emplea una función de escala (u ondeleta padre) $\psi_0 \in L^2(\mathbb{R})$ y $(M-1)$ ondeletas a madre $\psi_m \in L^2(\mathbb{R})$, $m \in \mathbb{N}_M^*$ [132]. Estas funciones son soluciones a las ecuaciones siguientes a 2 escalas de resolución:

$$\forall m \in \mathbb{N}_M, \quad \frac{1}{\sqrt{M}}\psi_m\left(\frac{t}{M}\right) = \sum_{k=-\infty}^{\infty} h_m[k]\psi_0(t-k), \quad (4.10)$$

donde las secuencias $(h_m[k])_{k \in \mathbb{Z}}$ son de cuadrados integrables. En lo subsiguiente, supondremos que estas funciones (así como las secuencias asociadas $(h_m[k])_{k \in \mathbb{Z}}$) son reales. La transformada de Fourier de $(h_m[k])_{k \in \mathbb{Z}}$ es una función periódica en 2π denotada H_m :

$$\forall \omega \in [0, 2\pi), \quad H_m(\omega) = \sum_{k=-\infty}^{\infty} h_m[k]e^{-ik\omega}. \quad (4.11)$$

Las ecuaciones a 2 escalas de resolución en (4.10) pueden entonces expresarse en el dominio de la frecuencia:

$$\forall m \in \mathbb{N}_M, \quad \sqrt{M}\widehat{\psi}_m(M\omega) = H_m(\omega)\widehat{\psi}_0(\omega). \quad (4.12)$$

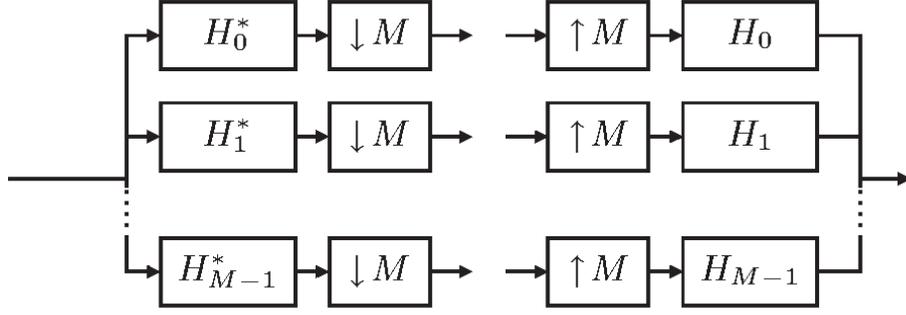


Figura 4.1: Un banco de filtros de M-bandas

Para que $\bigcup_{m=1}^{M-1} \{M^{j/2}\psi_m(M^{-j}t-k)(j,k) \in \mathbb{Z}^2\}$ forme una base ortonormal en $L^2(\mathbb{R})$, se debe cumplir la condición siguiente:

$$\forall(m, m') \in \mathbb{N}_M^2, \quad \sum_{p=0}^{M-1} H_m(\omega + p\frac{2\pi}{M})H_{m'}^*(\omega + p\frac{2\pi}{M}) = M\delta_{m,m'}. \quad (4.13)$$

El filtro de respuesta en frecuencia H_0 es pasa bajos mientras que los filtros de respuestas en frecuencia H_m , $m \in \{1, \dots, M-2\}$ (o bien $m = M-1$) son pasa bandas (o bien pasa-altos). En ese caso, los bancos de filtros M-bandas en cascada para el análisis y la síntesis que se representan en la Figura 4.1 permiten descomponer y reconstruir perfectamente una señal dada.

Análogamente al caso diádico, los coeficientes resultantes para una señal f se definen como:

$$\forall m \in \mathbb{N}_M, \quad c_{j,m}[k] = \left\langle f, \frac{1}{M^{j/2}}\psi_m\left(\frac{t}{M^j} - k\right) \right\rangle. \quad (4.14)$$

La transformada en ondeletas M-bandas también se puede llevar a la práctica con la ayuda de bancos de filtros de M-bandas. El tipo de banco de filtros utilizado es ilustrado en la Figura 4.1. El análisis multi-resolución de M-bandas se realiza colocando en cascada estos bancos de filtros de tal suerte que se repita la descomposición de los coeficientes aproximación (ver Figura 4.2).

El hecho de considerar bancos de filtros de M-bandas con $M > 2$, permite obtener una descomposición más precisa en frecuencia. Además, esto permite una mayor libertad en la elección de los filtros. Por ejemplo, es posible generar ondeletas simétricas, ortogonales y de soporte compacto [132], mientras que en el caso diádico esto es posible únicamente para la base de Haar.

4.1.5. Ondeletas biortogonales

En ocasiones puede ser conveniente utilizar ondeletas de soporte compacto que posean propiedades de simetría. Para esto, la condición de ortogonalidad debe ser relajada. Entonces se utilizan

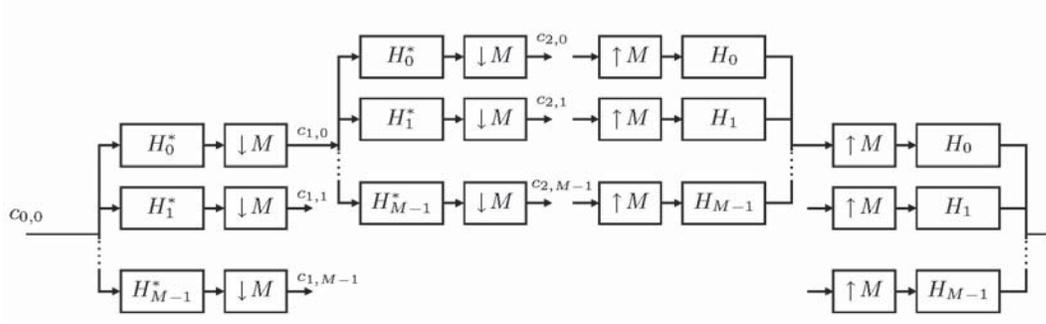


Figura 4.2: Análisis/síntesis de M-bandas sobre 2 niveles de resolución

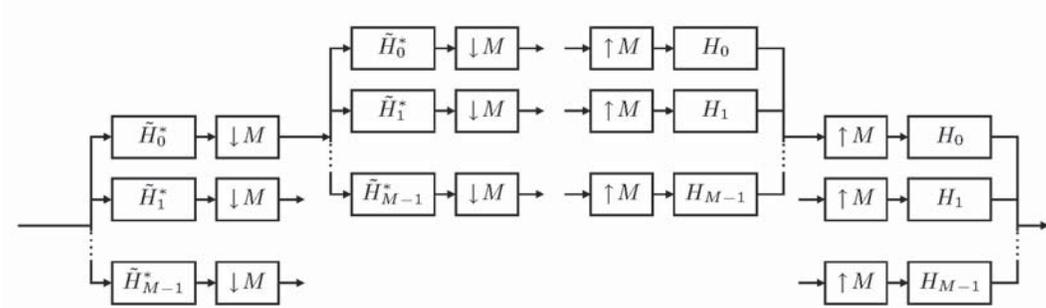


Figura 4.3: Análisis/síntesis multi-resolución M-bandas biortogonales sobre 2 niveles de resolución

ciertas ondeletas para el análisis y otras para la síntesis, en un esquema denominado como de ondeletas biortogonales [35]. En este caso, dos bases de ondeletas de $L^2(\mathbb{R})$ (no ortonormales) son consideradas en forma conjunta:

$$\bigcup_{m \in \mathbb{N}_M^*} \{M^{-j/2} \psi_m(t/M^j - k), \quad (j, k) \in \mathbb{Z}^2\}$$

y

$$\bigcup_{m \in \mathbb{N}_M^*} \{M^{-j/2} \tilde{\psi}_m(t/M^j - k), \quad (j, k) \in \mathbb{Z}^2\}.$$

Para calcular los coeficientes de esta descomposición, procedemos igual que antes. Los bancos de filtros correspondientes al análisis y a la síntesis se representan en la Figura 4.3.

Utilizamos entonces filtros de respuestas en frecuencias $(\tilde{H}_m^*)_m$ y $(H_m)_m$ para el análisis y para la síntesis, respectivamente. Las respuestas en frecuencias de estos filtros deben cumplir:

$$\forall (m, m') \in \mathbb{N}_M^2, \quad \sum_{p=0}^{M-1} H_m(\omega + p\frac{2\pi}{M}) \tilde{H}_{m'}^*(\omega + p\frac{2\pi}{M}) = M\delta_{m,m'}. \quad (4.15)$$

con el fin de que haya reconstrucción perfecta.

4.1.6. Descomposición de paquete de ondeletas

La transformada de paquete de ondeletas [36] nació de la necesidad de adaptarse a las características de tiempo-frecuencia de las señales. La transformada de paquete de ondeletas permite la descomposición recurrente de las sub-bandas de alta frecuencia, en contraparte a la transformada de ondeletas discreta clásica la cual sólo repite el proceso sobre la sub-banda de bajas frecuencias. Definimos las funciones $(\psi_m)_{m \in \mathbb{N}}$ de manera siguiente:

$$\forall m \in \mathbb{N}, \quad \sqrt{2}\widehat{\psi}_{2m}(2\omega) = A_0(\omega)\widehat{\psi}_m(\omega) \quad (4.16)$$

$$\sqrt{2}\widehat{\psi}_{2m+1}(2\omega) = A_1(\omega)\widehat{\psi}_m(\omega) \quad (4.17)$$

donde A_0 y A_1 son las respuestas en frecuencias de los filtros pasa-bajos y pasa-altos asociados al banco de filtros de banda dual para síntesis.

Denotemos a W_j^m el subespacio vectorial cuya base es $\{2^{-j/2}\psi_m(t/2^j - k), \quad k \in \mathbb{Z}\}$. Al nivel de resolución $j + 1$, se obtiene a las familias

$$\{2^{-(j+1)/2}\psi_{2m}(t/2^{j+1} - k), \quad k \in \mathbb{Z}\}$$

y

$$\{2^{-(j+1)/2}\psi_{2m+1}(t/2^{j+1} - k), \quad k \in \mathbb{Z}\}$$

que son las bases ortonormales correspondientes a los espacios vectoriales W_{j+1}^{2m} y W_{j+1}^{2m+1} , que cumplen:

$$W_j^m = W_{j+1}^{2m} \oplus W_{j+1}^{2m+1}.$$

Las ondeletas de M-bandas se pueden generar a partir de descomposiciones de paquetes de ondeletas diádicos correspondientes a un análisis uniforme en sub-bandas iguales [15], lo que significa que todas las sub-bandas obtenidas son nuevamente descompuestas sobre P niveles de resolución (ver Figura 4.4). Debido a esto es que el valor de M está limitado a una potencia de 2. Específicamente, la función de escalamiento y las ondeletas asociadas para M-bandas se definen por $(\psi_m)_{0 \leq m < M}$ con $M = 2^P$, $P \in \mathbb{N}^*$. Estas ondeletas cumplen las ecuaciones (4.16) y (4.17).

Generalizando la ecuación a 2 escalas de resolución para el caso diádico ((4.12), con $M = 2$), se obtiene la ecuación siguiente:

$$\widehat{\psi}_m(\omega) = \widehat{\psi}_0\left(\frac{\omega}{2^P}\right) \prod_{i=1}^r \left(\frac{1}{\sqrt{2}}A_{\epsilon_i}\left(\frac{\omega}{2^i}\right)\right) \prod_{i=r+1}^P \left(\frac{1}{\sqrt{2}}A_0\left(\frac{\omega}{2^i}\right)\right) \quad (4.18)$$

donde para toda $m \in \mathbb{N}_M^*$, $(\epsilon_1, \epsilon_2, \dots, \epsilon_r) \in \{0, 1\}^r$, $r \in \mathbb{N}_M^*$, son los valores de la descomposición de m en base 2, o sea:

$$m = \sum_{i=1}^r \epsilon_i 2^{i-1}. \quad (4.19)$$

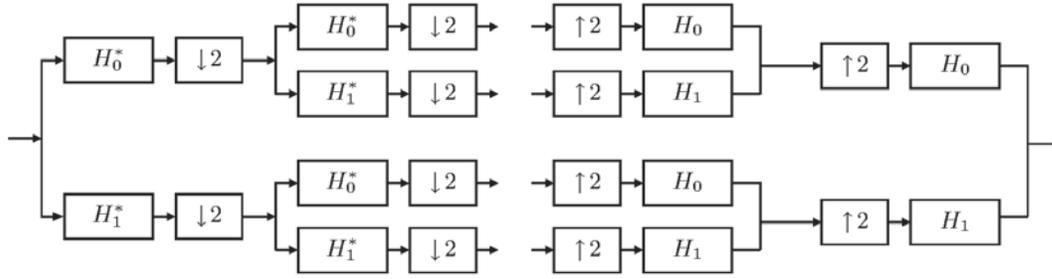


Figura 4.4: Ejemplo de descomposición uniforme por paquete de ondeletas (análisis en sub-bandas iguales)

Finalmente, en este caso las respuestas en frecuencias de los filtros asociados de M-bandas están definidos por:

$$H_m(\omega) = \prod_{i=1}^r A_{\epsilon_i}(2^{P-i}\omega) \prod_{i=r+1}^P A_0(2^{P-i}\omega) \quad (4.20)$$

4.2. Relaciones elementales

Consideremos que la escala unitaria esta compuesta por la función padre ψ_0 que en lo sub-siguiente denominaremos como φ , para simplificar la notación. Se trata de una función de corta duración (es decir con un ancho o soporte finito) y sus desplazadas $\varphi_u = \varphi(t - u)$. Dicha función es la función de escalamiento y es tal que:

$$\langle f, \varphi_0 \rangle = \int_{-\infty}^{+\infty} f(t)\varphi^*(t) \approx f(t_0) \quad (4.21)$$

donde t_0 es el centro del soporte de $\varphi(t)$. Si se considera la desplazada $\varphi_u = \varphi(t - u)$, entonces se pueden efectuar mediciones en cualquier posición u .

Además, se cumple que $\langle \varphi_m, \varphi_n \rangle = \delta_{m,n}$, es decir la escala unitaria V_0 forma un espacio vectorial con base $\{\varphi_0, \dots, \varphi_n\}$. Entre más angosta sea $\varphi(t)$ mayor es la resolución. A la operación de escalamiento de $\varphi(t)$ se le conoce también como dilatación y la definiremos para el caso diádico como:

$$\mathbf{D}\varphi(t) = \frac{1}{\sqrt{2}}\varphi\left(\frac{t}{2}\right)$$

Sean V_j los espacios vectoriales derivados de V_0 , $V_j = \mathbf{D}^j V_0$ con las bases ortogonales

$$\{\varphi_{j,0}, \dots, \varphi_{j,n}\},$$

donde $\varphi_{j,n} = \mathbf{D}^j \mathbf{T}^n \varphi$, tales que las V_j satisfacen entre sí

$$V_{j+1} \subset V_j.$$

Así, por inclusión $V_1 \subset V_0$, por lo que $\mathbf{D}\varphi \in V_0$ y puede ser representado por la base $\{\varphi_0, \dots, \varphi_m\}$, entonces

$$\mathbf{D}\varphi = \sum_m h_m \varphi_m \quad (4.22)$$

Por otra parte, sea W_1 , con base $\{\psi_0, \dots, \psi_m\}$, el complemento ortogonal de V_1 respecto de V_0 , entonces se cumple:

$$\mathbf{D}\psi = \sum_m g_m \varphi_m \quad (4.23)$$

También se cumple que $W_{j+1} \subset V_j$, lo que implica que $V_j = V_{j+1} \oplus W_{j+1}$, entonces tenemos dos bases ortogonales para V_j :

$$\{\varphi_{j,0}, \dots, \varphi_{j,n}\}$$

y

$$\{\varphi_{j+1,0}, \dots, \varphi_{j+1,k}\} \cup \{\psi_{j+1,0}, \dots, \psi_{j+1,k}\}$$

En general, si $f \in V_j$, se cumple que

$$f = \sum_n c_n \varphi_{j,n} \quad (4.24)$$

pero también

$$f = \sum_k a_k \varphi_{j+1,k} + \sum_k d_k \psi_{j+1,k} \quad (4.25)$$

donde c_n , a_k y d_k son las proyecciones de f en V_j , V_{j+1} y W_{j+1} , respectivamente. Es decir, que los coeficientes c_n contienen la información de la aproximación de f en la escala j , mientras que a_k contienen la aproximación de f en la escala $j+1$ y d_k contienen el detalle de f en la escala $j+1$. Estos coeficientes se obtienen mediante los siguientes productos escalares:

$$c_n = \langle \varphi_{j,n}, f \rangle \quad (4.26)$$

$$a_k = \langle \varphi_{j+1,k}, f \rangle \quad (4.27)$$

$$d_k = \langle \psi_{j+1,k}, f \rangle \quad (4.28)$$

Sustituyendo (4.25) en (4.26) y recordando que $\varphi_{j,m} = \mathbf{D}^j \mathbf{T}^m \varphi$, que $\psi_{j,m} = \mathbf{D}^j \mathbf{T}^m \psi$, y que en el

caso diádico $\mathbf{D}\mathbf{T}^n = \mathbf{T}^{2n}\mathbf{D}$ tenemos [13]:

$$\begin{aligned}
c_n &= \left\langle \mathbf{D}^j \mathbf{T}^n \varphi, \sum_k a_k \varphi_{j+1,k} + \sum_k d_k \psi_{j+1,k} \right\rangle \\
&= \left\langle \mathbf{D}^j \mathbf{T}^n \varphi, \sum_k a_k \mathbf{D}^{j+1} \mathbf{T}^k \varphi + \sum_k d_k \mathbf{D}^{j+1} \mathbf{T}^k \psi \right\rangle \\
&= \left\langle \mathbf{D}^j \mathbf{T}^n \varphi, \sum_k a_k \mathbf{D}^{j+1} \mathbf{T}^k \varphi \right\rangle + \left\langle \mathbf{T}^n \varphi, \sum_k d_k \mathbf{D}^{j+1} \mathbf{T}^k \psi \right\rangle \\
&= \left\langle \mathbf{D}^j \mathbf{T}^n \varphi, \sum_k a_k \mathbf{D} \mathbf{D}^j \mathbf{T}^k \varphi \right\rangle + \left\langle \mathbf{T}^n \varphi, \sum_k d_k \mathbf{D} \mathbf{D}^j \mathbf{T}^k \psi \right\rangle \\
&= \sum_k a_k \langle \mathbf{T}^n \varphi, \mathbf{D} \mathbf{T}^k \varphi \rangle + \sum_k d_k \langle \mathbf{T}^n \varphi, \mathbf{D} \mathbf{T}^k \psi \rangle \\
&= \sum_k a_k \langle \varphi_n, \mathbf{T}^{2k} \mathbf{D} \varphi \rangle + \sum_k d_k \langle \varphi_n, \mathbf{T}^{2k} \mathbf{D} \psi \rangle
\end{aligned}$$

Ahora, sustituyendo (4.22) y (4.23), tenemos

$$\begin{aligned}
c_n &= \sum_k a_k \left\langle \varphi_n, \mathbf{T}^{2k} \sum_m h_m \varphi_m \right\rangle + \sum_k d_k \left\langle \varphi_n, \mathbf{T}^{2k} \sum_m g_m \varphi_m \right\rangle \\
&= \sum_k a_k \left\langle \varphi_n, \sum_m h_m \varphi_{m+2k} \right\rangle + \sum_k d_k \left\langle \varphi_n, \sum_m g_m \varphi_{m+2k} \right\rangle \\
&= \sum_k a_k \sum_m h_m \langle \varphi_n, \varphi_{m+2k} \rangle + \sum_k d_k \sum_m g_m \langle \varphi_n, \varphi_{m+2k} \rangle
\end{aligned}$$

Pero como se cumple que $\langle \varphi_m, \varphi_n \rangle = \delta_{m,n}$, tenemos

$$c_n = \sum_k a_k \sum_m h_m \delta_{n,m+2k} + \sum_k d_k \sum_m g_m \delta_{n,m+2k}$$

y ya que $\delta_{n,m+2k} = 1$ cuando $m = n - 2k$, esto implica que

$$c_n = \sum_k a_k h_{n-2k} + \sum_k d_k g_{n-2k} \quad (4.29)$$

Similarmente, sustituyendo (4.24) en (4.27) y en (4.28), por la propiedad de ortogonalidad de las bases se llega a:

$$a_k = \sum_n c_n h_{n-2k} \quad (4.30)$$

y

$$d_k = \sum_n c_n g_{n-2k} \quad (4.31)$$

Observamos que las proyecciones no dependen de la escala, sino únicamente de los coeficientes h_m y g_m , lo que simplifica mucho al sistema. La ventaja de separar una señal en bandas de frecuencia consiste en que cada una de estas bandas, al ser obtenidas por separación ortogonal, no están correlacionadas, por lo que se pueden diseñar técnicas convenientes de procesamiento digital en cada una de estas bandas como, por ejemplo, en las técnicas de eliminación de ruido.

4.3. Transformada de ondeleta discreta

La transformada de ondeleta discreta (DWT, por sus siglas en inglés) se aplica a conjuntos discretos de datos y produce una salida discreta. Transformar señales y vectores de datos mediante una DWT es un proceso que recuerda la transformada rápida de Fourier (FFT), es decir un método rápido para la transformada de Fourier aplicado a un conjunto discreto de mediciones.

La transformada de ondeleta discretas mapea datos del dominio del tiempo (el vector de datos original o de entrada) al dominio de la ondeleta. El resultado es un vector del mismo tamaño. Las transformadas de ondeleta son lineales y pueden ser definidas por matrices de dimensiones $n \times n$, si se aplican a una entrada de tamaño n . Dependiendo de las condiciones de frontera, tales matrices pueden ser ortogonales o aproximadamente ortogonales. Cuando la matriz es ortogonal, la transformación correspondiente es una rotación en \mathbb{R}^n de tal manera que el vector con los datos de entrada representa las coordenadas de un punto. Las coordenadas del punto en el espacio rotado corresponden a la transformada de ondeleta discreta de las coordenadas originales.

4.3.1. El algoritmo en cascada

En el contexto del procesamiento de imágenes, Burt y Anderson [22] desarrollaron los algoritmos piramidales ortogonales y biortogonales. Los métodos de pirámide o cascada procesan una imagen a diferentes escalas, clasificadas de lo fino a lo grueso, en un algoritmo arbolado. Las imágenes pueden ser depuradas (eliminación del ruido), resaltadas o comprimidas mediante apropiados procesamientos a nivel de las escalas.

Mallat [96] fue el primero en ligar las ondeletas, el análisis multi-resolución y los algoritmos de cascada de manera formal. El algoritmo de cascada de Mallat proporciona una receta constructiva y eficiente para realizar una transformación de ondeleta discreta. Este algoritmo relaciona los coeficientes de ondeleta, en los diferentes niveles de la transformación, con el filtrado mediante los filtros \underline{h} y \underline{g} . Por lo tanto, el algoritmo de Mallat puede ser visto como la contraparte para ondeletas del algoritmo de Danielson-Lanczos para la transformada rápida de Fourier.

Es conveniente ligar los datos originales con el espacio V_J , típicamente V_0 . Entonces los espacios de aproximación gruesa (suavizado) y los complementarios con el detalle son (V_{J+1}, W_{J+1}) , (V_{J+2}, W_{J+2}) , etc. Aquí, incrementar el índice de los espacios V es equivalente a hacer más gruesa la aproximación a los datos.

De (4.22) se deriva

$$\varphi(t) = \sum_{m \in \mathbb{Z}} h_m \sqrt{2} \varphi(2t - m) \quad (4.32)$$

que se conoce como ecuación de escalamiento, o ecuación de dos escalas, y que es fundamental para la construcción de las ondeletas.

A su vez, de (4.23) se deriva la función ondeleta a partir de la función de escalamiento:

$$\psi(t) = \sum_{m \in \mathbb{Z}} g_m \sqrt{2} \varphi(2t - m) \quad (4.33)$$

Ahora, por sustitución directa de índices en la ecuación de escalamiento (4.32) y (4.33), se obtiene

$$\varphi_{j+1,k} = \sum_{n \in \mathbb{Z}} h_{n-2k} \varphi_{j,k}(t) \quad (4.34)$$

y

$$\psi_{j+1,k} = \sum_{n \in \mathbb{Z}} g_{n-2k} \varphi_{j,k}(t) \quad (4.35)$$

Las expresiones (4.34) y (4.35) son fundamentales en el desarrollo de los algoritmos en cascada.

Considerando al análisis multi-resolución $\dots \subset V_{j+1} \subset V_j \subset V_{j-1} \subset \dots$, ya que $V_j = V_{j+1} \oplus W_{j+1}$, cualquier función $v_j \in V_j$ puede ser representada en forma única como $v_j(t) = v_{j+1}(t) + w_{j+1}(t)$ donde $v_{j+1} \in V_{j+1}$ y $w_{j+1} \in W_{j+1}$. Denotando $a_{j,k}$ a los coeficientes asociados a $\varphi_{j,k}$ y $d_{j,k}$ a los coeficientes asociados a $\psi_{j,k}$, entonces, por sustitución directa de índices en la ecuaciones (4.30) y (4.31),

$$a_{j+1,k} = \sum_n c_{j,n} h_{n-2k} \quad (4.36)$$

y

$$d_{j+1,k} = \sum_n c_{j,n} g_{n-2k} \quad (4.37)$$

El algoritmo de cascada funciona también en sentido inverso. Los coeficientes en la siguiente escala más fina, correspondientes a V_j , pueden ser obtenidos a partir de los coeficientes correspondientes a V_{j+1} y W_{j+1} . Así, la expresión

$$a_{j,n} = \sum_k a_{j+1,k} h_{n-2k} + \sum_k d_{j+1,k} g_{n-2k} \quad (4.38)$$

representa un paso en el algoritmo de reconstrucción.

4.3.2. Notación de operador de DWT

La notación de operador es una forma elegante para describir el algoritmo de cascada. El operador se puede expresar en términos de matrices o de filtros, pero es más práctico lo último. Para

ello, necesitamos definir las operaciones de decimación, dilatación y convolución en los espacios ℓ^2 .

La *decimación* $[\downarrow 2]$ es un mapeo de $\ell(\mathbb{Z})$ a $\ell(2\mathbb{Z})$ definido en forma de coordenadas como

$$([\downarrow 2]\underline{a})_k = \sum_n a_n \delta_{n-2k} = a_{2k} \quad (4.39)$$

Cuando se aplica la decimación a una secuencia se retiene únicamente a los valores de las posiciones con índices pares. La *dilatación* es un mapeo de $\ell(2\mathbb{Z})$ a $\ell(\mathbb{Z})$ definido como

$$([\uparrow 2]\underline{a})_k = \sum_n a_n \delta_{k-2n} \quad (4.40)$$

que expande la secuencia mediante la inserción de ceros entre los valores originales.

Si empleamos \mathbf{H} y \mathbf{G} para denotar las convoluciones con los filtros \underline{h} y \underline{g} , entonces

$$\mathbf{H} : \ell(\mathbb{Z}) \mapsto \ell(\mathbb{Z}) \quad (\mathbf{H}\underline{a})_k = \sum_n h_{n-k} a_n$$

$$\mathbf{G} : \ell(\mathbb{Z}) \mapsto \ell(\mathbb{Z}) \quad (\mathbf{G}\underline{a})_k = \sum_n g_{n-k} a_n$$

tal que $\mathcal{H} \equiv [\downarrow 2]\mathbf{H}$ y $\mathcal{G} \equiv [\downarrow 2]\mathbf{G}$.

Se puede verificar que para una secuencia $a = \{a_n\}$ en ℓ^2 , los operadores \mathcal{H} y \mathcal{G} satisfacen las siguientes relaciones en forma de coordenadas:

$$(\mathcal{H}a)_k = \sum_n h_{n-2k} a_n$$

$$(\mathcal{G}a)_k = \sum_n g_{n-2k} a_n$$

La aplicación de los operadores \mathcal{H} y \mathcal{G} corresponden a un paso en la transformada de ondeleta discreta.

En forma ilustrativa, asumiremos que la longitud del vector de datos es una potencia de 2. Denotemos la señal original por $\underline{a}^{(0)} = \{a_k^{(0)}\}$. Si la señal es de longitud 2^J , entonces $\underline{a}^{(0)}$ puede ser interpolada por la función $f(t) = \sum_k a_k^{(0)} \varphi(t - k)$ desde V_0 . En cada paso de la transformada de ondeleta, se pasa al siguiente nivel de aproximación más burda $\underline{a}^{(j+1)}$ mediante la aplicación del operador \mathcal{H} , es decir $\underline{a}^{(j+1)} = \mathcal{H}\underline{a}^{(j)}$. La información del detalle, perdida al aproximar $\underline{a}^{(j)}$ con la promediada $\underline{a}^{(j+1)}$, está dada por $\underline{d}^{(j+1)} = \mathcal{G}\underline{a}^{(j)}$.

Entonces, la transformada de ondeleta discreta de una secuencia $\underline{y} = \underline{a}^{(0)}$ de longitud 2^J puede ser representada como

$$(\underline{a}^{(m)}, \underline{d}^{(m)}, \underline{d}^{(m-1)}, \underline{d}^{(m-2)}, \dots, \underline{d}^{(2)}, \underline{d}^{(1)}) \quad (4.41)$$

Note que las longitudes de \underline{y} y su transformada en (4.41) coinciden. Debido a la decimación, la longitud de $\underline{a}^{(j)}$ es el doble que la longitud de $\underline{a}^{(j+1)}$ y $2^J = 2^{J-m} + \sum_{i=1}^m 2^{J-i}$ con $1 \leq m \leq J$, donde m son los niveles de descomposición.

Mediante el empleo de la notación de operador, es posible resumir la transformada de ondeleta discreta, con paro en el m -ésimo nivel de descomposición, en una simple línea:

$$\underline{y} \mapsto (\mathcal{H}^m \underline{y}, \mathcal{G}\mathcal{H}^{m-1} \underline{y}, \mathcal{G}\mathcal{H}^{m-2} \underline{y}, \dots, \mathcal{G}\mathcal{H}^2 \underline{y}, \mathcal{G}\mathcal{H} \underline{y}, \mathcal{G} \underline{y}).$$

El valor para el nivel de descomposición m puede ser, arbitrariamente, cualquier entero entre 1 y J y está asociado con el espacio con suavizado más burdo V_m en el que se interrumpe la transformada. En términos de espacios de multi-resolución, (4.41) corresponde a la descomposición multi-resolución

$$V_m \oplus W_m \oplus W_{m-1} \oplus \dots \oplus W_1.$$

Cuando $m = J$ el vector $\underline{a}^{(J)}$ es un vector de un solo elemento, $a^{(J)}$.

Si la longitud del filtro es mayor a 2 coeficientes, entonces es necesario definir la operación del filtro fuera de las fronteras de la secuencia a la que se aplica el filtro. La ondeleta de Haar no tiene problemas de frontera debido a que el tamaño de su filtro es de sólo 2 coeficientes. Existen diferentes enfoques para resolver el problema de las fronteras. La señal puede ser prolongada más allá de sus fronteras en una forma periódica $(\dots, y_{n-1}, y_n | y_1, y_2, \dots)$, en forma simétrica $(\dots, y_{n-1}, y_n | y_{n-1}, y_{n-2}, \dots)$, rellenas con una constante o extrapoladas por un polinomio. Las transformadas de ondeleta pueden ser confinadas a un intervalo y las extensiones periódicas y simétricas pueden ser vistas como casos especiales.

La fórmula de reconstrucción también es simple en términos de los operadores. Primero se definen a los operadores adjuntos \mathcal{H}^* y \mathcal{G}^* como sigue:

$$(\mathcal{H}^* a)_n = \sum_k h_{n-2k} a_k$$

$$(\mathcal{G}^* a)_n = \sum_k g_{n-2k} a_k$$

Los operadores \mathcal{H}^* y \mathcal{G}^* son aplicados a $\underline{a}^{(j+1)}$ y $\underline{d}^{(j+1)}$, respectivamente, y los resultados se suman. El vector $\underline{a}^{(j)}$ se reconstruye como

$$\underline{a}^{(j)} = \mathcal{H}^* \underline{a}^{(j+1)} + \mathcal{G}^* \underline{d}^{(j+1)} \quad (4.42)$$

que es una re-expresión de (4.38) con operadores. La expresión (4.42) también puede llevarse a la práctica en forma recursiva.

4.4. Ondeletas y la estimación estadística

El modelado estadístico multi-escala se ha convertido en los últimos años en un área floreciente, tanto en la estadística teórica como en la aplicada, y ha comenzado a impactar en desarrollos de la metodología estadística así como en diversos campos de la ciencia aplicada [136]. Métodos soportados por las ondeletas se están desarrollando dentro de la estadística en áreas como la regresión, densidad y estimación de funciones, análisis de factores, modelación y predicción en el análisis de series de tiempo y estadísticas espaciales. La revelación de conexiones entre las ondeletas y el modelo de la estadística Bayesiana han generado atractivas vertientes donde convergen ambas áreas de investigación, con gran potencial de impacto en trabajos aplicados.

Se capturó el interés de la comunidad estadística cuando Mallat encontró una conexión entre las ondeletas y el procesamiento de señales, a la vez que Donoho y Johnstone demostraban que el umbralado de ondeletas tenía propiedades estadísticas óptimas. Desde entonces, las ondeletas han probado ser útiles en muchas disciplinas estadísticas, sobre todo en estadística no paramétrica y en el análisis de series de tiempo. Más recientemente, los conceptos Bayesianos y los enfoques de modelado han sido identificados como prometedores marcos de referencia para las aplicaciones de eliminación de ruido soportadas por ondeletas.

Adicionalmente, al remplazar a las bases ortonormales tradicionales (por ejemplo, Hermite o Fourier) en diversos problemas estadísticos, las ondeletas derivaron en nuevas técnicas y vigorizaron algunas de las ya existentes. Así se trate de un simple reemplazo de las bases ortogonales tradicionales por bases de ondeletas, en muchas ocasiones se obtiene mejor localización y economía. Las ondeletas pueden proporcionar una aproximación no lineal eficiente y adaptabilidad a contornos de suavidad desconocida mediante el encogimiento de los coeficientes de ondeleta. El encogimiento de ondeletas se logra mediante el uso implícito o explícito de modelos estadísticos en el dominio de las ondeletas. A continuación revisaremos brevemente algunas características de la descomposición de ondeletas que son especialmente útiles en el campo de la estimación:

4.4.1. Compactación de la energía

La transformada de ondeleta tiende a compactar la energía de los datos de entrada. Así, aunque la transformación preserva la energía (la norma ℓ^2 de los datos), ésta está concentrada mayormente en unos cuantos de los coeficientes de los datos transformados. Esta compactación reduce la gama de modelos estadísticos factibles y facilita el umbralado.

4.4.2. Buena localización

Las ondeletas, como bloques constructivos de un modelo, están bien localizadas tanto en el tiempo como en la escala (frecuencia). Señales con cambios locales rápidos (señales con discontinuidades, picos, impulsos de corta duración, etc.) pueden ser representados en forma precisa mediante unos pocos coeficientes de ondeleta. Generalmente, esta característica no aplica para las bases ortonormales típicas que pueden requerir muchos coeficientes compensadores para describir eventos de

discontinuidad o para suprimir los efectos de Gibbs. Las ondeletas forman diccionarios completos y proporcionan una representación económica de fenómenos reales con elaborados comportamientos tanto en tiempo como en frecuencia.

4.4.3. Manejo del principio de Heisenberg

El principio de incertidumbre de Heisenberg afirma que al modelar un fenómeno en el tiempo y en la frecuencia, no es posible ser precisos simultáneamente en ambos dominios. En el procesamiento de señales se presenta una instancia principio de Heisenberg sobre todo con el análisis de Fourier. La naturaleza de las ondeletas automáticamente balancea la precisión en el tiempo y en la frecuencia. Así, la representación económica de las ondeletas puede ser vista como la capacidad de las ondeletas para enfrentar las limitantes del principio de Heisenberg en una forma que depende únicamente de los datos.

4.4.4. Propiedad de emblanquecimiento

Existe amplia evidencia teórica y empírica de que las transformaciones de ondeleta tienden a simplificar la estructura de dependencia de los datos originales. Entonces, es posible construir una base biortogonal que des-correlacione cierta serie de tiempo estacionaria.

4.4.5. Control de suavidad

En ciertas condiciones de suavidad, las ondeletas proporcionan bases “incondicionales” para importantes espacios suaves (como los espacios de Besov, espacios de Hölder, etc.). Lo que significa que controlando la magnitud de los coeficientes en el dominio de las ondeletas, se controla la suavidad de la función que ha sido descompuesta. Esta vinculación proporciona el marco teórico para el suavizado soportado por ondeletas, así como para la estimación de funciones y de densidad soportada por ondeletas.

4.5. Eficiencia de la transformada de ondeleta discreta

La Figura 4.5 muestra información empírica acerca de los tiempos de ejecución tanto para la transformada rápida de Fourier (`fft()`) de Matlab como para la transformada discreta de ondeleta ortogonal (`FWT_PO()`) disponible en la biblioteca Wavelab para Matlab [5], con el máximo nivel de descomposición, que se emplearon durante el desarrollo de este trabajo. Ambas transformadas se ejecutaron en una computadora personal con un procesador Core Duo de Intel a 2GHz y el sistema operativo Windows. La figura muestra los resultados después de haber aplicado las dos transformaciones sobre conjuntos de datos de tamaño n . Se realizaron 1000 repeticiones para cada valor de n a fin de promediar los tiempos de ejecución. Finalmente, se graficó el tiempo promedio de ejecución que resulta de dividir el tiempo total consumido entre n .

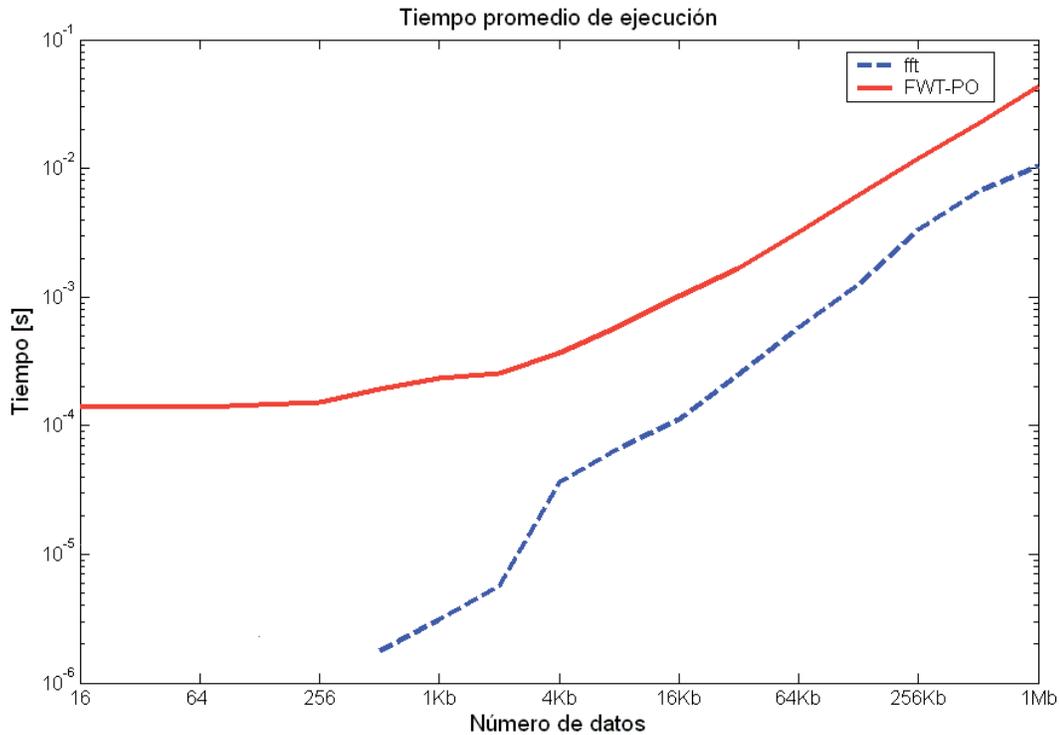


Figura 4.5: Tiempo de ejecución promedio (tiempo consumido entre n) para la realización de la FFT (fft) y la transformada de ondeleta discreta (FWT-PO de Wavelab) en Matlab

En la Figura 4.5 se pueden observar dos secciones muy claras para ambas transformadas. Las secciones se distinguen por el cambio significativo en la pendiente a partir de 2048 datos. En la primera sección, por debajo de 2048 datos, el tiempo de ejecución promedio se mantiene aproximadamente constante para la DWT, mientras que después de 2048 datos el tiempo de ejecución promedio se incrementa conforme aumenta n . Por otra parte, para la FFT el tiempo de ejecución promedio se incrementa conforme aumenta n en todo el rango medido, aunque con el susodicho cambio en la pendiente. Se observa que la FFT es más rápida que la DWT en todo el rango, pero conforme aumenta el número de datos la diferencia disminuye hasta mantenerse, a partir de 4096 datos, en poco menos de 10 veces.

Se ha reportado que, en teoría, la complejidad computacional de una transformada de ondeleta discreta básica es del orden de n , comparado con el orden $n \log n$ para la FFT [109]. Para el caso del algoritmo de DWT empleado en este trabajo, como se puede ver en la Figura 4.5, esta previsión teórica se cumple para un número de datos menor o igual a 2048 puntos. El cambio abrupto de la pendiente tanto en la curva de la DWT como en la curva de la FFT justo en $n = 2048$, se explica como consecuencia de la administración de memoria que realiza Matlab. En Matlab cada uno de los datos procesados ocupa 8 Bytes, entonces es claro que no es posible mantener el arreglo completo con los datos en el espacio de trabajo con memoria rápida a partir de cierto valor de n . Ya que una mayor cantidad de datos necesita ser almacenada en una memoria más grande pero con mayor

latencia, entonces se explica que el tiempo de ejecución promedio de las transformadas aumente significativamente, debido a un mayor tiempo para acceder a la memoria, a partir de cierto punto. Sin embargo, especialmente cuando el número de datos es mayor a 4096, es claro que el consumo de tiempo es aproximadamente del mismo orden para ambas transformadas, por lo que podemos afirmar que la transformada de ondeleta discreta se mantiene como un algoritmo rápido.

4.6. Construcción de las ondeletas

La ecuación de escalamiento (4.32) expresa la manera en que se relacionan las funciones de escalamiento entre dos escalas consecutivas. La expresión (4.32) dice que $\varphi(t)$ se construye mediante una combinación lineal (los coeficientes h_m) de una versión doblemente más fina de sí misma. Daubechies [43] probó un resultado clave que establece la existencia y construcción de las ondeletas:

- **Teorema 1 (Daubechies [43], p. 135).** Si $\{V_j\}_{j \in \mathbb{Z}}$ con φ que forma un análisis multi-resolución de $L^2(\mathbb{R})$, entonces existe una base de ondeletas ortonormales asociada $\{\psi_{j,k}(x) : j, k \in \mathbb{Z}\}$ para $L^2(\mathbb{R})$ tal que para toda $j \in \mathbb{Z}$

$$P_{j-1}f = P_j f + \sum_k \langle f, \psi_{j,k} \rangle \psi_{j,k}(x) \quad (4.43)$$

con

$$P_j f = \sum_{k \in \mathbb{Z}} a_{j,k} \varphi_{j,k}(x) \quad (4.44)$$

la proyección de la función f en el espacio V_j , con cierto conjunto de coeficientes $\{c_{j,k}\}$. Una posibilidad para la construcción de la ondeleta $\psi(x)$ es

$$\hat{\psi}(\omega) = e^{i\omega/2} H_0^*(\omega/2 + \pi) \hat{\varphi}(\omega/2) \quad (4.45)$$

donde $\hat{\psi}$ y $\hat{\varphi}$ son las transformadas de Fourier de ψ y φ , respectivamente, y donde

$$H_0(\omega) = \frac{1}{2} \sum_n h_n e^{-in\omega} \quad (4.46)$$

o equivalentemente

$$\psi(x) = \sum_n (-1)^{n-1} h_{-1-n}^* \varphi_{0,n}(x) \quad (4.47)$$

La expresión (4.47) es importante ya que expresa la forma en que la ondeleta es construida en términos de los coeficientes de la ondeleta padre de la escala anterior (más fina). Se tiene la siguiente notación para los coeficientes de la ondeleta:

$$g_n = (-1)^{n-1} h_{1-n} \quad (4.48)$$

A partir de (4.43), el Teorema 1 de Daubechies también deja en claro que la diferencia entre dos proyecciones $(P_{j-1} - P_j)f$ puede ser expresada como una combinación lineal de ondeletas. Por ello, el espacio W_j caracterizado por la base ortonormal de ondeletas $\{\psi_{j,k}(x)\}$ caracteriza el detalle perdido al pasar de P_j a P_{j+1} . La expresión (4.43) implica que para una función genérica $f(x)$:

$$f(x) = \sum_{k \in \mathbb{Z}} a_{j_0,k} \varphi_{j_0,k}(x) + \sum_{j=j_0}^{-\infty} \sum_{k \in \mathbb{Z}} d_{j_0,k} \psi_{j_0,k}(x) \quad (4.49)$$

que significa que una función puede ser representada por una parte suavizada o “núcleo” (*kernel-like*) relacionada con $\varphi_{j_0,k}$ y el conjunto de representaciones de detalle $\sum_{k \in \mathbb{Z}} d_{j_0,k} \psi_{j_0,k}(x)$ que acumulan información en un conjunto de escalas que van de j_0 al infinito (negativo). En este sentido, la primera parte de (4.49) puede ser vista como el nivel promedio o general de de función.

4.7. Momentos de desvanecimiento

Las ondeletas pueden poseer cierto número de momentos de desvanecimiento. Decimos que una función $\psi \in L^2(\mathbb{R})$ tiene m *momentos de desvanecimiento* si satisface

$$\int x^l \psi(x) dx = 0 \quad (4.50)$$

para $l = 0, \dots, m - 1$, bajo ciertas condiciones técnicas.

Los momentos de desvanecimiento son importantes, porque si una ondeleta tiene m momentos de desvanecimiento, ello significa que todos los coeficientes de ondeleta de cualquier función polinomial de grado m o menor serán exactamente cero. Así, si una función es muy suave y sólo ocasionalmente interrumpida por discontinuidades o singularidades, entonces los coeficientes de ondeleta serán muy pequeños, o incluso cero, para las partes suaves si éstas se expresan como un polinomio de orden m o menor. Esta representación económica (*sparse representation*) que resulta de la transformada de ondeletas, para un objeto dado, es una ventaja para muchos problemas de estimación. El hecho de tener sólo unos pocos coeficientes significativos se traduce en una disminución de los coeficientes que tienen que ser estimados. En pocas palabras y en términos de estimación, es mejor estimar unos pocos coeficientes para un conjunto de n datos que estimar n coeficientes para n datos.

4.8. Familias de ondeletas básicas

4.8.1. Ondeletas de Haar

Las ondeletas de Haar tienen limitaciones prácticas debido a su naturaleza discontinua que le impide aproximar funciones suaves. Sin embargo, las ondeletas de Haar son las mejores, por su

sencillez, para propósitos ilustrativos. La función de escalamiento (4.32) es muy simple para el caso de Haar:

$$\varphi(x) = \varphi(2x) + \varphi(2x - 1) = \frac{1}{\sqrt{2}}\sqrt{2}\varphi(2x) + \frac{1}{\sqrt{2}}\sqrt{2}\varphi(2x - 1) \quad (4.51)$$

lo que resulta en los siguientes coeficientes del filtro:

$$h_0 = h_1 = \frac{1}{\sqrt{2}}$$

con las siguientes funciones de transferencia:

$$H_0(\omega) = \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}e^{-i\omega 0}\right) + \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}e^{-i\omega 1}\right) = \frac{1 + e^{-i\omega}}{2}$$

$$H_1(\omega) = -e^{-i\omega}H_0^*(\omega + \pi) = -e^{-i\omega}\left(\frac{1}{2} - \frac{1}{2}e^{i\omega}\right) = \frac{1 - e^{-i\omega}}{2}$$

En el caso de la familia Haar es fácil comprobar algunas propiedades. Por ejemplo, ya que $\angle H_0(\omega) = -\frac{\omega}{2}$, la ondeleta de Haar tiene fase lineal, lo que implica una función de escalamiento simétrica en el dominio temporal. También es fácil comprobar que se cumple la condición de ortogonalidad $|H_0(\omega)|^2 + |H_1(\omega)|^2 = 1$

(4.52)

De (4.45) se deriva que

$$\hat{\psi}(\omega) = H_1(\omega)\hat{\varphi}\left(\frac{\omega}{2}\right) = \frac{1 - e^{-i\omega/2}}{2}\hat{\varphi}\left(\frac{\omega}{2}\right) = \frac{1}{2}\hat{\varphi}\left(\frac{\omega}{2}\right) - \frac{1}{2}\hat{\varphi}\left(\frac{\omega}{2}\right)e^{-i\omega/2}$$

que después de aplicar la transformada inversa de Fourier resulta en

$$\psi(x) = \varphi(2x) - \varphi(2x - 1)$$

para el dominio temporal, con lo que se encuentra la función de ondeleta ψ para la familia Haar. Entonces, de la expresión para H_1 se puede concluir que

$$g_0 = -g_1 = \frac{1}{\sqrt{2}}$$

4.8.2. Ondeletas con soporte compacto de Daubechies

Uno de los más importantes logros en la teoría de ondeletas es la construcción de ondeletas ortogonales con soporte compacto, como las ondeletas de Haar, pero más suaves. Daubechies construyó tales ondeletas de manera ingeniosa con la ecuación (4.32), lo que resultó en una familia de ondeletas ortonormales y más tarde en otras familias. Cada miembro de la familia suele ser indexado por un número N , que indica el número de momentos de desvanecimiento, aunque algunos autores emplean el índice para indicar el número de coeficientes del filtro, que es el doble

del número de momentos de desvanecimiento.

Las ondeletas simétricas resultan en filtros con fase lineal. Sin embargo, con excepción de las ondeletas de Haar, las ondeletas Daubechies con soporte finito de valores reales no pueden ser simétricas (familia conocida como de *fase extrema*). Para minimizar el grado de asimetría, se generó una familia de ondeletas menos asimétricas conocidas como *Symmlet*. Algunas funciones de escalamiento y ondeletas de la familia Daubechies y Symmlet se ofrecen en las Figuras 4.6 y 4.7, respectivamente.

4.9. Conclusiones del capítulo

El arribo del análisis de señales mediante la descomposición con bases de ondeletas (ondas de corta duración) introdujo un nuevo enfoque para realizar estimaciones con base en la información resultante del denominado análisis multi-resolución. Donoho [46] notó que la “heurística de Mallat” conduce a la noción de que una función con suavidad variante puede representarse económicamente por una base de ondeletas suaves y, por lo mismo, ser más fácilmente estimada [75]. Aquel primer trabajo sobre la estimación con base en la ondeleta derivó en técnicas de eliminación de ruido que, con ayuda de reglas de umbralado apropiadas, obtuvieron mejores resultados que los filtros lineales con fundamento en las estadísticas de la señal. Como veremos, estos principios son perfectamente aplicables para la estimación del canal PLC, con resultados muy útiles para estos ambientes contaminados con una gran cantidad de ruido.

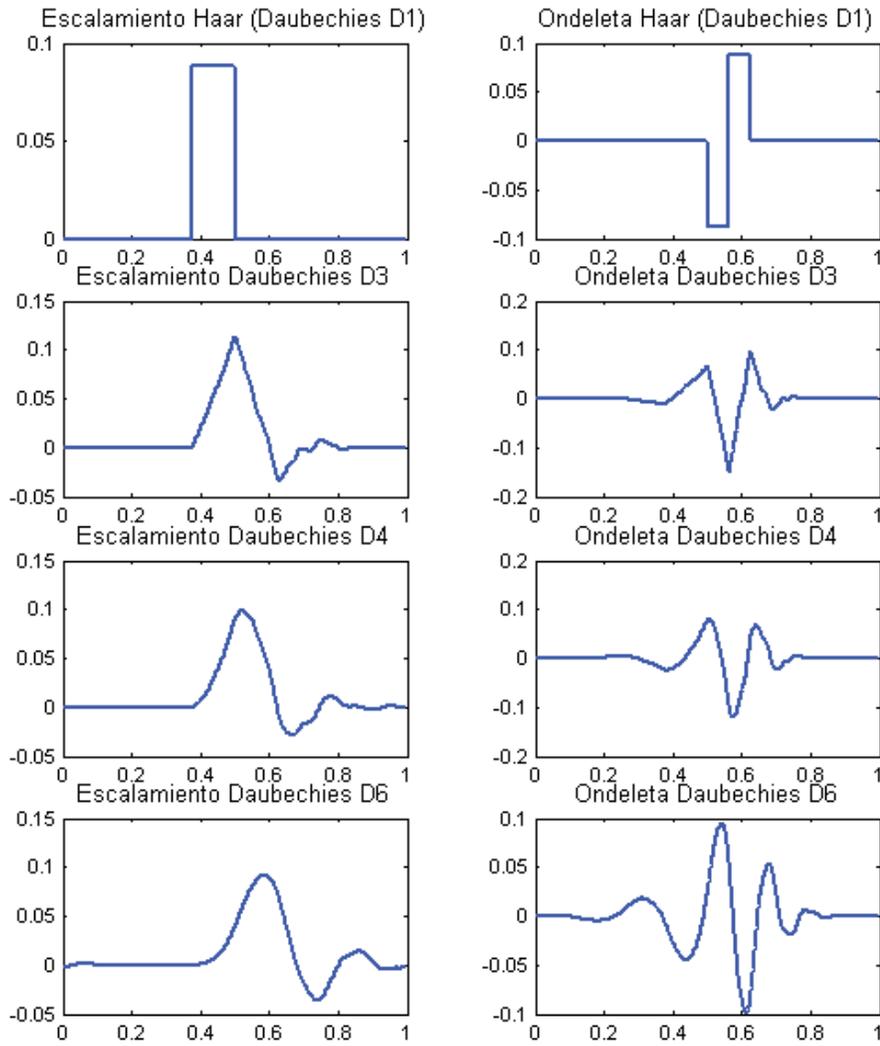


Figura 4.6: Gráficas para funciones de escalamiento y ondeleta de la familia Daubechies con $N=1, 3, 4$ y 6

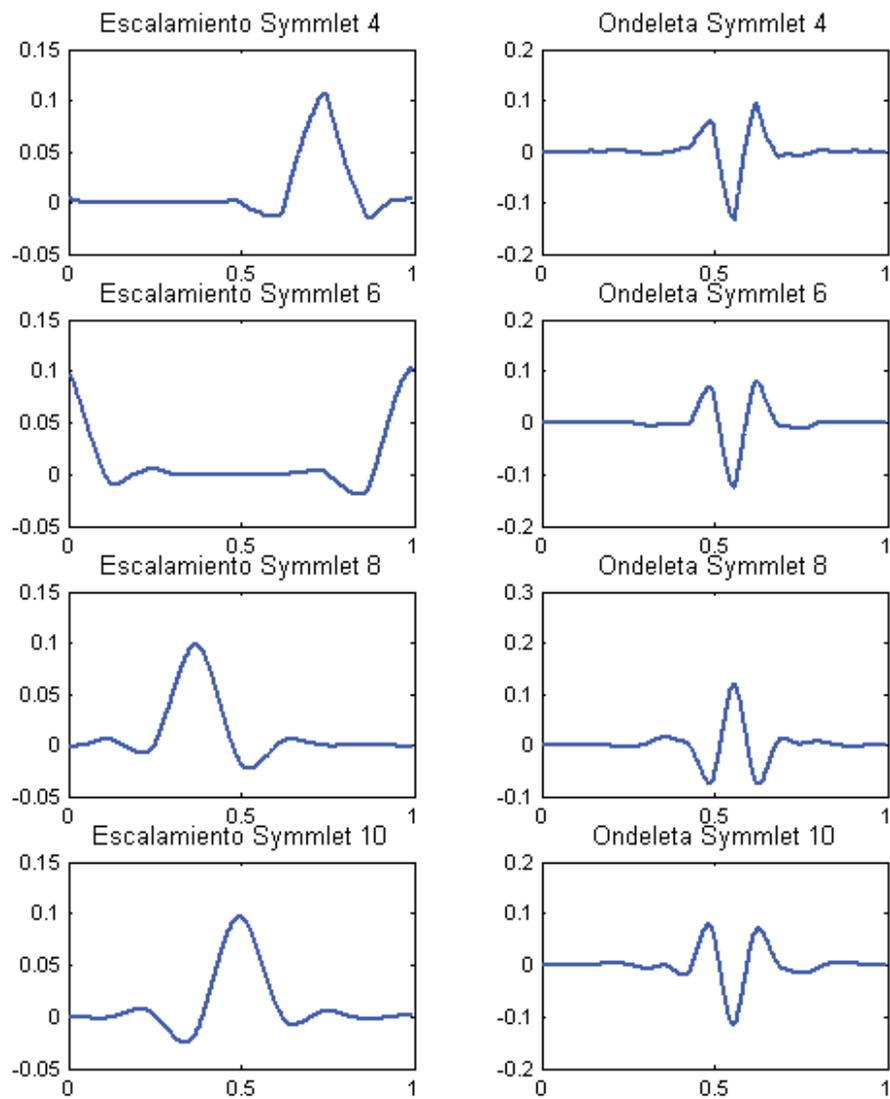


Figura 4.7: Gráficas para funciones de escalamiento y ondeleta de la familia Symmlet con $N=4, 6, 8$ y 10

Capítulo 5

Estimación de canales PLC soportada por descomposición de ondeleta

Aquí se describirá la aplicación de algunas técnicas de estimación y eliminación de ruido soportadas por la descomposición de ondeleta. Nos concentraremos en los sistemas OFDM y, especialmente, en las aplicaciones PLC.

5.1. Antecedentes

En los capítulos 1 y 2 vimos que la línea eléctrica usada como un canal de comunicación es un medio de propagación con retos técnicos considerables, ya que tiene una respuesta selectiva en frecuencia y variante en el tiempo; una función de transferencia dependiente de la posición, de la topología de la red así como de las cargas conectadas; además de sufrir de perturbaciones debidas a ruido cromático mezclado con ruido impulsivo. También vimos que la aplicación de sistemas con multiplexado por repartición de frecuencias ortogonales (OFDM, por sus siglas en inglés) y su variante conocida como Multi-tono Discreto (DMT, por sus siglas en inglés) tiene una amplia aceptación dentro de la comunidad PLC debido a sus ventajas en la transmisión de datos a altas velocidades sobre canales selectivos en frecuencia y variantes en el tiempo.

Ya que los sistemas OFDM a menudo emplean detección coherente, es necesario un conocimiento perfecto de las atenuaciones del canal y de la potencia del ruido en cada subcanal. Por lo general, para la estimación del canal, los transreceptores OFDM usan señales pilotos que son repetidas periódicamente para la estimación dinámica del canal cuando éste es variante en el tiempo, tal y como es el caso del canal PLC. Incluso, otras técnicas de estimación del canal, como las llamadas semi-ciegas y ciegas, también han encontrado aplicación en el ámbito PLC. Varios de los estimadores desarrollados tienen un alto costo computacional porque se fundamentan en el conocimiento de las estadísticas de segundo orden del proceso de atenuación, lo que además puede requerir del cálculo de la inversa de una o varias de matrices.

La observación de los coeficientes complejos de la función de transferencia del canal PLC es

afectada por diversos tipos de ruido (cromático, de banda angosta, periódico, etc.), sin embargo, el ruido impulsivo de corta duración y gran magnitud puede convertirse en la principal limitante en el desempeño de una aplicación PLC de alta velocidad [9]. La estimación del canal puede ser mejorada mediante técnicas de eliminación de ruido que remueven las perturbaciones aleatorias adicionadas pero, al mismo tiempo, mantienen las características relevantes de la señal tanto como es posible. Los filtros estadísticos como el filtro Wiener o el de promedio, pueden ser empleados para remover ruido, pero las técnicas soportadas por las ondeletas han probado obtener mejores resultados que dichos filtros clásicos [136].

Hasta donde sabemos, no se han publicado trabajos donde se aprovechen y adapten las técnicas de eliminación de ruido soportadas por ondeletas para la estimación del canal en entornos contaminados por ruido impulsivo ni, específicamente, en escenarios PLC. En la mayoría de la literatura especializada, se considera a los entornos inalámbricos que son perturbados solamente con ruido gaussiano. En este trabajo se propone una metodología para adaptar algunas de las técnicas de estimación y eliminación de ruido soportadas por ondeletas a fin de mejorar la estimación del canal, en el sentido de mínimos cuadrados (LS, por sus siglas en inglés), para sistemas OFDM con aplicación a los ambientes PLC.

5.2. Orígenes y evolución de la estimación empleando ondeletas

La estimación soportada por ondeleta, de señales y funciones, tiene su origen en los trabajos de la comunidad estadística quienes detectaron que Mallat [96] había asociado al umbralado algunas propiedades estadísticas óptimas muy convenientes. Desde entonces, las ondeletas han demostrado ser útiles en muchas disciplinas de la estadística, particularmente en la estadística no paramétrica y en el análisis de series de tiempo. Más tarde, se descubrió que los conceptos Bayesianos y los enfoques de modelación proporcionaban un marco prometedor para aplicaciones de eliminación del ruido soportadas por la ondeleta [136].

El arribo de las bases de ondeleta evidenció que una representación económica (*sparsity*) es un aspecto de importante repercusión para la estimación. Donoho [46] notó que la “heurística de Mallat” conduce a la noción de que una función con suavidad variante puede representarse económicamente por una base de ondeletas suaves y, por lo mismo, ser más fácilmente estimada [75]. Aquel primer trabajo sobre la estimación soportada por ondeletas derivó en técnicas de eliminación de ruido que, con ayuda de reglas de umbralado apropiadas, obtuvieron mejores resultados que los filtros lineales con fundamento en las estadísticas de la señal, como es el caso de los filtros Wiener o los filtros de promedio. En años recientes, se ha registrado un número considerable de investigaciones relacionadas tanto con la estimación soportada por ondeletas como con la eliminación de ruido en señales e imágenes, a partir del umbralado en el dominio de las ondeletas. El lector puede encontrar en [109] una revisión introductoria de los desarrollos más relevante de estas áreas en los últimos 15 años.

5.3. Trabajos relacionados con la estimación del canal empleando ondeletas

En la literatura de las comunicaciones digitales, sólo algunos trabajos se relacionan con el enfoque de la descomposición de ondeleta para la estimación del canal. En el trabajo pionero de V. Zaharov y F. Casco [141] se propone usar la Transformada Discreta de Ondeleta (DWT, por *Discrete Wavelet Transform*) para estimar un canal Rayleigh atenuado, a partir de la noción de que una DWT descompone los datos en subespacios que son más fáciles de modelar con predictores lineales.

En [59] los autores proponen una estimación ciega del canal descomponiendo el proceso de atenuación mediante una base de ondeleta y luego usando la técnica de Monte Carlo secuencial para rastrear tanto a los coeficientes de ondeleta como a los símbolos transmitidos, tanto en canales de atenuación llana como en los selectivos en frecuencia. En [94] se propone un modelo de canal a partir de ondeletas de M-bandas para estimar un canal variante en el tiempo y selectivo en frecuencia, pero los resultados reportados muestran un desempeño pobre cuando la relación señal a ruido (SNR, por *Signal to Noise Ratio*) es baja. En [86] se presenta una estimación del canal semi-ciega donde la matriz de autocorrelación de la señal recibida es depurada¹ en el dominio de la ondeleta. En [124] se propone un algoritmo de Maximización de la Esperanza en conjunto con la técnica de Máximo *a Posteriori* (EM-MAP, por sus siglas en inglés) soportada por ondeletas a fin de obtener una estimación semi-ciega del canal para el caso de las comunicaciones de banda ancha extrema (UWB, por sus siglas en inglés). En [140] se propone una técnica de eliminación del ruido que realiza cierto “encogimiento” (*shrink*) de los coeficientes de la descomposición de ondeleta para mejorar la estimación del canal con el criterio de Mínimos Cuadrados (LS, por sus siglas en inglés) en escenarios IEEE802.16.

En el campo de las comunicaciones PLC, los trabajos publicados relacionados con la aplicación de la descomposición de ondeleta se reparten principalmente en análisis del ruido [74, 64], análisis de componentes de eco [63], modelación del canal [12] y esquemas de modulación OFDM con ondeletas (WOFDM, por sus siglas en inglés) [72, 10].

5.4. Estimación del canal PLC con el enfoque de eliminación del ruido en el dominio de las ondeletas

En lo subsiguiente, se describirá una metodología para mejorar la estimación LS (en el sentido de Mínimos Cuadrados) de un canal PLC con ayuda de la descomposición de ondeleta, de tal manera que se evita el alto costo computacional que implica el conocimiento de las estadísticas del canal y el cálculo de la inversa de las matrices correspondientes. Se aprovecha la capacidad que tiene la descomposición de ondeletas para manejar señales heterogéneas e intermitentes, así como

¹En este contexto, *depuración* se usará como sinónimo de la eliminación de ruido o *denoising*.

su habilidad para eliminar eficazmente al ruido, todo ello mediante cálculos muy rápidos.

En general, el proceso de estimación soportado por ondeletas requiere un equilibrio entre la reducción del ruido y la preservación de los detalles significativos de la respuesta del canal. En este sentido, y para conseguir un buen desempeño, un algoritmo de eliminación del ruido tiene que adaptarse a las discontinuidades y singularidades de la respuesta del canal. La representación con ondeletas, de forma natural, facilita la construcción de dichos algoritmos adaptables al espacio. La descomposición de ondeletas concentra la información esencial de una señal en relativamente pocos, pero significativos, coeficientes que representan los detalles de la señal a diferentes escalas de resolución (condición que en inglés se denomina *sparsity*).

5.4.1. Descripción de estimación del canal con el criterio de Mínimos Cuadrados

Los métodos de estimación del canal más simples son aquellos diseñados en el sentido de reducir el error cuadrático y se conocen como técnicas de estimación por mínimos cuadrados (LS, por sus siglas en inglés). En ellas se asume el conocimiento de los símbolos transmitidos y la estimación simplemente se expresada como

$$\hat{\mathbf{c}}_k^{\text{LS}} = \begin{bmatrix} \frac{y_{k,0}}{x_{k,0}} & \frac{y_{k,1}}{x_{k,1}} & \dots & \frac{y_{k,N-1}}{x_{k,N-1}} \end{bmatrix} \quad (5.1)$$

donde $y_{k,n}$ son los símbolos de datos recibidos y $x_{k,n}$ son los símbolos de datos transmitidos para el $k^{\text{ésimo}}$ símbolo OFDM. Según el estimador LS, la estimación final para los coeficientes de atenuación del canal es típicamente una combinación lineal de los coeficientes $\hat{\mathbf{c}}_k^{\text{LS}}$ (por ejemplo, mediante un filtro de promedio). En la literatura, dos importantes familias de estimadores de canal con el enfoque de LS son: las técnicas asistidas por pilotos y las técnicas regidas por decisión.

Una forma práctica de mejorar una estimación del canal fue presentada en [33] y consiste en transformar la respuesta en frecuencia estimada al dominio temporal, mediante una IFFT, para ventanearla (*windowing*, en inglés) a fin de conservar únicamente las muestras dentro del tiempo de dispersión máximo permisible (menor o igual al intervalo de guardia), asumiendo que el resto de las muestras son solamente ruido. Procediendo de esta manera y volviendo al dominio de la frecuencia, mediante una FFT, se obtiene una respuesta suavizada en frecuencia, con lo que se mejora la estimación del canal mediante un método simple pero burdo. En este trabajo se retoma esta misma idea, pero procesando al ruido en el dominio transformado de las ondeletas y no directamente en la respuesta al impulso.

5.4.2. Umbralado en el dominio de las ondeletas

La medición de la respuesta del canal f (ya sea la respuesta en frecuencia o la respuesta temporal al impulso) es contaminada por cierto ruido aleatorio n , independiente e idénticamente distribuido (i.i.d.), resultando en la observación ruidosa $g = f + n$. El objetivo de la estimación es aproximar la respuesta característica f a partir de la observación ruidosa g , de tal manera que el

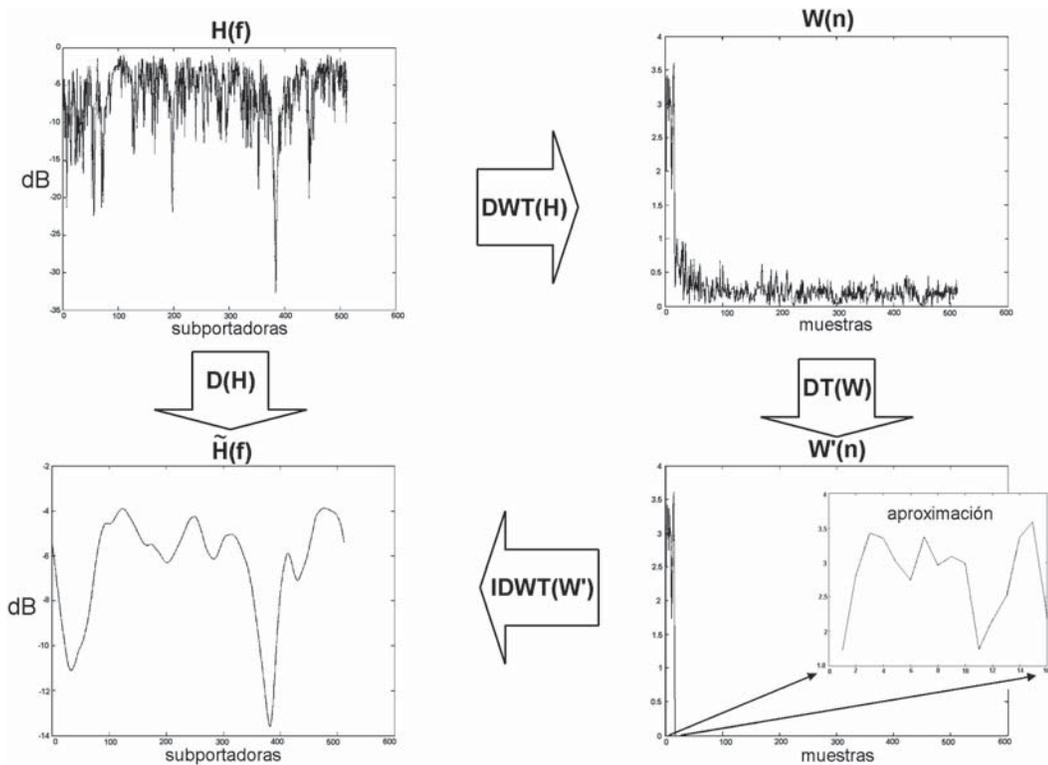


Figura 5.1: Mejora de la estimación del canal por umbralado en el dominio de la ondeleta

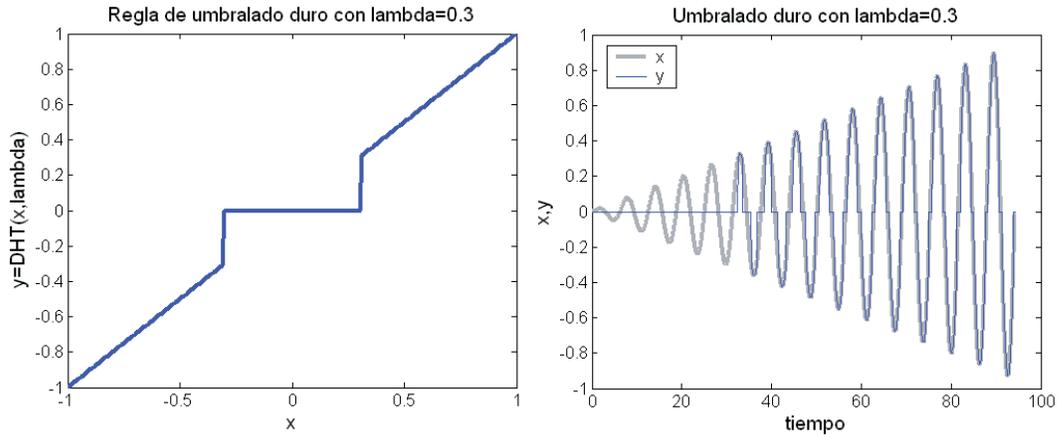


Figura 5.2: Umbralado duro

error cuadrático medio (MSE) sea mínimo. Para este fin, g es transformada al dominio de la ondeleta, descomponiéndola en sub-bandas que la desagregan en igual número de bandas de frecuencia. En principio, en los coeficientes con valores pequeños de cada sub-banda predomina el ruido, mientras que en los coeficientes con valores absolutos mayores se contienen más información de la señal útil que del ruido. Al forzar a cero a los coeficientes con valores por debajo de cierto umbral (coeficientes de ruido), la transformada de ondeleta inversa puede producir una reconstrucción menos ruidosa (ver el Figura 5.1). En tales procesos de eliminación del ruido normalmente son usadas las técnicas de umbralado *duro* y umbralado suave. En el umbralado duro el umbral λ se define como sigue:

$$\text{DHT}(x, \lambda) = \begin{cases} x & \text{if } |x| \geq \lambda \\ 0 & \text{if } |x| < \lambda \end{cases} \quad (5.2)$$

El umbralado duro es un procedimiento de “todo o nada” y, por ello, la función DHT no es continua (ver Figura 5.2). En el umbralado suave se sustituye a la función discontinua por una función continua, por ejemplo (ver Figura 5.3) [137]

$$\text{DST}(x, \lambda) = \begin{cases} x & \text{if } |x| \geq \lambda \\ 2x - \lambda & \text{if } \lambda/2 \leq x < \lambda \\ \lambda + 2x & \text{if } -\lambda < x \leq -\lambda/2 \\ 0 & \text{if } |x| < \lambda/2 \end{cases} \quad (5.3)$$

Adicionalmente, Donoho y Johnstone [47] propusieron un umbralado suave que “encoge” (*shrink*, en inglés) los coeficientes de ondeleta mediante la siguiente regla (ver Figura 5.4)

$$\text{DSS}(x, \lambda) = \text{sgn}(x) * \text{máx}(0, |x| - \lambda) \quad (5.4)$$

En todo esto, la elección de λ es crucial: si el umbral es demasiado pequeño o demasiado grande entonces la estimación, después del umbralado de los coeficientes en el dominio de la ondeleta, tenderá a seguir con exageración (*overfit*) a los detalles, incluso aquellos detalles que claramente son atribuibles al ruido, o a relajarse demasiado (*underfit*) en el ajuste de los datos. Donoho y Johnstone

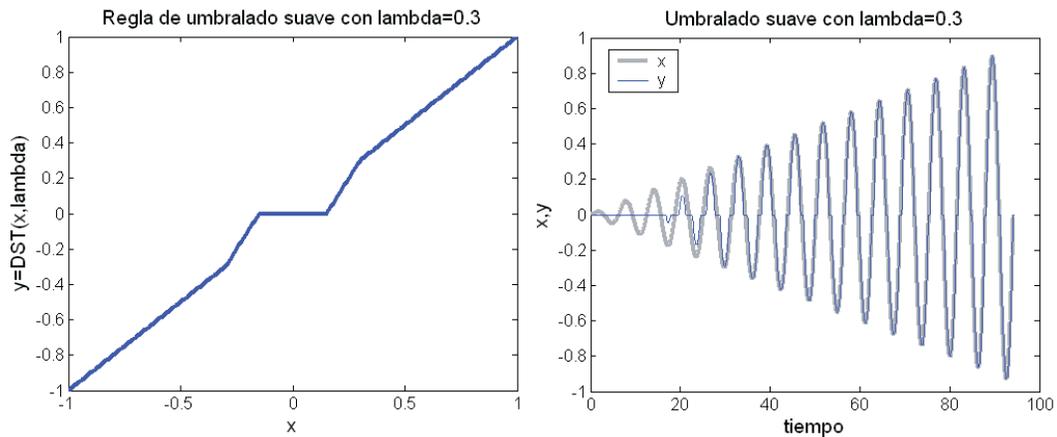


Figura 5.3: Umbralado suave

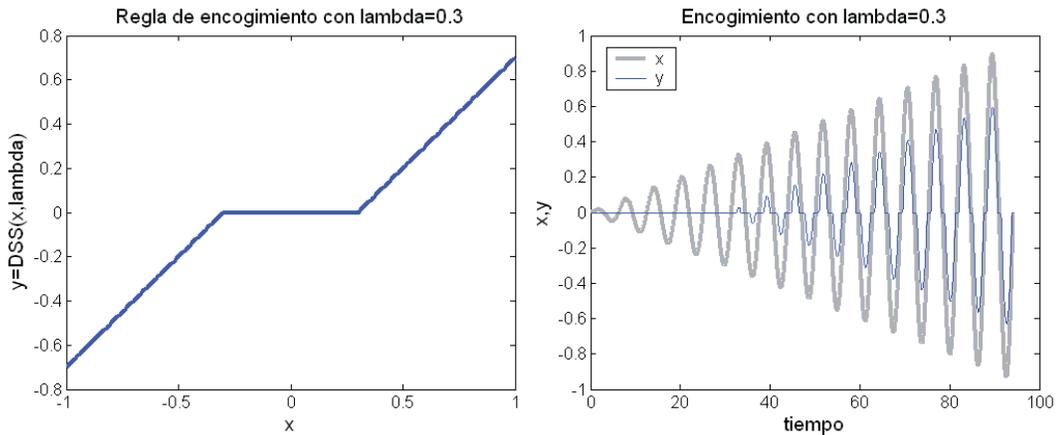


Figura 5.4: Umbralado suave por encogimiento

[47] propusieron un umbral *universal* $\lambda_{DJ} = \sigma\sqrt{2\log(n)}$, donde σ es la desviación estándar del ruido y n es el número de muestras observadas a fin de estimarlo. No obstante la simplicidad de dicha regla, ellos demostraron que la estimación no lineal soportada por ondeletas que resulta es espacialmente adaptable y cercana, en forma asintótica, al desempeño del método *minimax* [49] dentro del rango completo de los espacios de Besov. Además demostraron que, en ciertos espacios, se supera en forma asintótica a cualquier estimador lineal.

El umbral universal está determinado por los datos únicamente mediante la estimación de σ , pero también fueron desarrolladas reglas más complejas de umbralado que están completamente *determinadas por los datos* como, por ejemplo, el método *SureShrink* [48] y el *BayesThresh*, este ltimo un umbralado con enfoque Bayesiano [2].

Originalmente, todos estos métodos fueron desarrollados asumiendo que la señal observada era contaminada por ruido Gaussiano. Debido a ello, resulta interesante probar los más representativos

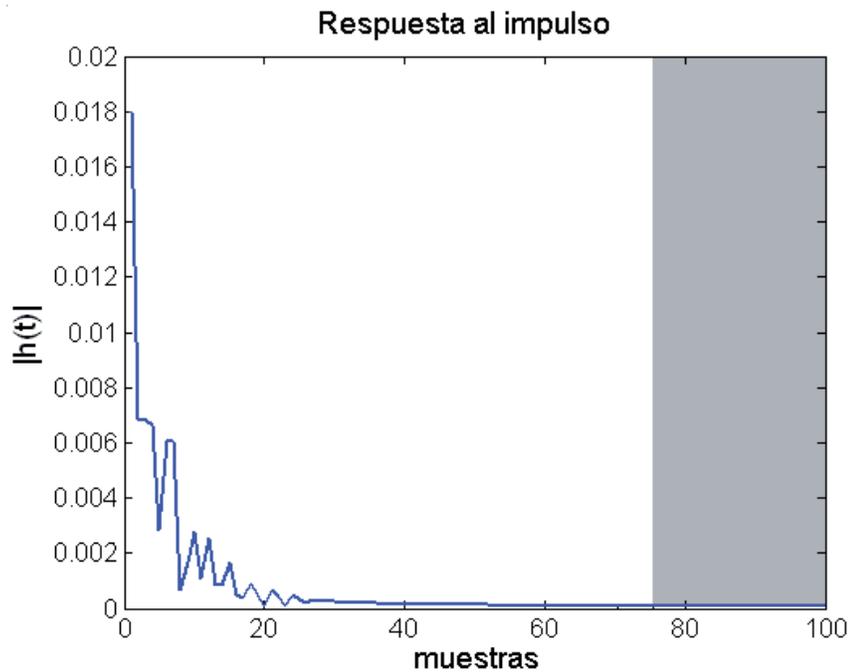


Figura 5.5: Máximo tiempo de dispersión para el canal simulado. Más allá de la muestra No. 76 la magnitud de la respuesta al impulso es despreciable

de los métodos de umbralado en un ambiente afectado por ruido impulsivo, a fin de evaluar su potencial ventaja y aplicación en ambientes PLC.

5.4.3. Metodología propuesta para la estimación del canal soportada por on-deletas

Como se comentó antes, un aspecto importante en la eliminación de ruido soportada por on-deletas es el cálculo de un valor apropiado para el umbral λ . En este trabajo se adoptó el umbral *universal* de Donoho y Johnstone, λ_{DJ} . Para calcular λ_{DJ} , se estimó la desviación estándar del ruido empleando las muestras ubicadas después del máximo tiempo de dispersión del canal, asumiendo que después de dicho intervalo toda muestra de la respuesta al impulso es prácticamente puro ruido.

Por ejemplo, en el canal simulado para este trabajo el intervalo de guardia (prefijo cíclico) tiene una longitud equivalente al 15 % de un símbolo OFDM simple. Asumiendo que hay una muestra por cada subportadora, tenemos 512 muestras (subportadoras) en un símbolo OFDM, por lo que el intervalo de guardia adiciona otras 76 muestras al símbolo OFDM transmitido (ver Figura 5.5). En consecuencia, se puede ver la máxima dispersión temporal como una frontera, pues cualquier muestra más allá de este punto se considera como puro ruido. En la Figura 5.6 se grafica la observación (magnitud) de la respuesta en frecuencia obtenida mediante un método asistido por pilotos. La estimación de la respuesta al impulso correspondiente se obtiene mediante una transformada de Fourier inversa (IFFT) y se grafica en la Figura 5.7 para nuestro ejemplo.

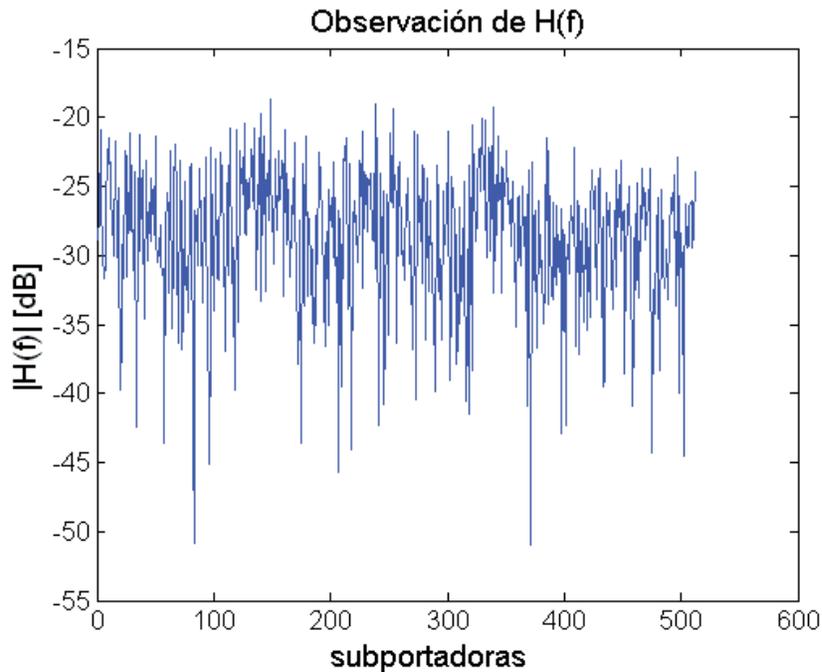


Figura 5.6: Observación de la estimación del canal obtenida mediante un método asistido por pilotos

Tomando en consideración las propiedades de la transformada de Fourier discreta, podemos observar en la Figura 5.7 que la parte positiva se compone de 256 muestras y la negativa por otras 256 muestras. Entonces, la proporción para el tiempo máximo de dispersión es de 38 muestras en este dominio de transformación, tanto para la parte positiva como para la negativa. Por lo tanto, las muestras restantes pueden ser consideradas como una ventana de puro ruido, tal y como se resalta en la Figura 5.7 mediante la región sombreada. Esta es, precisamente, la ventana con la región de ruido que se procesa para calcular la potencia promedio del ruido y obtener la desviación estándar σ . Con este valor de σ se calcula $\lambda_{DJ}[t] = \sigma\sqrt{2\log(n)}$, como nivel de umbral para el proceso de eliminación de ruido, lo que finalmente resulta en una estimación del canal soportada por ondeletas, como se ilustra en la Figura 5.8.

Después de ejecutar las simulaciones y medir la mejora obtenida en la estimación LS del canal mediante eliminación de ruido en la observación de la respuesta en frecuencia H o en la respuesta al impulso h , se observó que el último método es más eficaz (en términos de la tasa de errores en los bits, BER), sobre todo cuando la SNR es baja, para hacer frente a una mezcla de ruido Gaussiano más ruido impulsivo significativo.

La metodología para mejorar la estimación LS del canal mediante la eliminación de ruido soportada por ondeletas y partiendo de la respuesta al impulso $h(t)$ es la siguiente:

1. Obtener un vector con la observación de los coeficientes de atenuación del canal (la respuesta

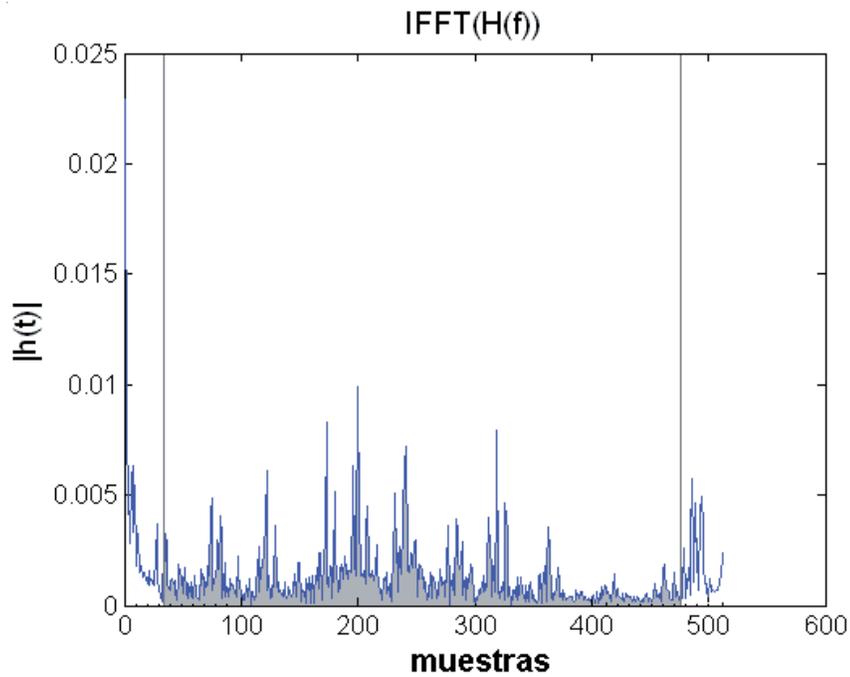


Figura 5.7: Respuesta al impulso estimada a partir de la respuesta en frecuencia observada. La ventana centrada en la mitad de las muestras incluye todas las muestras que pueden ser consideradas ruido puro

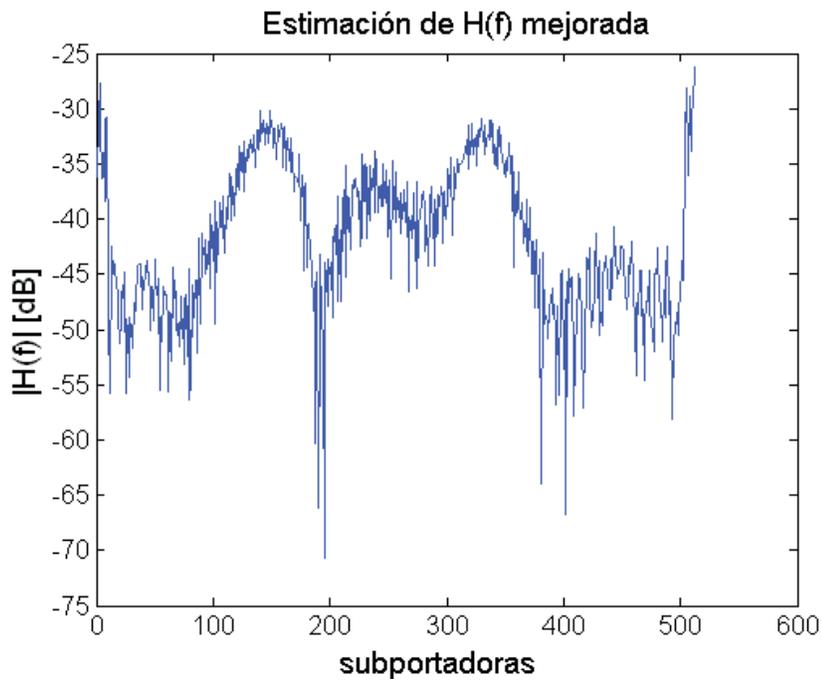


Figura 5.8: Estimación mejorada del canal mediante umbralado de la respuesta al impulso en el dominio de la expansión de ondeletas

en frecuencia $H(f)$) mediante una técnica de estimación del canal asistida por pilotos o regida por decisión.

2. Obtener la respuesta al impulso $h(t)$, aplicando una IFFT a la respuesta en frecuencia $H(f)$.
3. Asumir que las muestras más allá del tiempo de dispersión, delimitado por la duración del intervalo de guardia, son únicamente ruido y calcular su desviación estándar σ para calcular el umbral universal en el dominio del tiempo como $\lambda_{DJ}[t] = \sigma \sqrt{2 \log(n)}$, con n el número de muestras de ruido.
4. Transformar la respuesta al impulso $h(t)$ al dominio de la ondeleta mediante una DWT, con suficientes niveles de descomposición para que la información de aproximación (información gruesa) se represente por una escasa, pero razonable, cantidad de coeficientes representativos. Experimentalmente se determinó que un coeficiente (2^0 , equivalente al máximo nivel de descomposición) en el valor óptimo. No obstante, a fin de ahorrar tiempo de cómputo, se puede interrumpir la descomposición en el momento de alcanzar una aproximación con 16 coeficientes, pero con el costo de una ligera pérdida en el desempeño de la estimación soportada por ondeletas (ver Apéndice C).
5. Aplicar cualquier regla de umbralado a los coeficientes de ondeleta correspondiente a los componentes del detalle.
6. Después de preservar la información de aproximación y de umbralar la información del detalle, aplicar una DWT inversa (IDWT) sobre la descomposición de ondeleta resultante para reconstruir la respuesta al impulso estimada $\hat{h}(t)$.
7. Transformar la respuesta al impulso suavizada $\hat{h}(t)$ al dominio de la frecuencia, mediante una FFT, a fin de obtener la estimación del canal mejorada $\hat{H}(f)$.

La metodología para mejorar la estimación LS del canal mediante la eliminación de ruido soportada por ondeletas y partiendo de la respuesta en frecuencia $H(f)$ es la siguiente:

1. Obtener un vector con la observación de los coeficientes de atenuación del canal (la respuesta en frecuencia $H(f)$) mediante una la técnica de estimación del canal asistida por pilotos o regida por decisión.
2. Obtener la respuesta al impulso $h(t)$, aplicando una IFFT a la respuesta en frecuencia $H(f)$.
3. Asumir que las muestras más allá del tiempo de dispersión, delimitado por la duración del intervalo de guardia, son únicamente ruido y calcular el umbral universal para el dominio del tiempo $\lambda_{DJ}[t]$. Luego, calcular el umbral universal para el dominio de la frecuencia a partir del umbral universal temporal y considerando el principio de conservación de la energía de Parseval. A saber, $\lambda_{DJ}[f] = \sqrt{N} \times \lambda_{DJ}[t]$, con N el tamaño de la ventana FFT.
4. Transformar la respuesta en frecuencia $H(f)$ al dominio de la ondeleta mediante una DWT, con suficientes niveles de descomposición para que la información de aproximación se represente por una escasa, pero razonable, cantidad de coeficientes representativos (menor a 16 coeficientes).

5. Aplicar cualquier regla de umbralado a los coeficientes de ondeleta correspondiente a los componentes del detalle.
6. Después de preservar la información de aproximación y de umbralar la información del detalle, aplicar una DWT inversa (IDWT) sobre la descomposición de ondeleta resultante para reconstruir la respuesta en frecuencia suavizada $\hat{H}(f)$ que constituye la estimación del canal mejorada.

5.4.4. Simulaciones por computadora

A fin de evaluar la mejora en el desempeño de la estimación del canal PLC, alcanzada por las diversas técnicas de eliminación de ruido soportadas por ondeletas, se simuló en Matlab un enlace de comunicación DMT para un canal PLC contaminado por una mezcla de ruido Gaussiano más ruido impulsivo. Para poner en práctica las técnicas de eliminación de ruido soportadas por ondeletas usamos la biblioteca con funciones de transformada de ondeleta ofrecida en el paquete Wavelab para Matlab [50], así como el paquete EbayesThresh [5, 131] para disponer, a manera de referencia, con una regla de umbralado de enfoque Bayesiano. Entonces, para mejorar la estimación del canal LS, fueron probadas y comparadas cuatro técnicas eliminación del ruido soportadas por ondeletas: umbralado duro (DHT), según la regla (5.2); umbralado suave (DST), según la regla (5.3); umbralado suave con encogimiento (DSS), según la regla (5.4); y un umbralado Bayesiano (DBT), usando el paquete EbayesThresh con parámetros por defecto. También fue incluida, a manera de referencia, la técnica de eliminación del ruido por ventaneo de la respuesta de impulso (DW), tal y como se describe en la sección 5.2. En todos los casos donde se requería el umbral λ se usó el umbral universal λ_{DJ} correspondiente.

Al principio se usó la ondeleta Symmlet8, la más simétrica de las ondeletas ortogonales de Daubechies, como es sugerido en [2]. Sin embargo, también fueron probadas, las ondeletas de Daubechies Db4, la menos asimétrica de las ondeletas de soporte compacto, y la ondeleta de Daubechies Db6, como referencia, que es claramente asimétrica. Específicamente, se empleó la descomposición de ondeletas ortogonales mediante la función `FWT_PO()` del paquete Wavelab, con los niveles de descomposición necesarios a fin de conservar 16 coeficientes de aproximación (en nuestro caso, 5 niveles de descomposición ya que se usan 512 subportadoras). Se realizaron simulaciones Monte Carlo para el enlace PLC y se calculó la tasa de errores en los bits (BER) para varios valores de relación señal a ruido (SNR). En cada experimento se mantiene constante la densidad del ruido impulsivo. Con relación al comportamiento del canal, las simulaciones fueron realizadas en dos escenarios PLC, ambos selectivos en frecuencia: escenario con canal estacionario y escenario con canal ciclo-estacionario. En la siguiente sección se reporta y discute el desempeño, medido en términos de la BER, de las diversas técnicas de estimación soportadas por ondeletas que se probaron.

5.4.5. Sistema OFDM de referencia

En este caso se empleó un sistema DMT con datos entrelazados y codificados mediante la técnica Reed-Solomon (orientada a octetos, con 240 octetos de datos y 15 octetos redundantes) y mapeados con QPSK, mientras que la estimación del canal estuvo apoyada por una estrategia asistida por pilotos (PA), para el escenario estacionario, o por una estrategia regida por la decisión (DD), para el escenario ciclo-estacionario. Finalmente, se debe aclarar que el número real de subportadoras empleadas estuvo determinado por la atenuación del canal, estimada por la trama de datos previa. Específicamente, el nivel de conmutación fue colocado en -35 dB.

5.4.6. Resultados numéricos

Se consideró una topología de red PLC simple con un nodo en T (un canal con una derivación), como se muestra en la Figura 2.7 del capítulo 2. El entorno de simulación respecto al canal PLC es el especificado en la sección 2.6, a la vez que el transreceptor simulado corresponde al sistema básico especificado en la sección 2.7. Los valores empleados para la densidad del ruido impulsivo fueron 6000, 600 y 10 impulsos por ciclo de la línea. El valor de 6000 impulsos por ciclo representa la incidencia de aproximadamente 6 impulsos de ruido de alta energía por cada símbolo OFDM, lo que genera un ambiente contaminado por ruido impulsivo de alta densidad, que es nuestro principal objeto de estudio. Estos entornos ruidosos son de gran interés en nuestro trabajo debido a la conocida limitación que representan para el desempeño total del sistema.

Respecto de la relación de potencias entre el ruido Gaussiano y el ruido impulsivo dada en (2.33), ésta se fijó en $\Gamma = 0,00001$ y se mantuvo constante a lo largo de todas las simulaciones. Este valor representa ruido impulsivo de alta energía, que se traduce en picos de magnitud significativamente mayor al promedio de la señal útil. El valor $\Gamma = 0,00001$ garantiza que la potencia espectral del ruido impulsivo pueda superar por más de 40 dB al ruido Gaussiano, como se reporta en [9].

En las Figuras 5.9 y 5.10 podemos ver el desempeño, en términos de la probabilidad de error en los bits (BER), de las variantes de estimación en un canal PLC contaminado con ruido Gaussiano más ruido impulsivo con una densidad de 6000 impulsos por ciclo de la línea, para el caso estacionario y ciclo-estacionario, respectivamente. En las Figuras 5.11 y 5.10 observamos el desempeño con ruido Gaussiano más ruido impulsivo con una densidad de 600 impulsos. Finalmente, en las Figuras 5.12 y 5.14 apreciamos el desempeño con una baja densidad de ruido impulsivo de tan sólo 10 impulsos por ciclo de la línea (prácticamente puro ruido Gaussiano).

5.4.7. Discusión de resultados

Las simulaciones numéricas mostraron que, en todos los escenarios, el desempeño alcanzado por las bases Symmlet8, Db4 y Db6 fue muy similar. Debido a que todos los resultados son consistentes, en esta sección únicamente se graficarán los resultados correspondientes a la base Db4, la ondeleta de menor soporte (sólo 8 coeficientes diferentes de cero). En especial, haremos énfasis

en los resultados obtenidos para el caso de una alta densidad del ruido impulsivo (6000 impulsos por ciclo de la línea) debido a que nuestro interés está enfocado en determinar la capacidad de la estimación soportada por ondeletas para contrarrestar el ruido impulsivo. Fueron probadas las variantes de estimación del canal soportada por ondeletas partiendo tanto del dominio del tiempo ($D_{xx}(h)$) como del dominio de la frecuencia ($D_{xx}(H)$) pero, sobretodo en los escenarios con una alta densidad de ruido impulsivo, el desempeño resultante al eliminar el ruido partiendo de la respuesta al impulso con la técnica de umbralado suave de Donoho y Johnstone ($DSS(h)$) siempre fue mejor.

En las figuras presentadas se puede contrastar el desempeño de las variantes tanto para el caso estacionario como para el caso ciclo-estacionario. Por ejemplo, en la Figura 5.9 observamos el desempeño de todas las variantes probadas para un canal estacionario con estimación del canal apoyada por una estrategia asistida por pilotos. Por otro lado, en la Figura 5.10 podemos observar el desempeño de todas las variantes probadas para un canal ciclo-estacionario, con una estimación del canal apoyada por una estrategia regida por decisión. En las gráficas podemos observar la curva correspondiente a la estimación LS pura del canal, la estimación LS mejorada por eliminación del ruido mediante el ventaneo de la respuesta de impulso (DW), la estimación soportada por ondeletas con umbralado suave (DST), la estimación soportada por ondeletas con umbralado suave mediante la regla de encogimiento de Donoho y Johnstone (DSS), y la estimación soportada por ondeletas con umbralado Bayesiano (DBT). Nótese como en los casos de escenarios ciclo-estacionarios todos los métodos probados son asintóticos a una línea de error irreducible que se determina por la capacidad del método DD para dar seguimiento a las variaciones temporales del canal.

Aunque se usaron datos entrelazados, codificados con códigos Reed-Solomon y mapeados conforme a QPSK; los efectos del ruido impulsivo no fueron totalmente blanqueados (minimización del efecto de ráfaga). A fin de apreciar mejor el efecto del ruido impulsivo en el desempeño del sistema y para determinar el rango de confiabilidad de la estimación soportada por ondeletas, podemos notar el cambio en los órdenes de magnitud de la probabilidad de error en bits (BER) de los escenarios estacionarios con diferentes densidades de ruido impulsivo. En los escenarios ciclo-estacionarios también se pueden observar los efectos del ruido impulsivo, pero es más difícil apreciar los efectos del ruido impulsivo por sí solo debido al nivel de probabilidad irreducible debido a las limitaciones del método de seguimiento.

En la Figura 5.9, para un canal estacionario contaminado con ruido impulsivo de alta densidad a razón de 6000 impulsos por ciclo de potencia, comprobamos que los efectos sobre el rendimiento del sistema son devastadores. Observamos que hasta los 18 dB, la probabilidad de error está en un orden de magnitud entre 1 % y 10 % para todas las técnicas de estimación probadas. Se alcanzan niveles de BER menores a 1 % sólo hasta que la SNR es mayor a 18 dB. En la Figura 5.11, para un canal estacionario con ruido impulsivo de alta densidad a razón de 600 impulsos por ciclo de potencia, observamos que las curvas de BER se colocan en los niveles de operación considerados por las aplicaciones comerciales: una probabilidad de error del orden de 10 % si la SNR es menor a 2 dB, pero disminuye a 1 % cuando la SNR aumenta a 10 dB. Finalmente, en la Figura 5.13, para un canal estacionario con ruido esencialmente Gaussiano (tan sólo 10 impulsos por ciclo de la

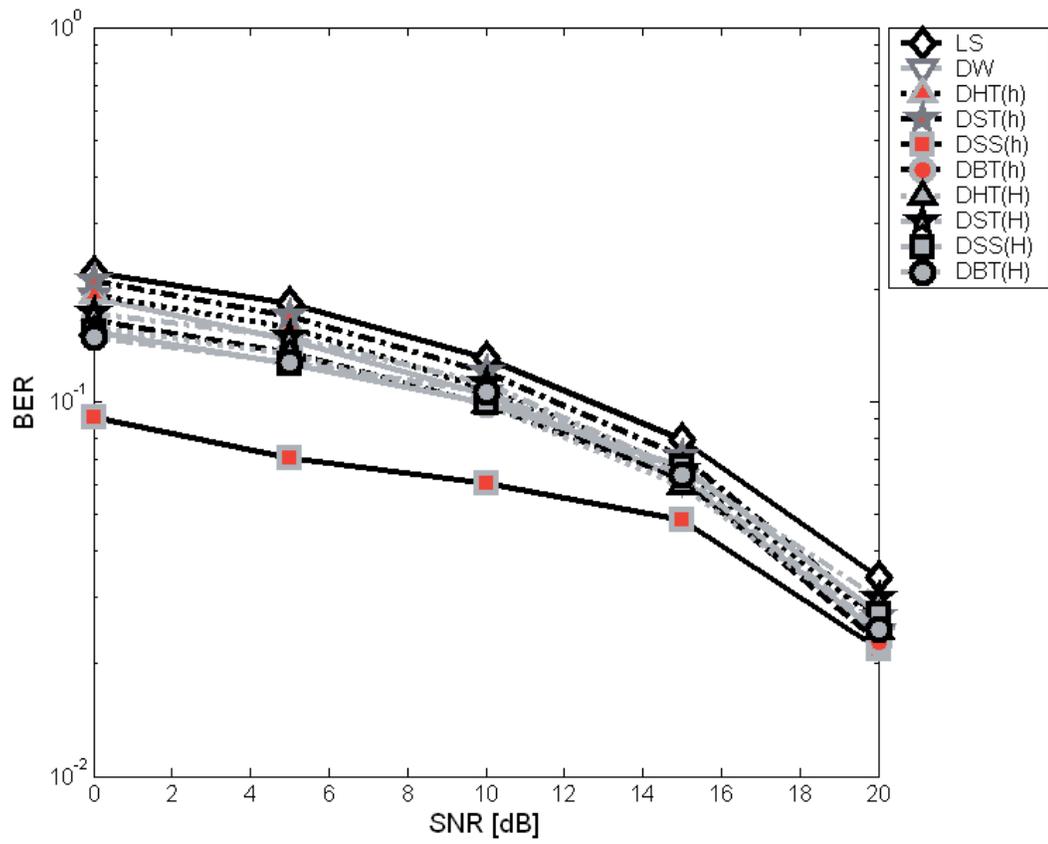


Figura 5.9: Escenario estacionario contaminado con ruido Gaussiano más ruido impulsivo con una densidad de 6000 pulsos por ciclo

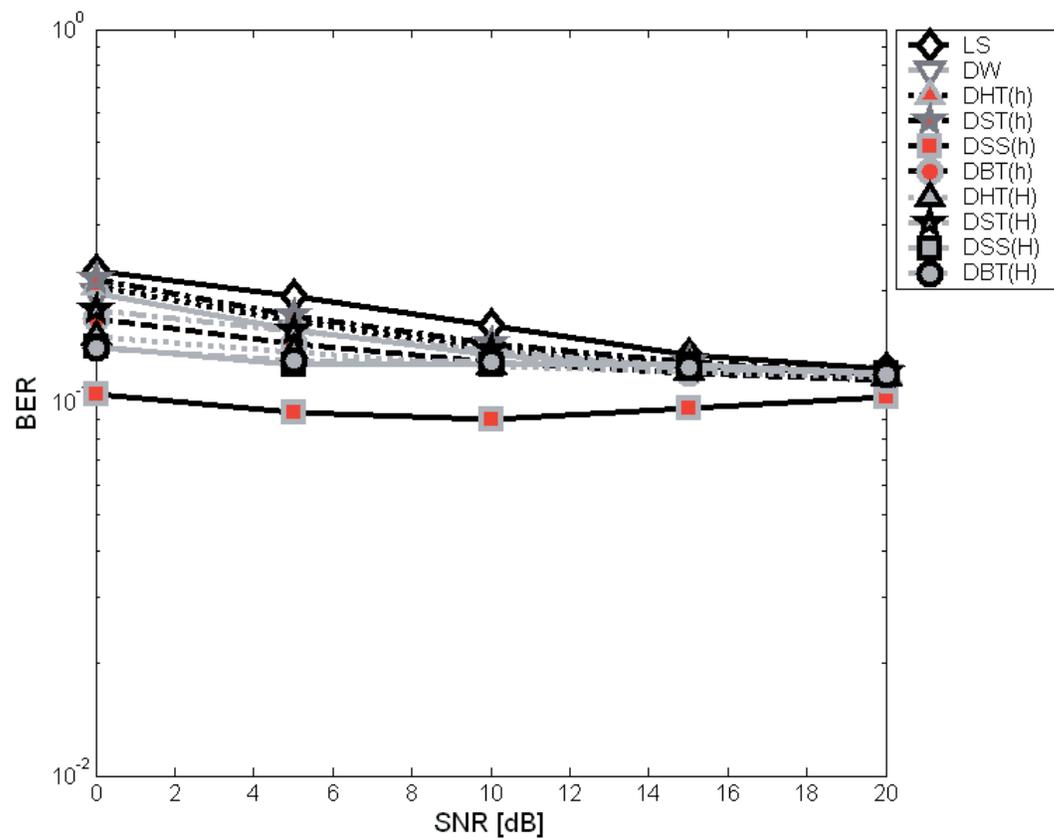


Figura 5.10: Escenario ciclo-estacionario contaminado con ruido Gaussiano más ruido impulsivo con una densidad de 6000 pulsos por ciclo

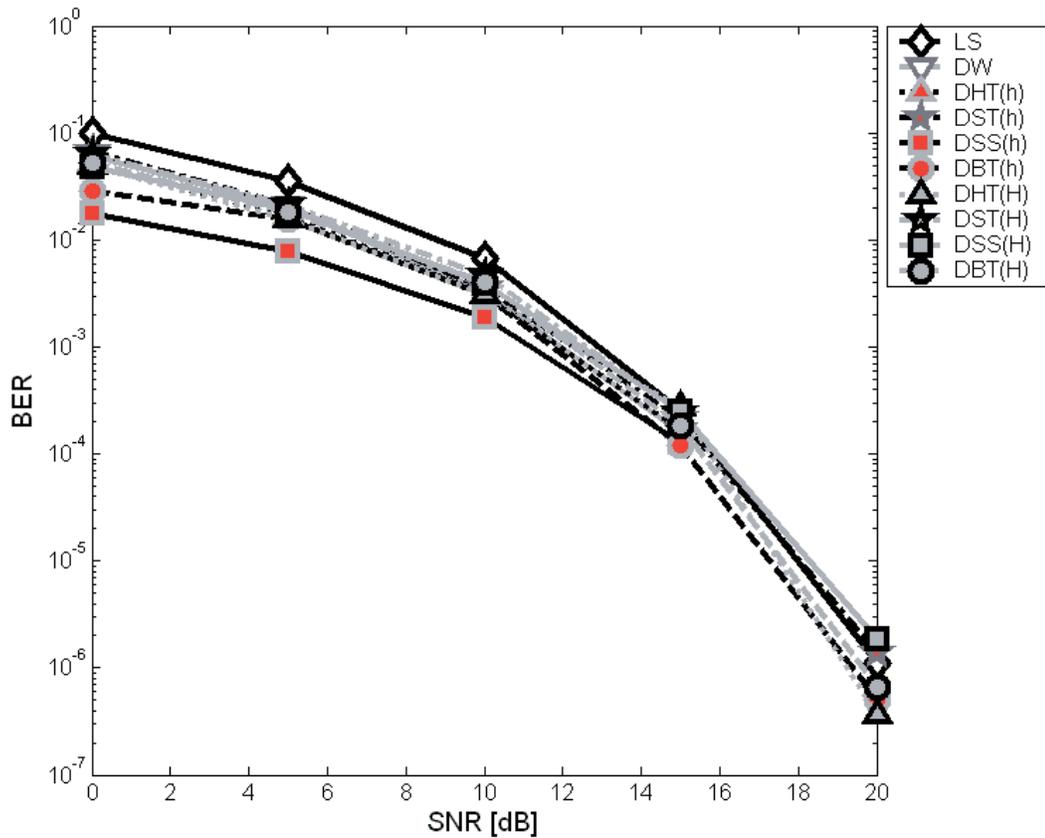


Figura 5.11: Escenario estacionario contaminado con ruido Gaussiano más ruido impulsivo con una densidad de 600 pulsos por ciclo

línea), podemos ver que se alcanzan los niveles de error de 1 % con una SNR de tan sólo 5 dB.

En todas estas gráficas es claro que la variante $DSS(h)$ tiene un mejor desempeño, en términos de la BER respecto de todas las técnicas probadas, lo cual es especialmente notorio cuando la SNR es baja, aunque que su ventaja va disminuyendo conforme aumenta la SNR. Esta disminución de la ventaja de la variante $DSS(h)$ con valores de SNR grandes se debe a que el proceso de eliminación del ruido, al suavizar demasiado (*underfit*) la verdadera característica de canal, resulta en una característica de canal sólo parcialmente aproximada, cuya limitación se hace evidente cuando la SNR es grande.

En las gráficas para los escenarios ciclo-estacionarios (Figuras 5.10, 5.12 y 5.14) también se puede observar, para todas las densidades de ruido impulsivo, una clara ventaja de la técnica de estimación del canal soportada por ondeletas en su variante $DSS(h)$, especialmente cuando el valor de SNR es bajo. Sin embargo, en este caso se observa un fenómeno muy peculiar enmarcado por la limitación del método de seguimiento empleado. Para los escenarios ciclo-estacionarios la estimación obtenida por la variante $DSS(h)$ se aproxima por debajo y asintóticamente al nivel de

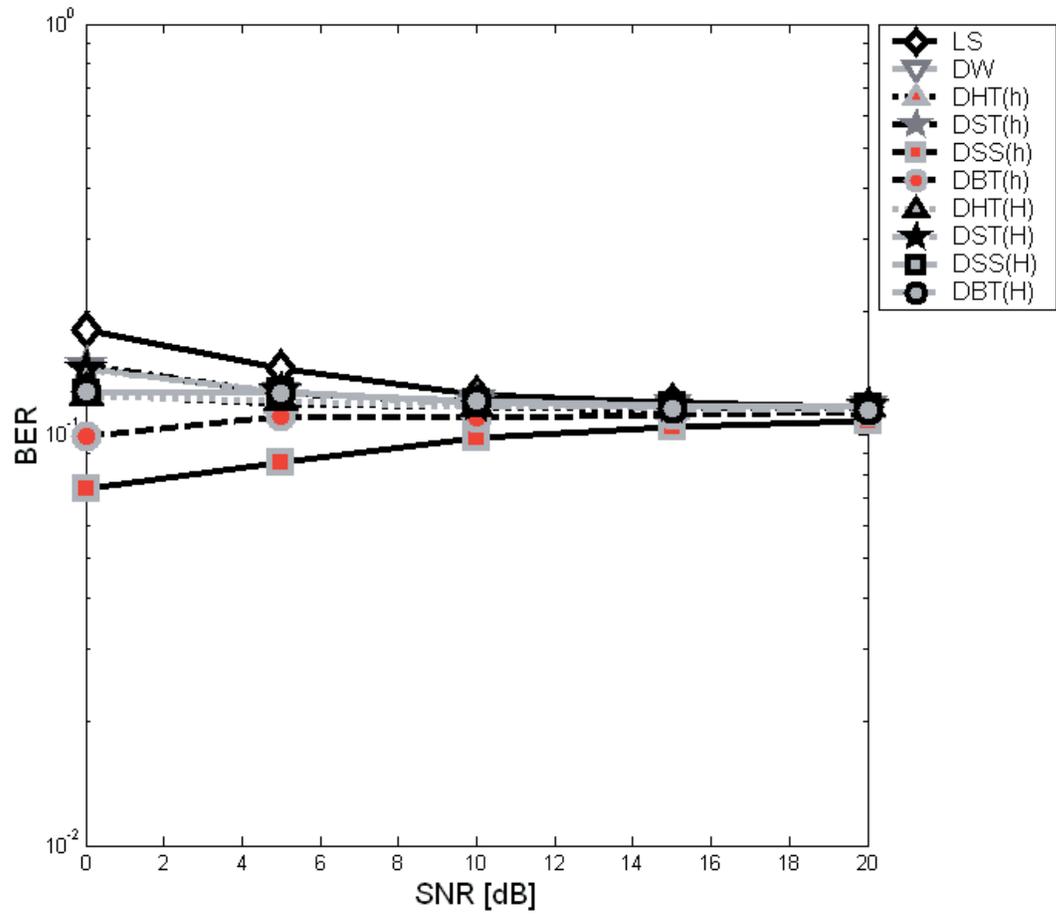


Figura 5.12: Escenario ciclo-estacionario contaminado con ruido Gaussiano más ruido impulsivo con una densidad de 600 pulsos por ciclo

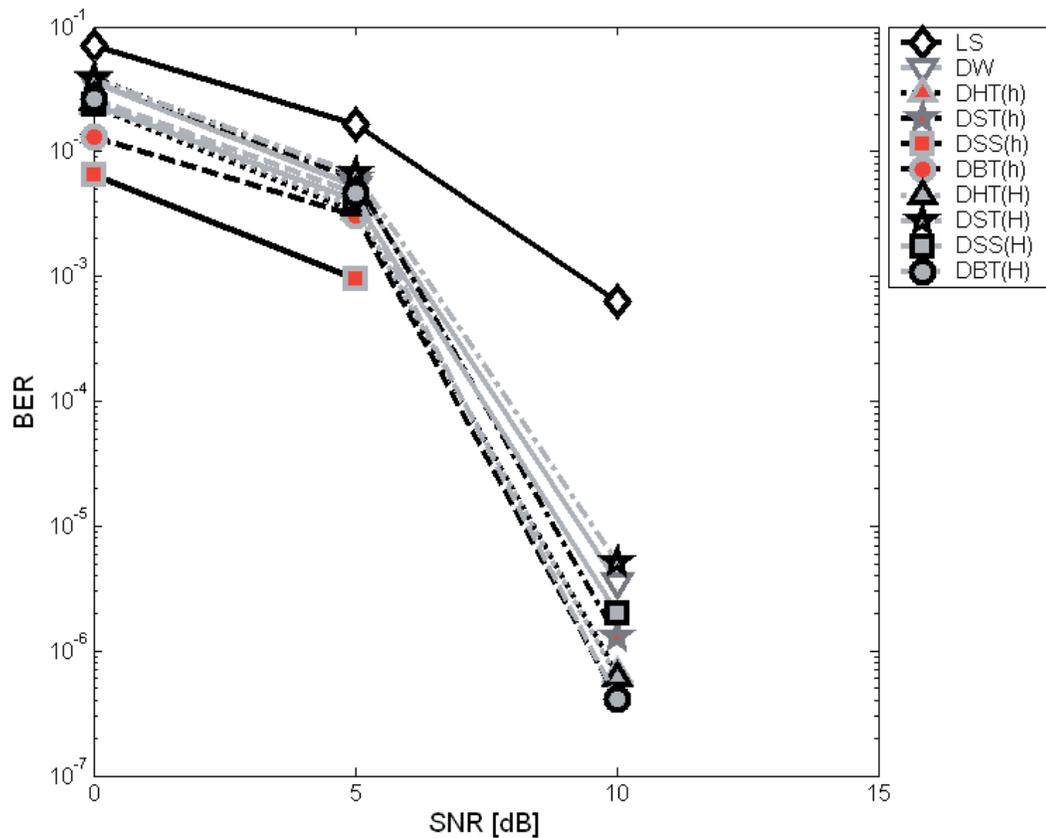


Figura 5.13: Escenario estacionario contaminado con ruido prácticamente Gaussiano (ruido impulsivo con una densidad de 10 pulsos por ciclo)

BER irreducible, lo que implica que la estimación del canal obtenida por la variante $DSS(h)$ resulta en mejores desempeños cuando los valores de SNR bajos que cuando son altos. Esto indica, en nuestro caso particular, que las aproximaciones del canal (variante en el tiempo) obtenidas por la variante $DSS(h)$, a partir de una observación muy ruidosa, son mejores que las aproximaciones obtenidas por la sola estrategia de seguimiento del canal regida por decisión (DD) que se emplea en nuestro transreceptor simulado.

Los resultados de las simulaciones nos permiten afirmar que en todos los escenarios (ya sean estacionarios o ciclo-estacionarios) y con todas las densidades de ruido impulsivo probadas (6000, 600 y 10 impulsos por ciclo de potencia) la variante $DSS(h)$ es la más eficaz, en términos de la probabilidad de error en bits (BER), especialmente en los ambientes altamente impulsivos (6000 impulsos por ciclo de potencia) y cuando la SNR se encuentra en el rango de 0 a 5 dB. Es muy interesante notar que un simple encogimiento de los coeficientes de ondeleta resulta en una estimación más eficaz, incluso con respecto a reglas de umbralado más complejas como, por ejemplo, la que emplea enfoque Bayesiano.

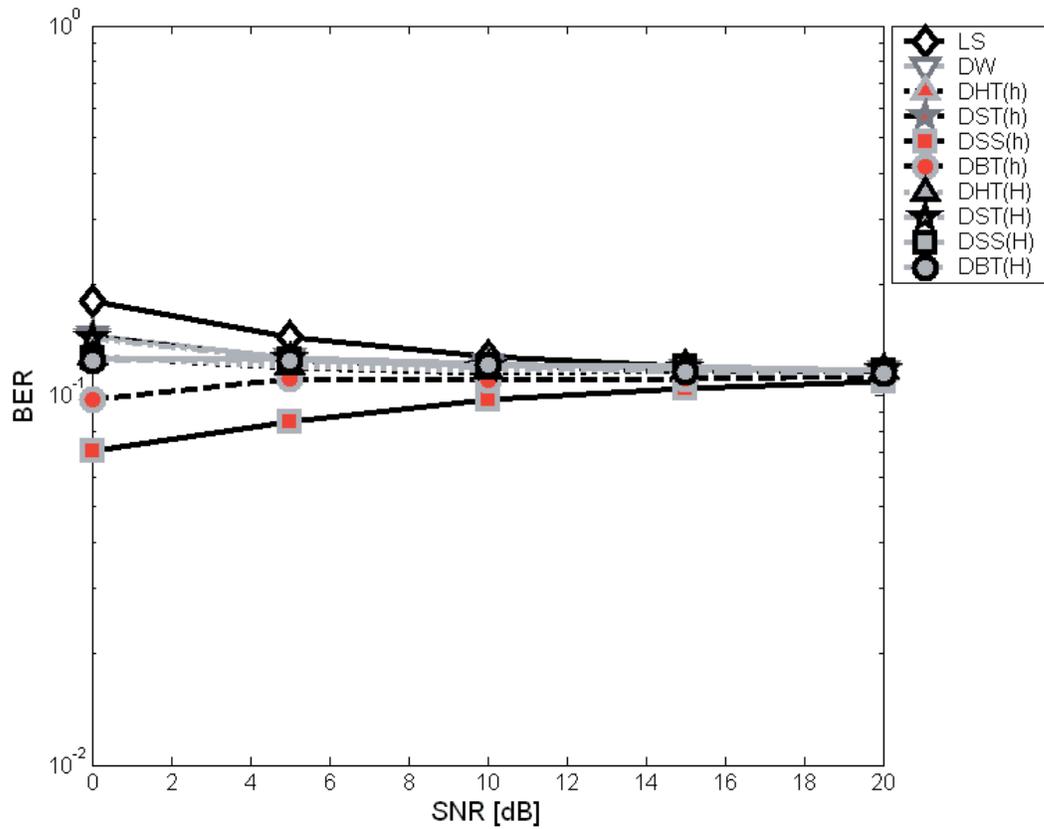


Figura 5.14: Escenario ciclo-estacionario contaminado con ruido prácticamente Gaussiano (ruido impulsivo con una densidad de 10 pulsos por ciclo)

La razón por la que la variante $DSS(h)$ enfrenta mejor la mezcla de ruido gaussiano más ruido impulsivo es que, al aplicar la regla de encogimiento para eliminar el ruido de la respuesta al impulso h , se trabaja en un dominio más conveniente para ello en virtud de que el ruido impulsivo es de alta intensidad y corta duración en el dominio temporal (ver Sección 2.4), mientras que en el dominio de la frecuencia su espectro es de banda ancha y una magnitud compensada para cumplir con el principio de Parseval. En este sentido, la capacidad de la ondeleta para eliminar transitorios de corta duración tiene natural aplicación en el dominio temporal. Por otra parte, el aplicar la regla de encogimiento permite suavizar los detalles en un rango mayor al restringido por el umbral implícito en la potencia promedio del ruido que, en efecto, está representando fundamentalmente a la componente del ruido gaussiano más que a la del ruido impulsivo. Entonces, al encoger a los coeficientes de detalle con amplitudes incluso mayores a las que considera un umbralado duro o un umbralado suave, tiene como resultado que se suavicen los detalles de alta intensidad que, de nuevo, es la característica del ruido impulsivo.

5.5. Conclusiones del capítulo

En este capítulo se ha propuesto una metodología para mejorar la estimación del canal LS aplicando los conceptos de estimación soportada por ondeletas. La aplicación de la metodología se centró en ambientes de comunicación sobre la línea eléctrica, que son contaminados por ruido Gaussiano y ruido impulsivo de gran densidad. Los resultados obtenidos muestran que la metodología de estimación del canal propuesta ofrece una forma simple de aproximar la característica del canal, produciendo una estimación mejorada del canal, sobre todo cuando el ruido impulsivo es denso y cuando la SNR es baja.

Los resultados obtenidos permiten destacar que la mejor técnica de estimación del canal soportada por la ondeleta para enfrentar eficazmente una mezcla de ruido Gaussiano y ruido impulsivo, en términos del desempeño del sistema medido con la probabilidad de error en los bits (BER), es la variante que emplea la regla de encogimiento propuesta por Donoho y Johnstone para eliminar el ruido partiendo de la respuesta al impulso h . Esto es especialmente cierto cuando el canal está contaminado por ruido impulsivo de gran densidad, pero también cuando la SNR es baja. Ya que la transformada de ondeleta discreta (DWT) es el fundamento de la metodología propuesta y debido a que ésta puede ser calculada muy rápido, la estimación del canal soportada por ondeletas mediante el enfoque de eliminación del ruido se vuelve una opción muy atractiva para mejorar la estimación de los canales PLC.

Finalmente, los resultados de las simulaciones realizadas dejan en claro la necesidad de contar con un método efectivo para el seguimiento de las variaciones temporales del canal, ya que lo contrario se traduce en una disminución en el desempeño del sistema que se caracteriza por un nivel significativo e irreducible para la probabilidad de error en los bits.

Parte II

Estimación del canal PLC, heurísticas bio-inspiradas y GPU

Como se comentó en la introducción, durante el desarrollo de esta tesis se siguieron tres líneas de investigación en lo relativo a maximizar el aprovechamiento del canal mediante el abordamiento del problema de la estimación del canal PLC con diferentes enfoques. La primera línea de investigación considera la aplicación de técnicas avanzadas de procesamiento como lo es la descomposición de ondeletas, cuyos resultados se presentan en la primera parte de esta tesis. La segunda línea de investigación se centra en el enfoque que propone enfrentar el problema de la estimación del canal mediante la aplicación de heurísticas bio-inspiradas, motivados por la exitosa aplicación de estas heurísticas en diversas disciplinas. Finalmente, la tercera línea de investigación involucra la incursión en nuevas arquitecturas de cómputo masivo que derivó en el empleo de dispositivos de procesamiento de gráficos (GPU), lo anterior a fin de superar algunas limitaciones de cómputo previstas para el caso de la aplicación de heurísticas bio-inspiradas al problema de la estimación del canal. Los resultados de las dos últimas líneas de investigación se presentan en esta segunda parte de la tesis.

Capítulo 6

Estimación semi-ciega del canal PLC mediante un algoritmo inspirado en PSO

Considerando la necesidad de contar con un método efectivo para seguir las variaciones temporales del canal en un entorno plagado de ruido y en virtud de la exitosa aplicación de las heurísticas bio-inspiradas en diversas disciplinas de la ciencia aplicada, así como la escasez de trabajos publicados en lo relativo a la aplicación de estas heurísticas al campo de la estimación del canal (por ejemplo, ver [4]), se exploró el potencial de algunos algoritmos poblacionales bio-inspirados aplicados para este propósito.

En este capítulo se propone una nueva técnica de estimación semi-ciega del canal para sistemas de comunicación OFDM sobre las líneas eléctricas plagadas de ruido impulsivo. La técnica propuesta está inspirada en las propiedades de auto-organización y rastreo de las poblaciones PSO. El algoritmo de optimización por cúmulo de partículas (PSO, por sus siglas en inglés) es una técnica poblacional bio-inspirada que se emplea en este capítulo para rastrear semi-ciegamente la respuesta del canal de la línea eléctrica, el cual es variante en el tiempo y está afectado por ruido gaussiano e impulsivo. Los resultados de las simulaciones numéricas para una red doméstica muestran el potencial del procedimiento de estimación propuesto en entornos ruidosos.

6.1. Introducción

Respecto de la estimación ciega de canales PLC, en [133] se propone un estimador ciego del canal con base en una red neuronal Kohonen. Motivados por ese trabajo, ahora aquí se propone un nuevo estimador semi-ciego del canal (es decir, que va siguiendo los cambios del canal), empleando una técnica inspirada en el algoritmo de optimización por cúmulo de partículas (*Particle Swarm Optimization*). El comportamiento en el desplazamiento de una población PSO implícitamente la vuelve una población rastreadora y seguidora que puede ser vista como un cúmulo auto-organizado [112], en analogía con un mapa auto-organizado de tipo Kohonen el cual puede agrupar datos empleando aprendizaje no supervisado. Con todo esto, parece una consecuencia natural el considerar la adaptación de algoritmo PSO para realizar el seguimiento del canal.

La técnica PSO es bio-inspirada y originalmente fue propuesta como una alternativa para resolver difíciles problemas de optimización, obteniendo resultados prometedores [78]. La idea básica de esta técnica es la de trabajar con poblaciones de individuos que simultáneamente buscan la mejor solución, con base en reglas específicas y procesos estocásticos subyacentes. Esta técnica heurística ha encontrado exitosa aplicación en prácticamente todos los campos de conocimiento y ahora, en este capítulo, se propone un método inspirado en PSO para la estimación semi-ciega y seguimiento del canal en sistemas OFDM para la línea eléctrica [84]. El algoritmo básico PSO tomado como referencia se detalla en el Apéndice G

6.2. Método de estimación semi-ciega propuesto

En general, en un sistema OFDM los símbolos de modulación con amplitudes en cuadratura¹ (QAM, por sus siglas en inglés) son repartidos en bloques y, a su vez, los datos dentro de estos bloques son transmitidos paralelamente a través de muchas subportadoras que ocupan sub-bandas tan angostas que el correspondiente subcanal puede considerarse con una respuesta en frecuencia prácticamente llana (*flat channel*). Entonces la señal recibida para cada $m^{\text{ésimo}}$ símbolo OFDM, puede expresarse como

$$\mathbf{y}_m = \mathbf{H}\mathbf{x}_m + \mathbf{n}_m \quad (6.1)$$

donde $\mathbf{y}_m = [y_0 \ y_1 \ \dots \ y_{N-1}]^T$ es el vector de símbolos recibidos de modulación QAM; $\mathbf{H} = \text{diag}(\mathbf{h}_m)$, con $\mathbf{h}_m = [h_0 \ h_1 \ \dots \ h_{N-1}]^T$, el vector de coeficientes complejos de atenuación del canal; $\mathbf{x}_m = [x_0 \ x_1 \ \dots \ x_{N-1}]^T$ el vector de símbolos de modulación transmitidos; y \mathbf{n}_m es el vector de ruido (mezcla de ruido Gaussiano más impulsivo).

Sin menoscabo del caso general, se asume que los símbolos de modulación transmitidos pertenecen a una constelación QPSK, es decir $x_i \in \{1, j, -1, -j\}$, con i el índice de la subportadora. Si se asume un canal estacionario, los valores complejos recibidos en una subportadora i dada pueden pertenecer a alguno de los cuatro agrupamientos en el plano complejo, con centroides vigentes

$$\{C_{1,i}, C_{2,i}, C_{3,i}, C_{4,i}\}.$$

Sin embargo, si la respuesta en frecuencia del canal es variante en el tiempo, los coeficientes de atenuación del canal pueden cambiar en sucesivos símbolos OFDM, lo cual resulta en un desplazamiento de los centroides hacia una nueva posición. Enseguida se describe un nuevo algoritmo rastreador y seguidor, inspirado en la metáfora del algoritmo PSO, para dar seguimiento a los centroides en movimiento y poder detectar los símbolos recibidos.

La idea principal del método propuesto, siguiendo la metáfora de PSO, es considerar a cada agrupamiento de símbolos recibidos como un cúmulo, los símbolos como partículas, y sus centroides como la mejor estimación para los correspondientes símbolos de modulación recibidos.

¹En la modulación por amplitudes en cuadratura, QAM, se modulan dos subportadoras de igual frecuencia pero desfasadas en 90° , lo que puede representarse en forma compleja con una componente en fase y otra en cuadratura.

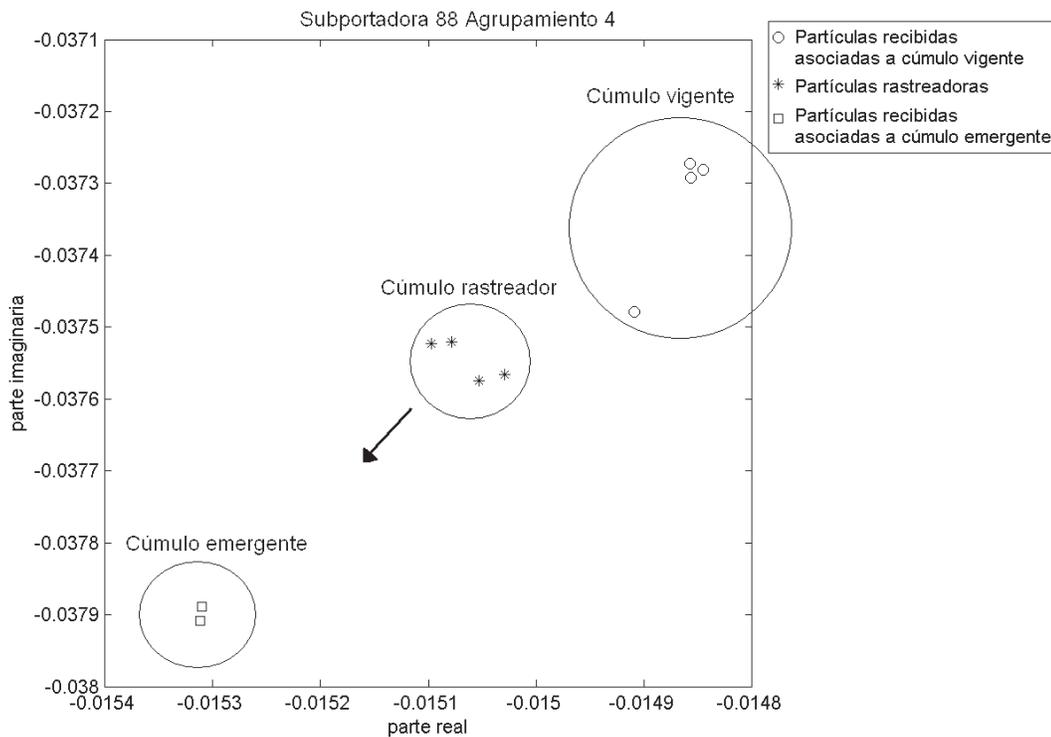


Figura 6.1: Creación de cúmulo emergente y cúmulo rastreador

Teniendo en mente la metáfora de los cúmulos de partículas, en el resto de esta sección las palabras *agrupamiento* y *cúmulo* serán usadas indistintamente como sinónimos. Entonces, el algoritmo propuesto consiste en asignar a cada cúmulo vigente (es decir, agrupamiento vigente) dos cúmulos temporales: un *cúmulo emergente* y un *cúmulo rastreador* (ver Figura 6.1). El cúmulo emergente es creado cuando se detecta que la posición de la partícula recibida se encuentra fuera de la región del cúmulo vigente (llamada, por lo tanto, *partícula despedida*).

El límite de la región de un cúmulo dado está determinado mediante una simple distancia Euclidiana, de tal manera que el cúmulo emergente forma un nuevo agrupamiento con las partículas despedidas. Eventualmente, si la tendencia muestra un crecimiento irreversible del cúmulo emergente, entonces el centroide del cúmulo emergente reemplaza al centroide vigente siempre y cuando se compruebe que las partículas despedidas no son eventos aislados debidos a ruido. El cúmulo rastreador sigue al centroide emergente partiendo desde el centroide vigente y con una velocidad de vuelo suficientemente lenta para verificar que el cúmulo emergente es confiable como nueva posición para el cúmulo vigente (ver Figura 6.2). Durante el proceso de rastreo, se asume que la mejor estimación para los correspondientes símbolos de modulación recibidos es el centroide del cúmulo rastreador. Es necesario notar que, de estos tres cúmulos (es decir, el vigente, el emergente y el rastreador), el cúmulo rastreador es el único que se construye y se desplaza conforme a las reglas de PSO detalladas en el Apéndice G. Los cúmulos vigentes y emergentes se constituyen únicamente por las partículas arribadas (es decir, los símbolos recibidos). Así, para tramas de datos

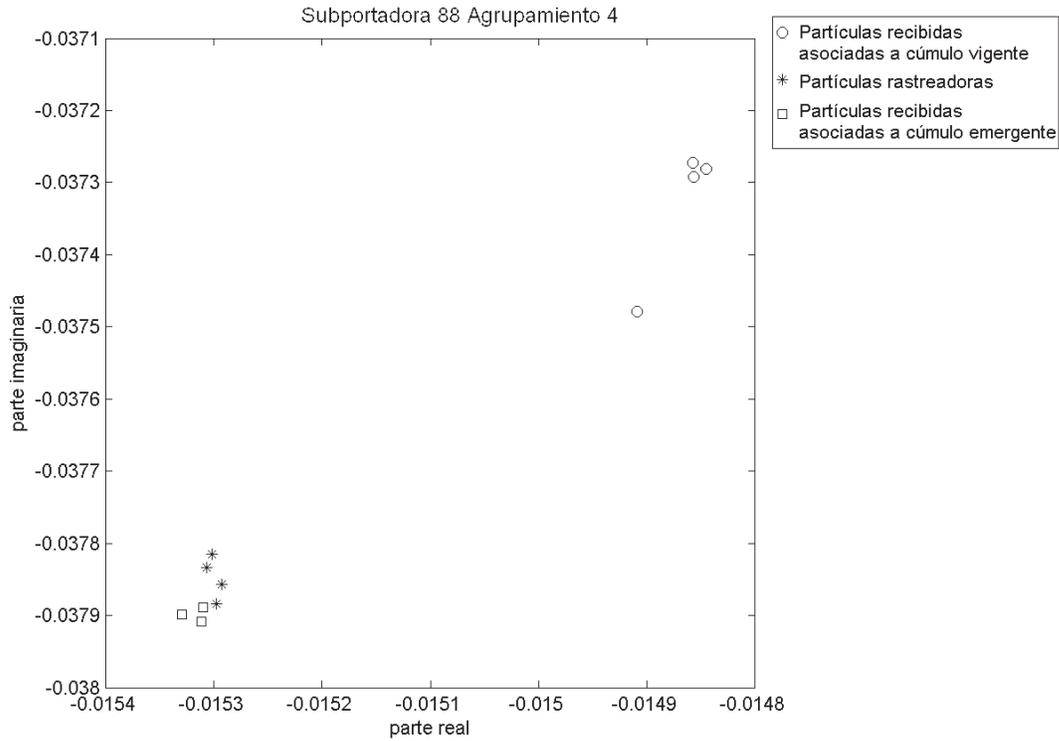


Figura 6.2: Atracción del cúmulo rastreador por el cúmulo emergente después de algunas iteraciones del algoritmo

compuestas por símbolos OFDM, el algoritmo de rastreo inspirado en PSO puede ser resumido como sigue:

1. Ubicar la posición inicial de los agrupamientos con ayuda de una técnica asistida por símbolos pilotos. Una vez recibido un símbolo OFDM piloto, para cada subportadora i , se procesa un símbolo recibido (que en lo subsiguiente será llamado *partícula arribada*), a fin de detectar a qué clase, dentro de la constelación de posibles símbolos de modulación, pertenece y, por lo tanto, determinar el k –ésimo agrupamiento al que pertenece. Luego se procede a determinar la posición de los centroides vigentes ($C_{k,i}^m$) y se delimitan las regiones de todos los K cúmulos vigentes. En el plano complejo, la región de un cúmulo vigente puede ser delimitada por un círculo con su origen en el centroide correspondiente y radio,

$$R_{k,i}^m = \alpha \times \tilde{\delta}_{k,i}^m, \quad (6.2)$$

donde $\tilde{\delta}_{k,i}^m$ es la distancia promedio entre las últimas M partículas (p_i^m) asignadas al cúmulo vigente, y el centroide vigente ($\delta_{k,i}^m = \text{distance}(C_{k,i}^m, p_i^m)$). α es una constante determinada en forma empírica.

2. Agrupar las partículas arribadas. Después de que un símbolo OFDM piloto ha sido procesado, para los símbolos OFDM siguientes y para cada una de sus subportadoras, se procesan las

partículas arribadas como se indica a continuación. Si la posición de la partícula yace dentro de la región del cúmulo más cercano ($\delta_{k,i}^m \leq R_{k,i}^m$), entonces se asigna al cúmulo vigente y se actualiza la posición del centroide ($C_{k,i}^m$) con las últimas M partículas. Después, usar el centroide resultante como la mejor estimación para el símbolo de modulación recibido que pertenece al cúmulo actualizado y, por lo tanto, vigente.

3. Detectar posibles desplazamientos de los agrupamientos. Si una partícula arribada se encuentra cerca de la región de un cúmulo pero fuera de ésta ($\delta_{k,i}^m > R_{k,i}^m$), entonces se clasifica como una *partícula despedida* y se asigna al cúmulo emergente correspondiente. Si no existe el cúmulo emergente, entonces debe ser creado uno. Luego, el cúmulo emergente es alimentado con las siguientes partículas despedidas. Cuando se crea un cúmulo emergente también se crea su cúmulo rastreador correspondiente, con una población de M partículas, y se habilita un proceso de rastreo. También se calcula el centroide emergente ($E_{k,i}^m$) y la distancia promedio $\tilde{\epsilon}_{k,i}^m$ entre éste y las últimas M partículas despedidas. Durante la creación del cúmulo rastreador, se inicializa la posición de cada partícula rastreadora en forma aleatoria alrededor del centroide vigente, pero separado de éste en una distancia que debe ser menor al valor absoluto del parámetro PSO denominado como “velocidad máxima”, la cual definiremos en nuestro caso como

$$v_{\max} = \frac{1}{\beta} \text{distance}(C_{k,i}^m, E_{k,i}^m), \quad (6.3)$$

donde β es una constante determinada en forma empírica.

4. Refrescar el registro temporal para el proceso de rastreo. Durante la creación de los cúmulos rastreadores, por cada uno de ellos se habilitan automáticamente dos temporizadores de expiración: el temporizador propio del cúmulo emergente (t_E) y el temporizador asociado al cúmulo vigente (t_C), ambos inicializados con el mismo tiempo de expiración ($t_E = T_{\text{exp}}$ y $t_C = T_{\text{exp}}$). Los temporizadores son regresivos y expiran cuando la cuenta se hace cero. Durante el proceso de rastreo, si un temporizador dado expira antes de que su cúmulo asociado sea alimentado por alguna partícula, entonces dicho cúmulo muere. En contra parte, cada vez que una partícula es asignada a un cúmulo se restablece su temporizador con un nuevo tiempo de expiración T_{exp} . Específicamente, cuando un cúmulo emergente muere, también muere el cúmulo rastreador, y el centroide vigente vuelve a ser confiable. De otra manera, si el cúmulo vigente muere, entonces el cúmulo emergente se convierte en el nuevo cúmulo vigente. Ya que un cúmulo emergente funge como un atractor para el correspondiente cúmulo rastreador, el proceso de rastreo asociado se detiene automáticamente después de que muere un cúmulo emergente.
5. Ejecutar una iteración PSO. Durante el proceso de rastreo, y una vez que se completan los pasos 2-4, los cúmulos rastreadores habilitados ejecutan una sola iteración PSO por cada símbolo OFDM. Como en nuestro caso no existe una función de costo explícita, la actualización de la velocidad y posición de las partículas debe ser hecha de acuerdo con las reglas descritas en el Apéndice G, pero tomando en cuenta las siguientes reglas especiales:

- Hacemos g_{best} igual a la posición del centroide emergente, como primer atractor.

- Hacemos p_{best} igual a la posición del centroide vigente, como segundo atractor.
- Mantenemos a r_1 como variable aleatoria entre 0 y 1, pero ahora r_2 está relacionada a r_1 de la siguiente manera,

$$r_2 = 1 - r_1 \quad (6.4)$$

- Las constantes de ponderación, c_1 y c_2 , ahora representan la influencia relativa de los respectivos atractores de acuerdo con las siguientes expresiones heurísticas:

$$c_1 = \frac{t_C}{T_{\text{exp}}} \cdot \left(1 - \frac{\tilde{\delta}_{k,i}^m}{\tilde{\delta}_{k,i}^m + \tilde{\epsilon}_{k,i}^m}\right) \quad (6.5)$$

$$c_2 = \frac{t_E}{T_{\text{exp}}} \cdot \left(1 - \frac{\tilde{\epsilon}_{k,i}^m}{\tilde{\delta}_{k,i}^m + \tilde{\epsilon}_{k,i}^m}\right) \quad (6.6)$$

6. Seleccionar el centroide apropiado con fines de detección. Para cada subportadora y para cada símbolo dentro de la constelación de modulación (centroide), revisar si se ha habilitado un proceso de rastreo. Si un proceso de rastreo está en progreso, entonces se escoge el centroide del cúmulo rastreador como la mejor estimación para el correspondiente símbolo de modulación recibido. En otro caso, simplemente se usa al centroide del cúmulo vigente como la mejor estimación.
7. Repetir los pasos 2-6 mientras se reciban símbolos OFDM. En caso de recibir un símbolo piloto se va al paso 1 y se comienza de nuevo.

6.3. Resultados numéricos

Se consideró la red simple con un nodo en T (un canal con una derivación), como se muestra en la Figura 2.7 del Capítulo 2, a fin de comparar el método propuesto con el método lineal asistido por símbolos pilotos. El entorno de simulación para el canal PLC es el especificado en la sección 2.6, mientras que el transreceptor se simula con base en el sistema básico especificado en 2.7 pero se emplea una versión no codificada. Los valores probados de densidad del ruido impulsivo fueron 10, 60, 600 y 6000 impulsos por ciclo de la línea.

Respecto de la relación de potencias entre el ruido Gaussiano y el ruido impulsivo dada en (2.33), ésta se fijó en $\Gamma = 0,00001$ y se mantuvo constante a lo largo de todas las simulaciones. Este valor representa ruido impulsivo de alta energía, que se traduce en picos de magnitud significativamente mayor al promedio de la señal útil. El valor $\Gamma = 0,00001$ garantiza que la potencia espectral del ruido impulsivo pueda superar por más de 40 dB al ruido Gaussiano, como se reporta en [9].

Los parámetros PSO también fueron mantenidos constantes durante todas las simulaciones y sus valores fueron los siguientes: tanto el tamaño de la memoria como el número de partículas fue $M = 4$; $\alpha = 5$ en (6.2), para definir el umbral de pertenencia; $\beta = 8$ en (6.3), para definir v_{max} ;

$v_{\min} = -v_{\max}$; $w_{\max} = 0,9$; $w_{\min} = 0$; y $T_{\text{exp}} = 8(T_s + T_g)$ (ver Cuadro 2.1). Note que, en la práctica, los temporizadores pueden ser programados como simples contadores decrecientes, en virtud de que la duración de los temporizadores puede ser expresada en términos de un número de símbolos OFDM transcurridos.

6.3.1. Sistema OFDM de referencia

En este caso se simuló un sistema DMT no codificado donde la detección fue realizada con una técnica regida por el proceso de decisión (DD). La razón para emplear un esquema no codificado responde a la necesidad de medir el desempeño de la técnica propuesta sin ayuda de un esquema de corrección de errores. El número real de subportadoras empleadas estuvo determinado por la atenuación del canal, estimada por la trama de datos previa. Específicamente, el nivel de conmutación fue colocado en -35 dB.

6.4. Discusión de los resultados

Las simulaciones del método propuesto resultaron ser muy demandantes de recursos computacionales y el tiempo consumido hizo imposible la ejecución de un número suficiente de corridas Monte Carlo a fin de obtener curvas más suaves para la probabilidad de error en los bits (BER), que es la figura de mérito empleada para evaluar el desempeño del método de estimación y seguimiento del canal propuesto. No obstante, los resultados obtenidos arrojan prometedores indicios de la capacidad de un algoritmo poblacional para enfrentar ambientes ruidosos.

El desempeño del método propuesto para la estimación y seguimiento del canal PLC, inspirado en PSO, comparado contra una técnica lineal clásica (estimación de canal LS) asistida por pilotos, se muestra en las Figuras 6.3 y 6.4 para densidades de ruido impulsivo de 60 y 6000 impulsos por ciclo de la línea, respectivamente. Las Figuras 6.3 y 6.4 son representativas, en términos del impacto del ruido impulsivo porque éstas son los casos extremos de los experimentos realizados. En general, es evidente que el desempeño total del sistema DMT está fuertemente afectado por los niveles de atenuación simulados (debajo de -40 dB, en promedio) como se puede apreciar en la Figura 2.13. Con respecto a la relación señal a ruido (SNR), es muy interesante observar que si la $SNR < 10$ dB entonces el desempeño del método inspirado en PSO es ligeramente mejor que el lineal. No obstante, se observa un punto de cruce alrededor de una SNR de 10dB, lo cual indica que el desempeño del método propuesto es peor que el lineal cuando la SNR es grande. Sin embargo, es interesante notar que el método propuesto no solo tiene un mejor desempeño con valores de SNR bajos sino que, incluso, mejora conforme se incrementa la densidad del ruido impulsivo, lo cual es una característica prometedora considerando el entorno tremendamente ruidoso del canal PLC.

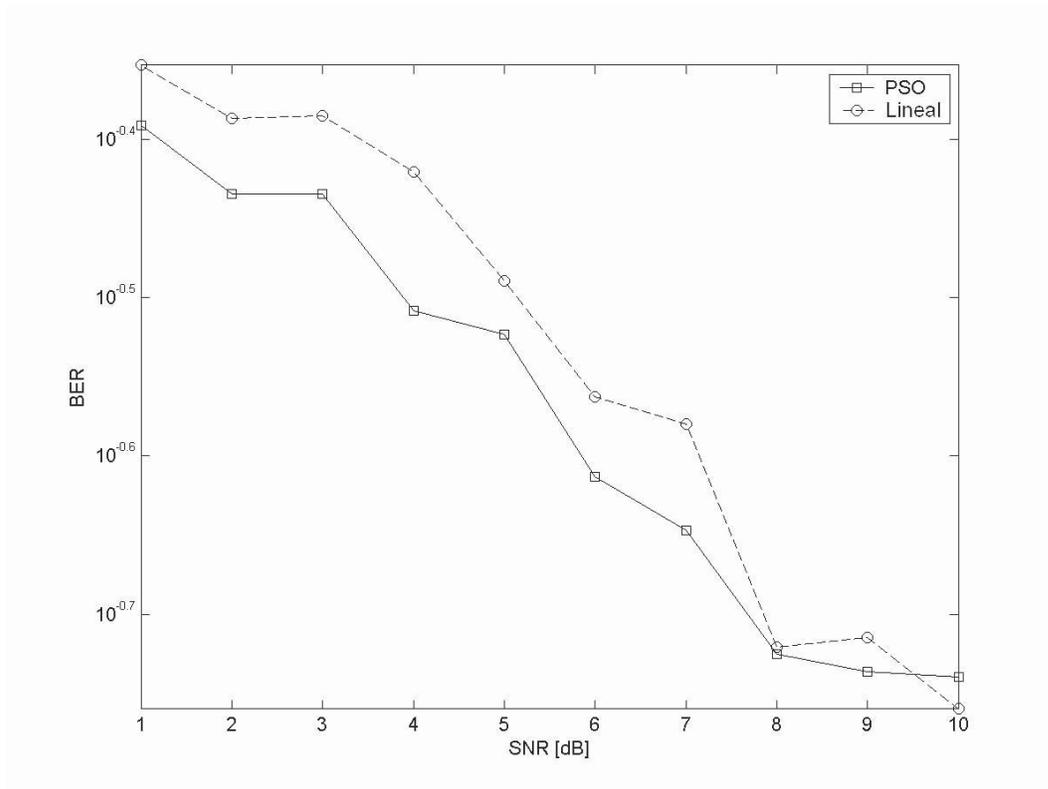


Figura 6.3: Comparación de la probabilidad de error entre el método de estimación lineal y el inspirado en PSO, con una densidad de ruido impulsivo de 60 impulsos por ciclo de línea

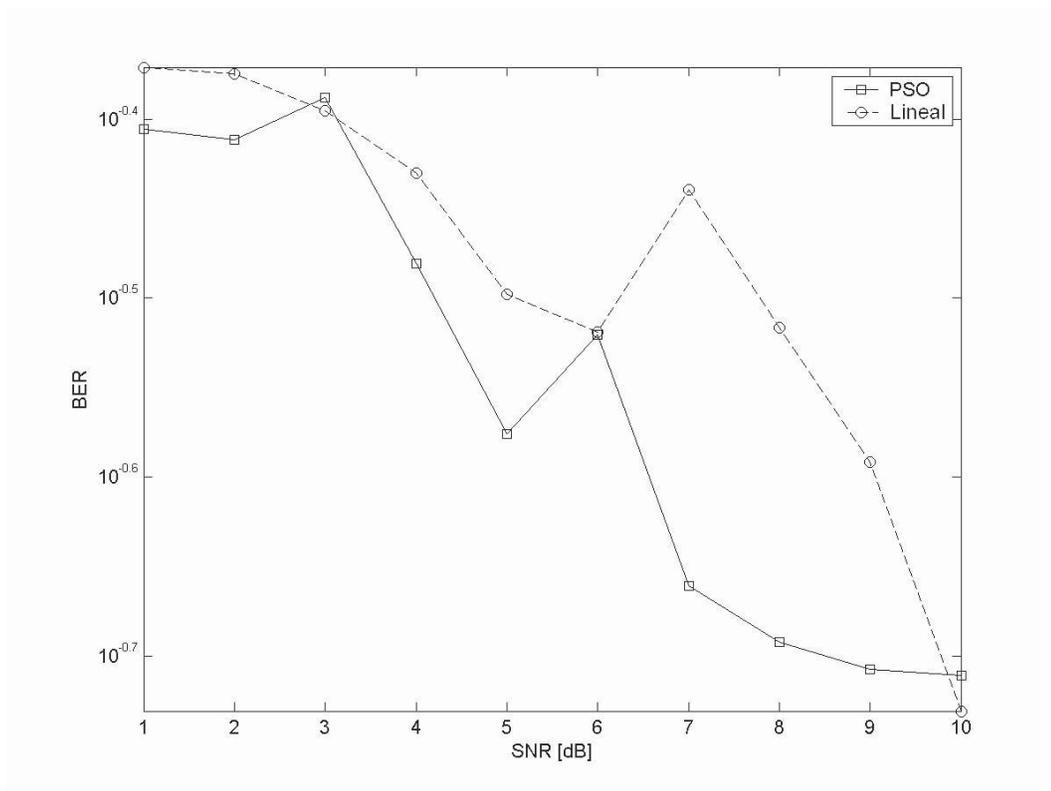


Figura 6.4: Comparación entre el método de estimación lineal y el inspirado en PSO, con una densidad de ruido impulsivo de 6000 impulsos por ciclo de línea

6.5. Conclusiones del capítulo

En este capítulo se ha propuesto una nueva técnica de estimación semi-ciega, para el canal PLC, inspirada en las características auto-organizativas y de rastreo de las poblaciones de optimización por cúmulo de partículas (PSO). Los resultados de las simulaciones muestran que el método propuesto enfrenta bien al ruido impulsivo por lo que tiene un potencial prometedor en aplicaciones para canales de comunicación selectivos en frecuencia y variantes en el tiempo, tal y como lo son los canales de las líneas eléctricas. Sin embargo, las simulaciones resultaron ser muy demandantes de recursos computacionales (memoria y tiempo), lo cual plantea un obstáculo para su realización práctica en un sistema de tiempo real con las tecnologías actualmente disponibles de circuitos de procesamiento de señales (DSP). Resulta, por lo tanto, de gran interés explorar las posibilidades de otras arquitecturas de gran poder de cómputo y procesamiento que en un futuro puedan ofrecer sus beneficios al campo de los DSP.

Capítulo 7

Programación paralela del algoritmo PSO en un dispositivo de procesamiento de gráficos con múltiples hilos

Como se vio en el capítulo anterior, una desventaja de las heurísticas poblacionales bio-inspiradas, y por lo tanto obstáculo para su aplicación práctica a nivel de los módems de comunicación PLC, es que pueden resultar muy demandantes de recursos computacionales. A propósito de esto, en este capítulo se presenta un estudio sobre el desempeño de la heurística de optimización por cúmulo de partícula (PSO, por sus siglas en inglés) en una nueva arquitectura de cómputo conocida como unidad de procesamiento de gráficos (GPU, por sus siglas en inglés) que permite acelerar considerablemente la velocidad de los algoritmos gracias a un nuevo modelo de programación paralela que promete una revolución en el estilo de programación a todos los niveles, incluso al nivel de los dispositivos de procesamiento de señales.

El algoritmo de optimización por cúmulo de partícula (*Particle Swarm Optimization*) es una alternativa bien conocida para la optimización global basada en una heurística poblacional bio-inspirado. El algoritmo PSO tiene buen desempeño, baja complejidad computacional y pocos parámetros. Las técnicas heurísticas han sido extensamente estudiadas en los últimos veinte años y la comunidad científica aún está interesada en alternativas tecnológicas que aceleren estos algoritmos a fin de aplicarlos a problemas más grandes y más complejos. Este capítulo presenta un estudio empírico de algunas variantes paralelas para un algoritmo PSO, puestas en práctica en una unidad de procesamiento de gráficos (*Graphics Processing Unit*) con capacidad de administrar de múltiples hilos y en conjunto con el modelo más reciente de programación paralela empleado en estos casos. La idea principal es mostrar que, con la ayuda de un GPU multi-hilos, es posible mejorar considerablemente el desempeño de algoritmo PSO por medio de una programación paralela simple y directa, consiguiendo el poder de cómputo de un *cluster* en una computadora personal convencional.

7.1. Antecedentes

Algunas técnicas bio-inspiradas, como el cómputo evolutivo [54], optimización por colonia de hormigas [51] y PSO [78], fueron propuestas como alternativas para solucionar problemas de optimización difíciles donde obtuvieron soluciones aceptables en un tiempo razonable. Debido a que estas técnicas trabajan con poblaciones de individuos, pueden probar simultáneamente diferentes soluciones con base a ciertas reglas específicas y procesos estocásticos subyacentes. Estas técnicas heurísticas han sido aplicadas en prácticamente todos los campos del conocimiento, obteniendo un buen desempeño incluso al ejecutarse en procesadores convencionales.

Las técnicas heurísticas obtienen soluciones aceptables en un tiempo “razonablemente corto” comparadas con los métodos tradicionales, como las técnicas deterministas y enumerativas que pueden resultar poco prácticas, especialmente al solucionar problemas de optimización difíciles, ya que estas últimas exploran cada una de las posibles soluciones de un problema. Sin embargo, el tiempo “razonable” que las técnicas heurísticas pueden consumir puede estar en la orden de los segundos, minutos, o hasta horas, según el problema y el procesador empleado. De este modo, los algoritmos más simples, como lo es PSO, se han hecho muy atractivos debido a su baja complejidad computacional que resulta en tiempos de ejecución más cortos. Sin embargo, cuando es necesario obtener una solución de tiempo real buena, hasta los algoritmos más simples pueden parecer ser lentos. Esta situación ha motivado la búsqueda de nuevas formas de acelerar el desempeño de estas heurísticas.

Recientemente, se propuso explotar el poder computacional disponible en las tarjetas gráficas de las computadoras personales a fin de solucionar problemas de propósito general [115], surgiendo, con ello, la idea del GPU para procesamiento de propósito general (GPGPU, por sus siglas en inglés). Desde entonces, tanto los fabricantes como los desarrolladores han considerado esta nueva aplicación de la computación como una prometedora área de investigación, considerando la amplia variedad de posibles aplicaciones que pueden aprovechar el paralelismo disponible en los actuales GPU.

Desde que los primeros algoritmos bio-inspirados fueron reportados, la idea de su programación paralela fue vista como una consecuencia natural de su condición poblacional [28]. Para el caso de un algoritmo PSO, podemos usar los modelos paralelos desarrollados en su momento para los algoritmos evolutivos, es decir: el modelo global, el modelo de isla y el modelo de difusión [16]. En este capítulo se presentan algunas variantes paralelas para el algoritmo PSO (dos variantes Globales y una que se denominó como incrustada) puestas en práctica en un GPU multi-hilos. Se reportan los resultados obtenidos usando el nuevo modelo de programación CUDA (Arquitectura unificada de dispositivo computador), que permite al programador escribir código orientado a hilos, además de operaciones paralelas en la memoria gráfica [111]. La idea principal es mostrar que, con la ayuda de un GPU multi-hilos, es posible mejorar el desempeño de algoritmo PSO, en forma considerable, por medio de una programación paralela simple y directa, con lo que se logra el poder computacional de un *cluster* en un computadora personal convencional.

7.2. Trabajo relacionado

La programación paralela, por lo general, implica la migración de un código secuencial existente hacia alguna arquitectura concurrente, paralela o distribuida. El algoritmo PSO secuencial no fue la excepción y se encuentran reportes sobre las primeras tentativas de aprovechar su natural paralelismo después de su presentación en 1995 [78] (por ejemplo, el trabajo de J.F. Schutte en 2003 [127]). La programación paralela del algoritmo PSO es todavía un tema muy actual, lo que se comprueba por la reciente publicación de trabajos de investigación donde se aplican algoritmos PSO paralelos para solucionar problemas de optimización muy complejos (por ejemplo, ver [95]).

En la literatura especializada, podemos encontrar, para cada algoritmo, propuestas de paralelismo con diversos enfoques. Aunque un enfoque tradicional es el de procesos concurrentes que corren en sólo un procesador (ver [14]), la mayor parte de las realizaciones paralelas son diseñadas para ser ejecutadas en sistemas distribuidos (es decir varios procesadores en una red). En los sistemas distribuidos la sobrecarga (*overhead*) por comunicación entre los diferentes procesadores es un factor que afecta significativamente el desempeño de la aplicación paralela. A causa de esto, es comprensible que algunas realizaciones paralelas para PSO fueran propuestas adoptando estrategias de comunicación. Incluso, es posible encontrar nuevas propuestas para la programación paralela del algoritmo PSO con el enfoque denominado como de “vectores paralelos”.

Finalmente, debemos notar que, tradicionalmente y por razones históricas, la mayor parte de los trabajos pioneros sobre programación paralela de algoritmos poblacionales en GPU se centraron en los algoritmos evolutivos y genéticos (ver [60]). En algunos casos las experiencias obtenidas fueron aplicadas más tarde para la programación paralela del algoritmo PSO (como en [89]).

En la literatura especializada no se menciona ningún estudio empírico referente a la realización de alguna heurística poblacional en un GPU con el nuevo modelo de programación paralela, conocido como CUDA, que aproveche el enfoque de múltiples hilos concurrentes. Esto podría ser atribuido al hecho que estas herramientas de programación paralela fueron introducidas muy recientemente [111]. Por lo tanto, este trabajo representa el primer estudio empírico que compara algunas variantes paralelas de PSO para un GPU multi-hilos.

7.3. Introducción a los dispositivos GPU y a la arquitectura de multi-hilos

Los Dispositivos GPU modernos tienen su origen en la arquitectura del procesador vectorial, que permite la ejecución simultánea de operaciones matemáticas sobre datos múltiples. En contraste, los procesadores de los CPU convencionales no pueden manejar más de una operación al mismo tiempo. Al principio, los procesadores vectoriales eran comúnmente usados en las computadoras científicas [108], pero más tarde fueron desplazados por arquitecturas de múltiples núcleos. Sin embargo, los procesadores vectoriales no fueron completamente eliminados ya que, esencialmente, muchas de las arquitecturas diseñadas para realizar gráficas por computadora y dispositivos

GPU modernos están inspirados por ellos.

Por lo general, un procesador convencional tiene que buscar la siguiente instrucción por ejecutar, lo que consume el tiempo y genera cierta latencia durante la ejecución de las instrucciones. Para reducir la latencia, los procesadores modernos ejecutan un conjunto de instrucciones en forma paralela, lo que se conoce como “entubamiento de instrucciones” (*pipelining*). En un ciclo de máquina convencional se sigue la secuencia “traer-decodificar-ejecutar” (*fetch-decode-execute*), en cambio, con la técnica de entubamiento de instrucciones, la decodificación de la siguiente instrucción inicia antes de que se haya concluido la ejecución de la primera, de tal manera que el decodificador de instrucciones es constantemente utilizado y, con ello, disminuye la latencia. En estas condiciones, el tiempo de ejecución para un conjunto de instrucciones es menor que en un procesador convencional, aumentando así el rendimiento total del procesador.

Los procesadores vectoriales llevan la misma idea más lejos, al realizar el entubamiento tanto de las instrucciones como de los datos. En este caso, una sola instrucción opera en muchos elementos de datos, lo que ahorra tiempo en la decodificación de instrucciones y produce un gran poder de cálculo. Considerando que en la realización de gráficas por computadora las imágenes tienen una representación natural con el formalismo de las matrices y que es deseable contar con la capacidad de realizar operaciones paralelas sobre múltiples de datos, se comprende que los procesadores vectoriales hayan sido ampliamente adoptados para resolver estas exigencias de un modo directo. Lo arriba expuesto explica la extendida aceptación de los procesadores vectoriales en el área de la realización de gráficas por computadora y su influencia en el desarrollo de los modernos dispositivos GPU, destinados a satisfacer la gran demanda del poder computacional para el procesamiento de imágenes en las más recientes aplicaciones de vídeo-juegos.

7.3.1. Arquitectura del GPU

La tendencia actual observada en el desarrollo de los dispositivos GPU que realiza el fabricante NVIDIA permite anticipar la consolidación del nuevo modelo de programación paralela ofrecido con la herramienta de programación CUDA, donde el GPU, además de ofrecer una mayor capacidad de cálculo paralelo, tiene un papel más preponderante como administrador de múltiples hilos [111]. En nuestra investigación se utilizó un dispositivo NVIDIA GeForce 8600GT que en lo subsecuente será referido como el dispositivo GPU. Este GPU proporciona una arquitectura unificada, tanto para gráficos como para cálculos, que NVIDIA llama arquitectura Tesla y que es fundamentalmente un arreglo escalable de multiprocesadores multi-hilo.

Cada multiprocesador consiste en ocho núcleos de procesamiento, una unidad de instrucción multi-hilos y memoria compartida dentro del circuito integrado (*on-chip*) (ver Figura 7.1). Cada multiprocesador se encarga de la creación, manejo y ejecución de los hilos que están activos físicamente en el dispositivo, manejando cientos de hilos (teóricamente hasta 512) con el enfoque denominado por NVIDIA como “una instrucción y múltiples hilos” (SIMT, por sus siglas en inglés) [111]. Debido a que el multiprocesador asigna cada uno de los hilo a un núcleo y a que cada hilo es ejecutado independientemente de los demás, con su propia dirección de instrucción y sus propios

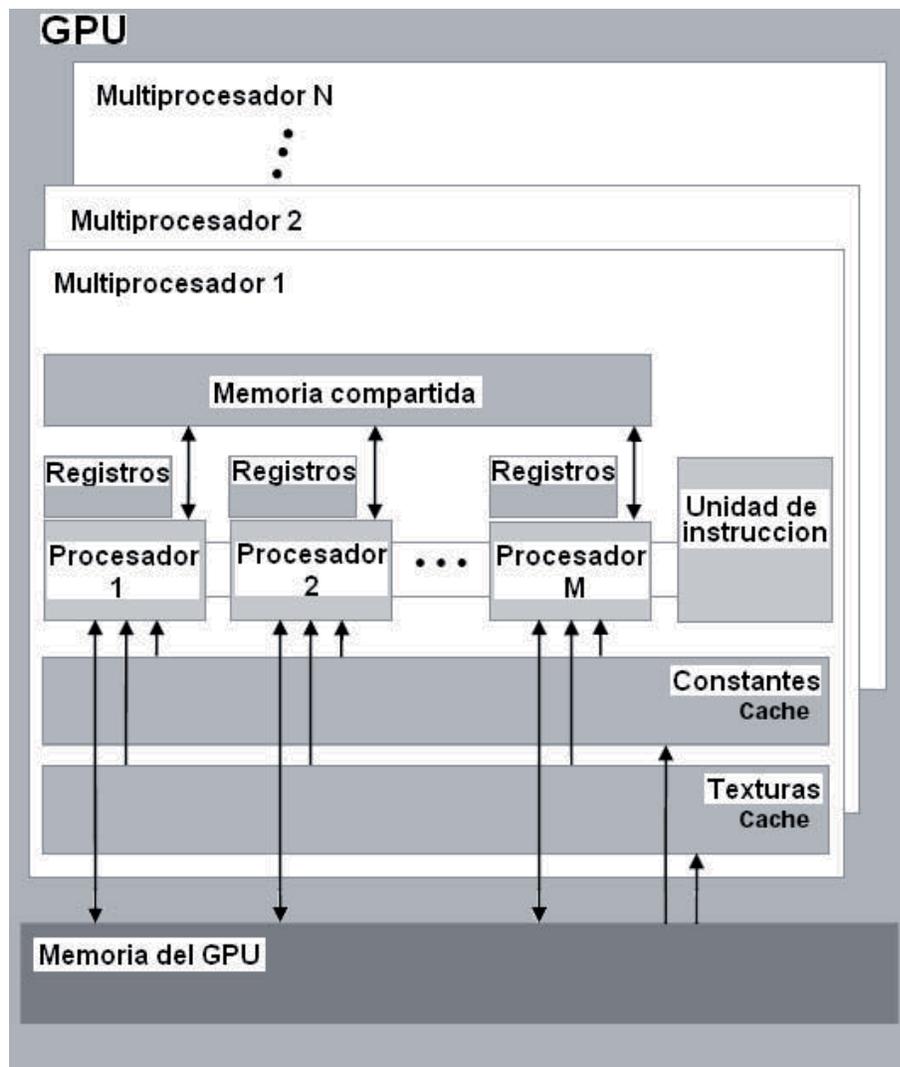


Figura 7.1: Arquitectura del GPU NVIDIA empleado [111]

registros de estado, las herramientas de programación de NVIDIA ofrecen algunas funciones que son enfocadas precisamente al manejo y optimización de múltiples hilos.

7.3.2. Arquitectura CUDA

La herramienta de programación conocida como CUDA está modelada por el enfoque de una instrucción y múltiples hilos (SIMT) donde múltiples hilos son ejecutados sobre muchos datos. La herramienta CUDA permite que los programadores escriban el código paralelo usando lenguaje C estándar mediante algunas extensiones de NVIDIA. La herramienta CUDA organiza el paralelismo en un sistema jerárquico de tres niveles: rejilla, bloque, e hilo. El proceso comienza cuando el procesador anfitrión (CPU) invoca una función del dispositivo GPU llamada *kernel*, luego se crea

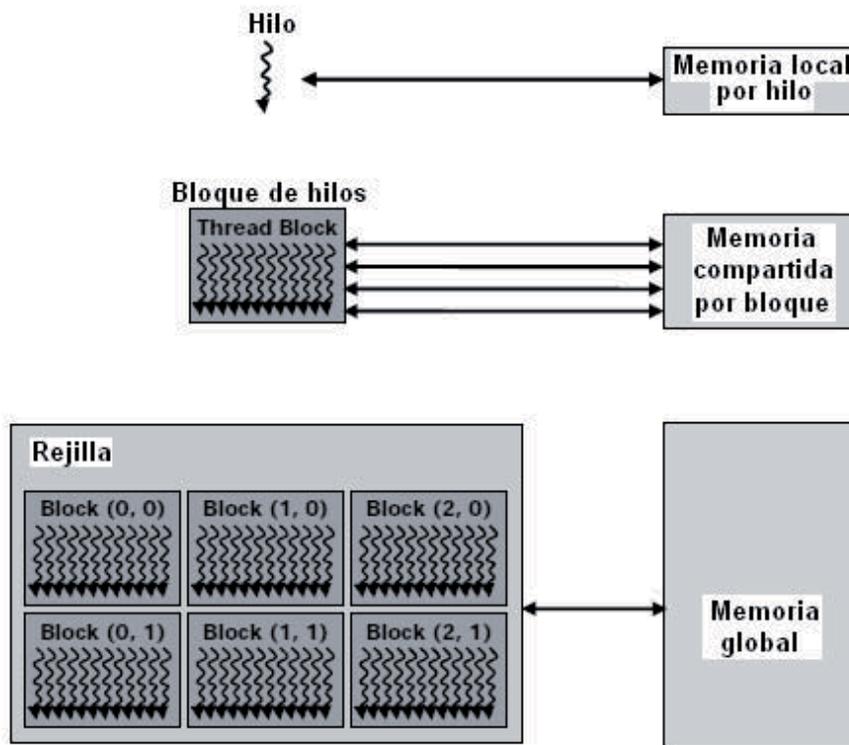


Figura 7.2: Jerarquía de la memoria del GPU NVIDIA [111]

una rejilla (arreglo) con bloques de múltiples hilos para luego distribuirla en los multiprocesadores disponibles. En CUDA, los programas arrancan la ejecución de los kernels paralelos mediante la siguiente sintaxis extendida para las llamadas a función:

```
kernel <<<dimGrid, dimBlock>>> ();
```

donde `dimGrid` y `dimBlock` son parámetros especializados que especifican, respectivamente, la dimensión, en bloques, de la rejilla de procesamiento paralelo, así como la dimensión, en hilos, de los bloques.

Durante la ejecución del kernel, los hilos tienen acceso a seis tipos de memoria en el GPU, según cierta jerarquía (o niveles de acceso) predefinida (ver Figura 7.2):

- Memoria global. Es una memoria para lectura/escritura y se localizada en la tarjeta del GPU.
- Memoria para constantes. Es una memoria rápida (*cache*) para lectura y se localiza en la tarjeta del GPU.
- Memoria para texturas. Es una memoria rápida (*cache*) para lectura y se localiza en la tarjeta del GPU.

- Memoria local. Es una memoria para lectura/escritura por hilo y se localiza en la tarjeta del GPU.
- Memoria compartida. Es una memoria para lectura/escritura por bloque y se localiza dentro del circuito integrado del GPU.
- Memoria de registros. Es la memoria más rápida para lectura/escritura por hilo y se localiza dentro del circuito integrado del GPU.

La memoria compartida y los registros son los más rápidos, pero su tamaño está limitado porque es memoria ubicada dentro del circuito integrado. Por otra parte, la memoria localizada en la tarjeta del dispositivo (local, global, para constantes y para texturas) es grande pero presenta mayor latencia en los accesos, en comparación con la memoria alojada dentro del circuito integrado. Ya que el multiprocesador ejecuta paralelamente los hilos en grupos de 32, llamados “tejidos” (*warp*), los hilos pueden tener un acceso más eficiente a la memoria global si se realiza en bloques de “mitades de tejidos” (*half-warp*), mediante lecturas/escrituras simultáneas de/a memoria fusionadas (*coalesced*) en una sola transacción de memoria, ya sea de 32, 64, o 128 bytes [111].

7.4. El algoritmo de optimización por cúmulo de partículas

El algoritmo PSO es una heurística poblacional que ha llamado la atención de muchos investigadores. En el Apéndice G se puede encontrar el algoritmo básico que fue empleado en este trabajo como referencia. En general, el algoritmo PSO es una estrategia por cooperación que encuentra una solución a consecuencia del movimiento de múltiples individuos que tratan de imitar al mejor individuo de cierta vecindad (local o global).

Específicamente, se programó una versión local (es decir, cada partícula tiene una vecindad local, sabe cuál partícula es la mejor localmente y qué partícula es la mejor en forma global) del algoritmo PSO, con vecindades de tamaño 20 para cada partícula, definidas aleatoriamente, y los siguientes parámetros, fijos para todas las funciones probadas:

$$c_1 = 1, \quad c_2 = 1, \quad v_{max} = 1, \quad v_{min} = 1,$$

7.5. Puesta en práctica de las variantes paralelas de PSO en el GPU

Al respecto de la clasificación de los modelos de programación paralela para el algoritmo PSO, se adoptó misma clasificación sugerida para los algoritmos evolutivos como en [16]:

1. Variante paralela con enfoque global. Existe un procesador principal (maestro) y varios esclavos. En este modelo, el maestro distribuye a los esclavos el trabajo relativo a la evaluación de la función de aptitud.

2. Variante paralela con el enfoque migratorio (también llamado enfoque de islas). En este modelo, la población es dividida en varias sub-poblaciones (*demes*) de tal manera que diferentes procesadores manejan el mismo algoritmo en cada sub-población. Finalmente, después de un cierto tiempo (llamado época), los procesadores pasan a una etapa de intercomunicación donde intercambian información y comparten las soluciones encontradas hasta el momento.
3. Variante paralela con enfoque de difusión. Este puede ser visto como un caso extremo del modelo de islas, donde la población de cada isla es un solo individuo y donde hay tantas islas como individuos.

Partiendo de dicha clasificación, se programaron algunas variantes paralelas para el algoritmo PSO que se detallan adelante. El objetivo principal fue programar variantes paralelas para el GPU que emplean código CUDA en vez de los métodos complejos y laboriosos que trabajan, sobre todo, con operaciones paralelas en la memoria gráfica (texturas, etc.). Esto con el fin de evaluar la ganancia en el desempeño que podrían ser alcanzadas por el GPU mediante una estrategia de programación paralela relativamente simple.

Entonces, las variantes paralelas propuestas en este capítulo se diferencian de las que se presentan en [89] porque no incluyen el estilo tradicional de programación paralela para un GPU, es decir, el que emplea operaciones paralelas sobre múltiples datos con el enfoque conocido como SIMD (siglas en inglés para “una instrucción múltiples datos”) [111]. En cambio, las aplicaciones paralelas presentadas aquí trabajan con el nuevo modelo de programación paralela concebido para aprovechar la característica multi-hilos de los GPU de NVIDIA, lo que permite manejar múltiples hilos concurrentes en forma muy eficiente. NVIDIA llama esta característica SIMT (siglas en inglés para “una instrucción múltiples hilos”) [111] y la ofrece como una característica adicional que complementa las operaciones paralelas sobre la memoria. Este nuevo modelo de programación y la herramienta asociada CUDA permiten que los programadores escriban códigos paralelos para los GPU de una forma más natural, con lo que se convierte a los GPU en verdaderos instrumentos de programación paralela para propósito general. De hecho, se prevé que este nuevo modelo de la programación paralela sea la referencia para una nueva especificación de un modelo de programación universal que hará posible la programación paralela, no sólo en los GPU, sino en cualquier arquitectura de múltiples núcleos o plataforma de súper cómputo que aparezca en el futuro.

Aquí se reportan tres variantes paralelas para el algoritmo PSO: dos variantes Globales y una más que se denominó como “incrustada” en vista de que esta última es una singular aplicación del modelo de difusión. Respecto del enfoque global, dos variantes fueron programadas en el GPU:

1. *Global_ev*: Donde únicamente el proceso de evaluación de la función objetivo (función de aptitud) es programado en forma paralela.
2. *Global_ev+up*: Donde todos los cálculos matemáticos, tanto para la función de aptitud como para la velocidad, posición e inercia, son programados en forma paralela.

Acerca de la variante paralela con el modelo de isla, se consideró que el modelo de difusión es un caso extremo del enfoque de isla y, por lo tanto, se decidió incluir ambas variantes en la misma

realización que se denominó como variante incrustada. En la variante incrustada el algoritmo PSO corre casi por completo en el dispositivo GPU, de tal manera que puede ser visto como una caja negra desde el punto de vista de procesador anfitrión. Más aún, mientras que en el modelo de difusión hay un procesador por partícula, en la variante incrustada se emplea un hilo por partícula.

Como código de referencia, se programó al algoritmo PSO secuencial descrito en el Apéndice G, para su ejecución en un solo hilo del GPU, a fin de evaluar en forma justa el desempeño de las variantes paralelas. En todas las variantes paralelas se siguió la estrategia de programación que consiste en crear de un hilo por cada partícula PSO. En esta se sustituye a todos los lazos secuenciales (expresamente aquellos donde las iteraciones están en función del número de partículas) por una sola llamada a un kernel multi-hilos. Con ello, el algoritmo PSO secuencial y sus variantes paralelas tienen esencialmente la misma estructura. En el código secuencial, cada una de las iteraciones de un lazo corresponde a una partícula que, a su vez, es independiente de todas las demás. Tales lazos pueden ser directamente transformados en kernels paralelos, donde las tareas dentro de las iteraciones que corresponden a una partícula se realizan por un hilo independiente [111].

Entonces, mientras que en el código secuencial los movimientos de las partículas son actualizados partícula por partícula, dentro de GPU la actualización de todas las partículas es ejecutada de manera concurrente. La programación paralela con la herramienta CUDA permite al programador arrancar la ejecución de un kernel y crear un bloque de hilos independientes que representan a la población de individuos.

Específicamente, un programa escrito con CUDA incluye los siguientes pasos:

1. Se asigna la memoria del dispositivo GPU.
2. Se copian los datos desde el procesador anfitrión (CPU) a la memoria del GPU.
3. El procesador anfitrión invoca la función de kernel.
4. El GPU ejecuta el código.
5. Se copia la salida resultante, de la memoria GPU a la memoria del procesador anfitrión.

En la Figura 7.3 se muestra la estructura general del algoritmo PSO secuencial. Los siguientes bloques funcionales de una meta-heurística pueden ser observados:

- Inicialización de la población. Se inicializa a las partículas de la población en forma aleatoria.
- Evaluación de la función de aptitud.
- Comparación. Se determina si alguna partícula tiene mejor aptitud que la mejor registrada.
- actualización (imitación). Cada partícula actualiza su posición de acuerdo a las reglas específicas del algoritmo.

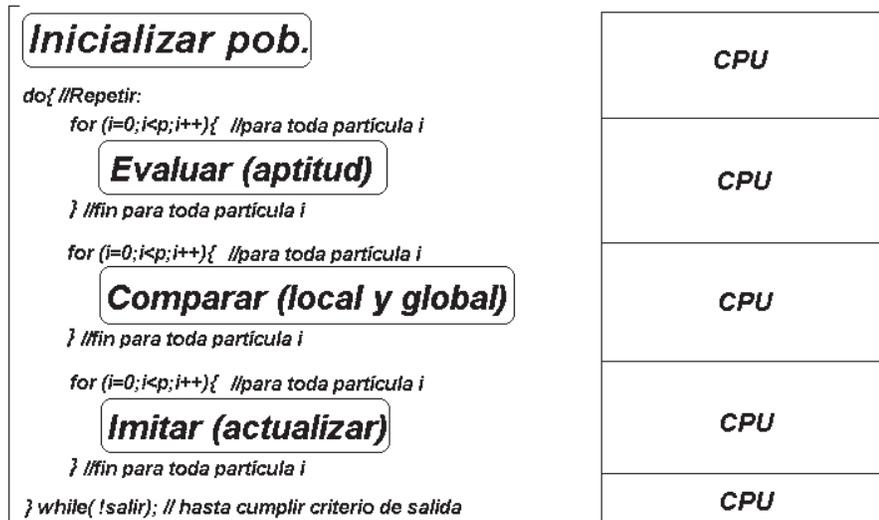


Figura 7.3: Estructura del algoritmo PSO secuencial

Intencionadamente, a fin de ilustrar cómo se realiza la programación paralela en forma directa, el código secuencial ha sido organizado destacando los lazos que están en términos de número de partículas. La idea principal es crear un hilo por cada partícula PSO. Note que en la versión secuencial de PSO (ver Figura 7.3) todos los módulos funcionales son ejecutados para todas las partículas en un solo hilo del GPU.

En la primera de las variantes paralela, Global.ev, sólo el módulo de evaluación de función de aptitud (ver Figura 7.4) se programa en forma paralela. Note que el lazo correspondiente fue sustituido por una llamada al kernel que distribuye el trabajo en múltiples hilos dentro del dispositivo GPU. En la segunda variante paralela, Global.ev+up, todo cálculo aritmético se distribuye en el GPU, sustituyendo al módulo de evaluación de la función de aptitud y al módulo de actualización de la posición por las correspondientes llamadas a un kernel asociado (ver Figura 7.5). Finalmente, en la tercera variante paralela, la denominada como incrustada, únicamente el módulo de inicialización se mantiene en el procesador anfitrión (ver Figura 7.6), mientras que los módulos de evaluación, comparación y de imitación se realizan dentro del GPU, hasta que se alcanza la condición de terminación, mediante las correspondientes llamadas a los kernel asociados.

Acerca del módulo de inicialización, conviene recalcar que la inicialización de las semillas de los individuos (una semilla por hilo), usadas para la generación de números aleatorios, se realiza en el procesador anfitrión y no dentro del GPU. Tal condición garantiza la generación de números aleatorios de buena calidad dentro del GPU y para cada hilo. Debido a esto es que la generación de números aleatorios se realiza de forma diferente para los códigos paralelos y para el secuencial. De hecho, resultó particularmente difícil generar números aleatorios diferentes dentro de GPU a partir de una sola semilla. Entonces, los códigos secuenciales emplearon un método tradicional para generar los números aleatorios (a partir de una sola semilla), mientras los códigos paralelos usaron

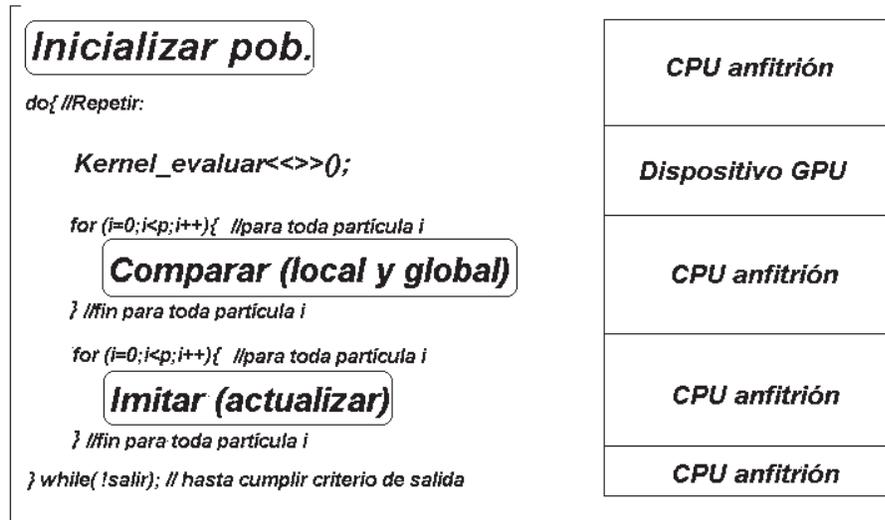


Figura 7.4: Estructura de la variante de Global_ev para el algoritmo PSO paralelo

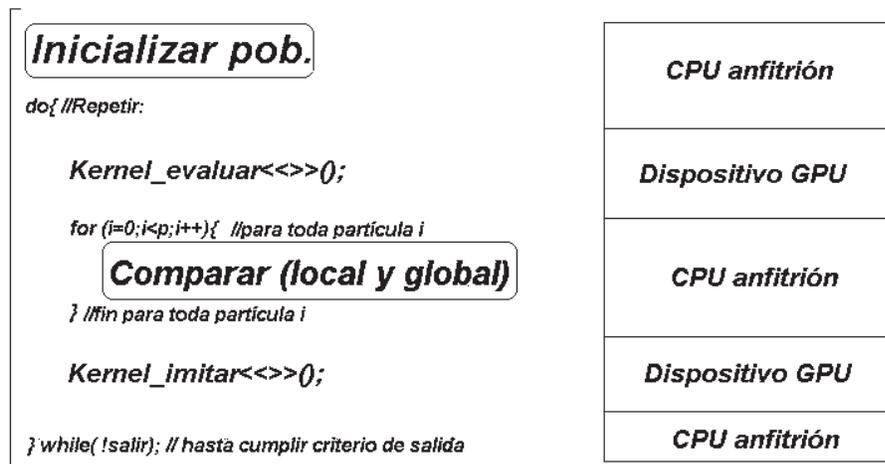


Figura 7.5: Estructura de la variante Global_ev+up para el algoritmo PSO paralelo

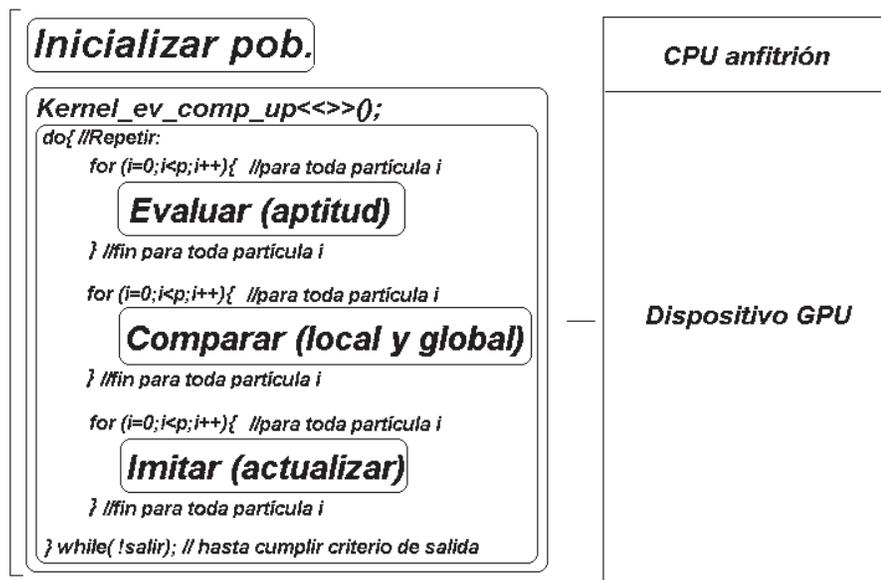


Figura 7.6: Estructura de la variante incrustada para el algoritmo PSO paralelo

tantas semillas como individuos componían a la población, como será detallado más adelante.

Resumiendo, las variantes del algoritmo PSO probadas en este trabajo fueron:

- Variante secuencial. En donde la mayor parte del algoritmo PSO, con excepción del módulo de inicialización de la población y la impresión de los resultados, es ejecutado para todas las partículas en forma secuencial por un solo hilo del GPU.
- Variante Global_ev. En este caso el GPU se encarga únicamente de evaluar en forma paralela la función de aptitud, mientras que el resto del código es ejecutado en el procesador anfitrión.
- Variante Global_ev+up. En este caso el GPU evalúa la función de aptitud y la actualización de la velocidad/posición en forma paralela, mientras que el resto del algoritmo se ejecuta en el anfitrión.
- Variante incrustada. En este caso el GPU ejecuta la mayor parte del algoritmo PSO en forma paralela, con excepción de la inicialización de la población y la impresión de resultados que se delegan al anfitrión.

También se deben tomar en cuenta algunas consideraciones prácticas durante la programación paralela de un algoritmo poblacional para su ejecución en un GPU multi-hilos, a fin de lograr un código funcional:

- **Sobrecarga de comunicación (*overhead*)**. El GPU puede presentar elevada sobrecarga de comunicación debido a la latencia en las transferencias de memoria entre el procesador anfitrión y el dispositivo GPU. Como estas transferencias son relativamente lentas, cualquier

código paralelo para un GPU debe minimizar su empleo. En este sentido, el código paralelo incrustado minimiza el intercambio de información entre el procesador anfitrión y el GPU ya que toda la información originada tanto por el proceso de actualización como por el de evaluación de la aptitud se encuentra disponible dentro del mismo GPU.

- **Sincronización.** Antes de que cualquier decisión sea tomada, por ejemplo durante el proceso de comparación y actualización, todos los hilos ejecutados deben estar sincronizados en algún punto a fin de obtener información confiable. En nuestro caso es particularmente importante una buena sincronización ya que en el código paralelo incrustado los hilos (es decir, los individuos) eventualmente se comunican entre sí, en el entendido de que en una heurística poblacional los individuos comparten información.
- **Competencia.** Este problema puede ocurrir si las variables globales son revisadas y actualizadas simultáneamente por varios hilos. Se deben tomar precauciones apropiadas para manejar este problema. Específicamente, en el algoritmo PSO esta situación puede ocurrir con la variable que contiene el índice al mejor global. En nuestro caso, los hilos se sincronizan justo antes del proceso de comparación y actualización, en seguida se emplea una estrategia de comunicación del tipo maestro-esclavo para garantizar que sólo uno de los hilos (alguno de los individuo que previamente se designa como el maestro) actualiza las variables globales a partir de la información de todo los demás, trabajando a la manera de un moderador, después de solicitar la información a cada individuo y una vez que todos ellos han sido completamente actualizados.
- **Generación de números aleatorios.** A fin de garantizar suficiente diversidad, la generación de números aleatorios de buena calidad es un tema fundamental para la correcta operación del algoritmo PSO paralelo. Este proceso puede volverse problemático si los números aleatorios son generados dentro del GPU sin una estrategia adecuada para la inicialización de las semillas. De hecho, cualquier llamada a la función `rand()` ejecutada en el GPU debe generar números diferentes en cualquier llamada y para cada hilo. Si la condición anterior no se cumple, se podría dificultar la convergencia del algoritmo a causa de una diversidad pobre. En los códigos paralelos propuestos aquí el procesador anfitrión crea e inicializa una semilla por cada individuo (es decir una semilla por hilo). Más tarde, cada hilo por sí mismo genera los números aleatorios que necesita y, durante el proceso, aprovecha para actualizar su semilla, como se puede ver en el código mostrado en la Figura 7.7, donde se siguen las recomendaciones dadas en [80] pero se actualiza a la semilla después de cada llamada a la función `devrand()`. En el código, `dseed[]` es el arreglo que almacena las semillas en la memoria del dispositivo y el índice `p` denota al individuo al que se asocia la semilla. De esta manera podemos generar números aleatorios, tanto fraccionarios como enteros.

7.6. Experimentos y resultados

Los experimentos fueron realizados en una computadora personal con un procesador anfitrión Core Duo de Intel con sistema operativo Linux. El GPU empleado es una tarjeta gráfica GeForce

```

__device__ unsigned int devrand(unsigned int* dseed, int p)
{
    dseed[p] = (3141592621*dseed[p]) + 3141592621;
    return(dseed[p]);
}

__device__ float devfracrand(unsigned int* dseed, int p)
//return a random number between 0 and 1
{
    unsigned int randval=devrand(dseed,p)%100;
    float randreal=(float)(randval);
    return(randreal/100.0f);
}

__device__ int devrandom(unsigned int* dseed, int p,int topint)
// return a random integer between 0 and topint
{
    return(devrand(dseed,p)%topint);
}

```

Figura 7.7: Generación de números aleatorios dentro del GPU

8600GT de NVIDIA, con 256 MBytes de la memoria de trabajo y 4 multiprocesadores, cada uno integrado por 8 núcleos, lo cual representa un total de 32 núcleos de procesamiento. Estos núcleos de procesamiento fueron programados por medio del ambiente CUDA, lo que permitió escribir en forma directa un código paralelo para el GPU NVIDIA, tal y como fue descrito párrafos arriba.

7.6.1. Procedimiento experimental

El objetivo esencial es evaluar el desempeño de las variantes paralelas propuestas para PSO en comparación con el código secuencial. El desempeño se midió en función de la complejidad de la función de aptitud, el número de partículas y el número de iteraciones. Los algoritmos fueron probados para solucionar algunas famosas funciones de referencia (*benchmark*) de optimización global y numérica [100]. Las siguientes funciones fueron seleccionadas porque todas ellas presentaban una complejidad significativa en lo relativo a la evaluación de aptitud [108]:

- F01 - Función de Rosenbrock generalizada.

$$f_1(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

$$-30 < x_i < 30$$

$$\text{mín}(f_1) = f_1(1, \dots, 1) = 0$$

con $n = 30$ dimensiones.

- F02 - Función de Rastrigin generalizada.

$$f_2(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$$

$$-5,12 < x_i < 5,12$$

$$\text{mín}(f_2) = f_2(0, \dots, 0) = 0$$

con $n = 30$ dimensiones.

- F03 - Función de Griewank generalizada.

$$f_3(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$$-600 < x_i < 600$$

$$\text{mín}(f_3) = f_3(0, \dots, 0) = 0$$

con $n = 30$ dimensiones.

Todas estas funciones objetivo son multi-modales, pero F01 y F03 son funciones separables mientras F02 es no separable, en el entendido de que las funciones separables pueden ser escritas como combinaciones lineales de funciones de variables individuales.

Los siguientes experimentos fueron realizados para medir el desempeño de cada una de las variantes de PSO programadas:

- Experimento 1. Mediciones relativas al tiempo de procesamiento consumido por cada uno de los módulos funcionales (es decir el de evaluación, comparación y actualización) dentro del algoritmo PSO secuencial.
- Experimento 2. Medición del tiempo de procesamiento consumido por cada una de las variantes de PSO (la secuencial y todas las paralelas) en términos de complejidad de la función de aptitud.
- Experimento 3. Medición del desempeño en función de número de iteraciones para cada una de las variantes paralelas de PSO.
- Experimento 4. Medición del desempeño en función de número de partículas para cada una de las variantes paralelas de PSO.

El objetivo del Experimento 1 es medir, a manera de referencia, el tiempo de procesamiento consumido por cada uno de los módulos funcionales dentro del algoritmo PSO secuencial. Este experimento fue realizado en dos sesiones. En la primera sesión, los parámetros fueron fijados en 30 dimensiones, 20 vecinos, y 2000 iteraciones, mientras que se varió el número de partículas. En la segunda sesión, los parámetros fueron fijados en 256 partículas, 20 vecinos, y 2000 iteraciones, mientras que se varió el número de dimensiones. La variante PSO secuencial fue ejecutada 10 veces para cada una de las 3 funciones de referencia probadas, midiendo el tiempo promedio consumido, en segundos, para cada uno de los módulos funcionales, así como la proporción, en porcentaje, respecto al total del tiempo consumido. Los resultados de este experimento permitieron determinar en forma empírica cuáles eran los módulos del algoritmo PSO que era más conveniente programar

en forma paralela.

El objetivo del Experimento 2 es medir la dependencia del tiempo de procesamiento consumido por cada una de las variantes PSO (secuencial y paralelas) en términos de complejidad de la función de aptitud. Los parámetros fueron fijados a 10,000 iteraciones, 256 partículas, y 30 dimensiones. Las variantes de PSO fueron ejecutadas 10 veces (para todas las funciones de referencia probadas: F01, F02 y F03) a fin de obtener la solución promedio encontrada, así como el tiempo promedio consumido, en segundos.

Los experimentos 3 y 4 se realizaron para probar la dependencia del desempeño de las variantes PSO en términos de dos parámetros fundamentales: el número de iteraciones y el número de partículas, respectivamente. Se reportan el error promedio de las soluciones encontradas y el tiempo promedio consumido, en segundos. En el experimento 3 se varió el número de iteraciones (10,000, 30,000, y 60,000 iteraciones), pero se fijaron los siguientes parámetros:

- Partículas = 256
- Dimensiones = 60
- Función = F03
- Criterio de parada = número de iteraciones.

Finalmente, en el experimento 4, se varió el número de partículas (64, 256, y 1024 partículas), pero se fijaron los siguientes parámetros:

- Iteraciones = 10,000
- Dimensiones = 60
- Función = F03
- Criterio de parada = número de iteraciones.

Después de observar que la optimización de F03 era la función de referencia que más tiempo de procesamiento tomaba a la variante secuencial de PSO, se decidió reportar únicamente los resultados referentes a la optimización F03 a fin de ilustrar el comportamiento general observado durante la optimización de las tres funciones de referencia probadas.

7.6.2. Métricas de desempeño para procesamiento paralelo

Para evaluar el desempeño de los códigos paralelos, tradicionalmente se definen las siguientes métricas:

- Costo computacional y
- aceleramiento.

El costo computacional C se define como el tiempo de procesamiento (en segundos) que un algoritmo dado consume. Entonces, el rendimiento computacional T se define como el inverso del costo computacional:

$$T = \frac{1}{C}$$

El aceleramiento S mide la mejora alcanzada en el tiempo consumido de ejecución y expresa cuán rápido es un código paralelo comparado con un código de referencia:

$$S = \frac{T_{targ}}{T_{ref}}$$

donde T_{targ} es el rendimiento del código paralelo bajo estudio y T_{ref} es, en nuestro caso, el rendimiento del código secuencial.

Finalmente, definimos la eficiencia paralela E como la razón que resulta de dividir el aceleramiento entre el número de núcleos de procesamiento:

$$E = \frac{S}{n}$$

donde n es el número de núcleos de procesamiento en el GPU (32 núcleos en nuestro caso).

7.6.3. Resultados experimentales

Experimento 1

Donde se obtienen las mediciones relativas al tiempo de procesamiento consumido por cada uno de los módulos funcionales (es decir, evaluación, comparación y actualización) dentro de la variante secuencial de PSO.

Los resultados experimentales mostraron que, después de optimizar la tres funciones n -dimensionales de referencia (F01, F02 y F03); la evaluación de la función de aptitud consumió una proporción relativamente baja del tiempo de procesamiento total, en comparación con los cálculos empleados durante la actualización de la velocidad y de la posición (ver Cuadros 7.1 y 7.2). Esto se debe a que las operaciones de actualización, aunque con menor complejidad aritmética, son calculadas más veces que la evaluación de función de aptitud. Además, el proceso que genera los números aleatorios puede consumir una cantidad significativa del tiempo de procesamiento cuando es el mismo procesador el que ejecuta el algoritmo PSO y también el que genera los números aleatorios.

Así, los resultados experimentales permiten afirmar que, para los tres problemas de optimización n -dimensionales representados por F01, F02 y F03, la mayor parte del tiempo de procesamiento del algoritmo PSO es consumido por las tareas de actualización (posición y velocidad) y no por la tarea de evaluación de la función de aptitud.

Cuadro 7.1: Distribución del costo computacional en función del número de partículas

Proporción del tiempo de ejecución total con 30 dimensiones, 2000 iteraciones						
Partículas						
64		256		1024		
Función	Costo de evaluación	Costo de comparación y actualización	Costo de evaluación	Costo de comparación y actualización	Costo de evaluación	Costo de comparación y actualización
F01	2.84 %	96.59 %	3.03 %	97.25 %	2.03 %	97.28 %
F02	27.54 %	70.34 %	26.93 %	69.38 %	27.16 %	69.43 %
F03	39.86 %	55.24 %	41.31 %	55.81 %	14.88 %	35.93 %

Cuadro 7.2: Distribución del costo computacional en función del número de dimensiones

Proporción del tiempo de ejecución total con 256 partículas, 2000 iteraciones						
Dimensiones						
30		60		120		
Función	Costo de evaluación	Costo de comparación y actualización	Costo de evaluación	Costo de comparación y actualización	Costo de evaluación	Costo de comparación y actualización
F01	3.03 %	97.25 %	3.26 %	96.74 %	3.38 %	96.66 %
F02	26.93 %	69.38 %	30.51 %	67.81 %	33.56 %	66.53 %
F03	41.31 %	55.81 %	44.92 %	53.55 %	46.95 %	49.49 %

Experimento 2

Donde se obtienen las mediciones de la dependencia del tiempo de procesamiento consumido por cada una de las variantes PSO, en términos de complejidad de la función de aptitud.

Respecto del tiempo consumido, se observó que el desempeño de la variante de Global_ev puede hacerse peor que el de la variante secuencial si la complejidad aritmética de la función objetivo es pequeña (ver Figura 7.8). Este comportamiento asombroso (es decir, la mejora en el desempeño de variante de Global_ev conforme aumenta la complejidad de la función objetivo) se explica si pensamos que una gran intensidad aritmética (es decir un gran número de operaciones aritméticas por cada operación de memoria realizada) permiten que el administrador de hilos del GPU traslape transacciones de memoria (ver en [111] el capítulo 5). Por lo tanto, los resultados experimentales muestran un mejor desempeño para la variante de Global_ev, en comparación con la variante secuencial, para el caso de la optimización de las funciones F02 y F03. Sorprendentemente, la mejora es más notable para la variante Global_ev+up cuando realiza la optimización de la función más compleja (F03) y es seguida por las optimizaciones de las funciones menos complejas (F01 y F02). El mejor desempeño se logró por la variante incrustada que realizó la optimización de F03, con un tiempo de ejecución promedio de 6 segundos, seguido de la optimización de F01 y F02. De nuevo, hay que notar que el mejor desempeño alcanzado se logra cuando se optimiza la más compleja de las funciones objetivo. En comparación con la variante secuencial, que consume 35.9 segundos, la mejora en el desempeño de la variante incrustada es notable, pues esta última sólo consume 6.6 segundos.

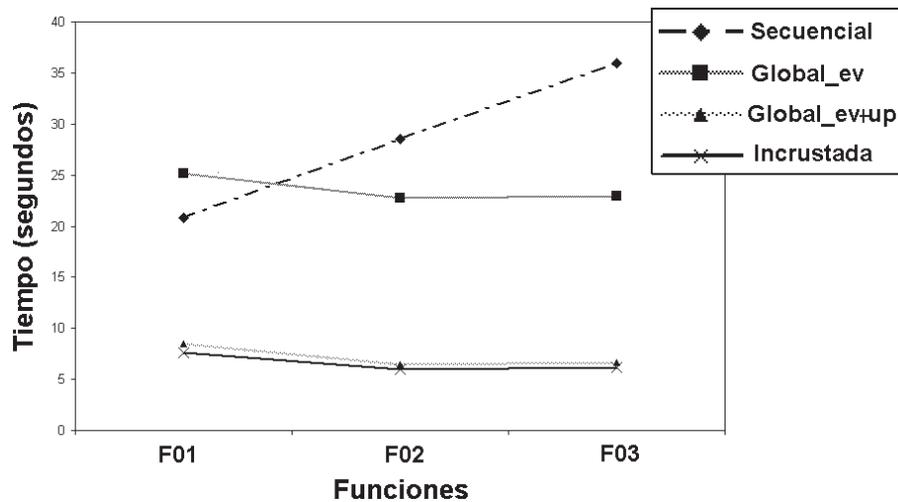


Figura 7.8: Desempeño de las variantes de PSO en términos de la complejidad de la función de aptitud. En este caso, complejidad (F01) < complejidad (F02) < complejidad (F03)

Entonces, los resultados experimentales permiten afirmar que el desempeño para las variantes paralelas Global_ev y Global_ev+up es menor que para la variante incrustada, lo que es debido, fundamentalmente, a la sobrecarga de comunicación que resulta de los intercambios de información entre el computadora anfitrión y el dispositivo GPU.

Experimentos 3 y 4

Donde se obtienen las mediciones de la dependencia del desempeño para cada una de las variantes paralelas de PSO, en términos de número de partículas e iteraciones.

En lo relativo al error, observamos en todas las variantes que al aumentar las iteraciones es posible obtener una solución muy cercana al óptimo global. Por ejemplo, al fijar 10,000 iteraciones como un punto de partida, se observó que la variante secuencial tiene un error bajo y muy uniforme. Es notable cómo la variante incrustada comienza con un error ligeramente mayor que las otras variantes, pero cuando aumenta el número de iteraciones, la solución obtenida logra la misma precisión para 30,000 y 60,000 iteraciones (ver Figura 7.9) que, por otra parte, se ejecuta en tan solo una fracción del tiempo consumido por la variante secuencial, a pesar de realizar el mismo número de iteraciones. De hecho, con 60,000 iteraciones la variante incrustada tiene un error inferior, en una relación de 5 a 1, que la secuencial con las mismas iteraciones (ver Figura 7.9).

En la Figura 7.10 se grafica el tiempo consumido cuando se aumenta el número de iteraciones. Note que el desempeño de la variante Global_ev+up es mejor que el de la variante de Global_ev; a pesar de que en la primera variante el GPU realiza la actualización de la velocidad y de la posición además de la evaluación de función de aptitud. En cuanto a la variante incrustada, es muy interesante notar que el tiempo consumido es prácticamente el mismo, independientemente del número de iteraciones.

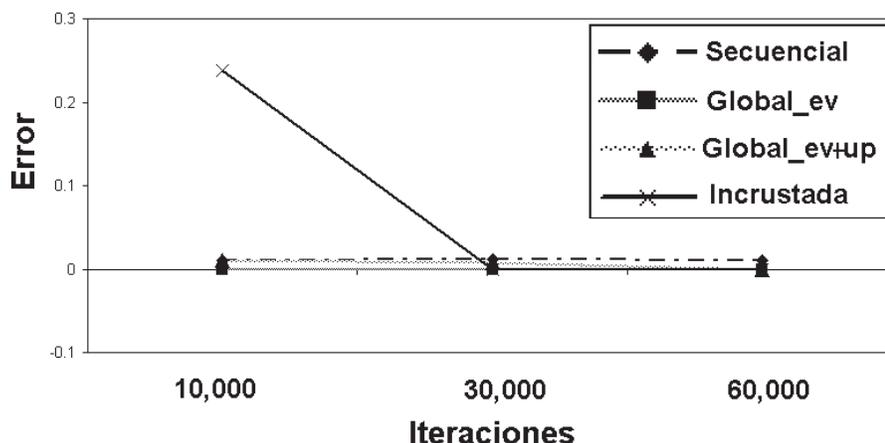


Figura 7.9: Error de variantes PSO, ejecutando la optimización F03, con diferente número de iteraciones

Los resultados experimentales muestran un comportamiento similar para el caso donde se varia el número de partículas (ver Figura 7.11). Se observa que el desempeño de la variante incrustada es mejor cuando el número de partículas se incrementa. En general, los resultados experimentales muestran que el desempeño mejora cuando el número de operaciones matemáticas distribuidas al GPU también aumenta.

Discusión sobre las métricas de desempeño

En general, el comportamiento observado es consistente para todas las funciones de referencia probadas. Los resultados experimentales permiten afirmar que el desempeño de las variantes paralelas mejora notablemente conforme más cálculos son distribuidos al dispositivo GPU. La métrica de desempeño para todas las variantes paralelas de PSO es presentada en las Figuras 7.12 a 7.15.

Es claro que el desempeño de las variantes paralelas es mejor respecto a la variante secuencial. Por ejemplo, la optimización de la función de Griewank generalizada (F03, con elevado costo computacional, para 60 dimensiones, 60,000 iteraciones y 256 partículas) le toma 214.13 segundos a la variante secuencial, 136.3 segundos a la variante de Global_ev, y 39.89 segundos a la variante Global_ev+up, pero la variante incrustada fue ejecutada en tan solo 7.66 segundos, lo que representa un aceleramiento de 27.97, comparado con la variante secuencial, y una eficiencia de prácticamente 1 (recuerde que en nuestro caso el GPU se integra por 32 núcleos).

Note que al aumentar el número de partículas e iteraciones, la eficiencia para las variantes Global_ev y Global_ev+up se mantiene prácticamente constante, aunque se incrementa el aceleramiento. Esto se debe a que en el caso de las variantes Global_ev y Global_ev+up la sobrecarga de comunicación aumenta proporcionalmente al número de partículas y al número iteraciones. En con-

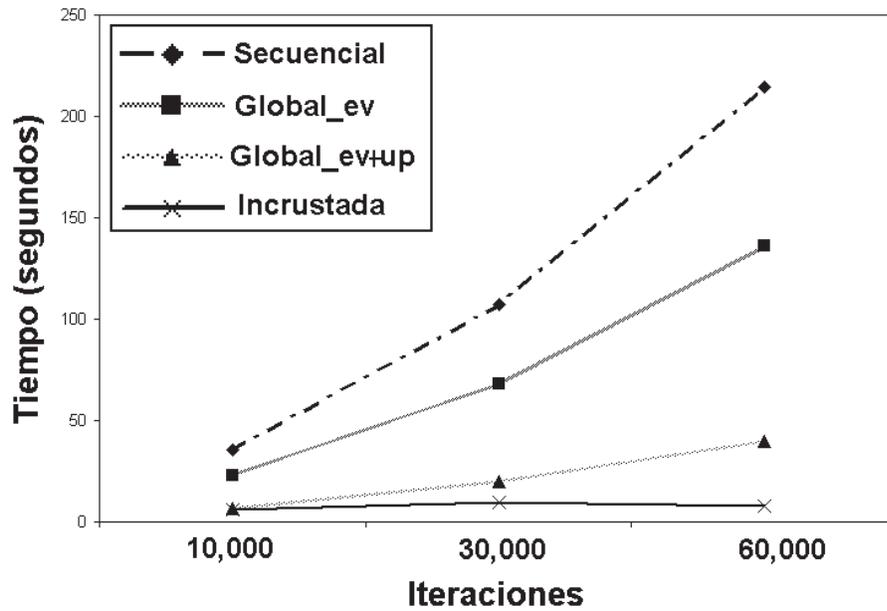


Figura 7.10: Tiempo de procesamiento consumido por las variantes de PSO, durante la optimización de F03, como función del número de iteraciones

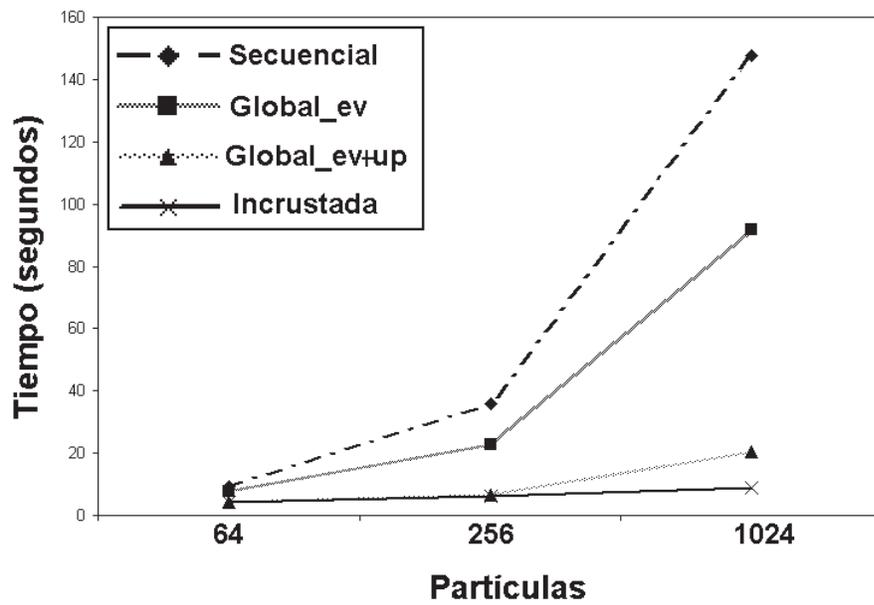


Figura 7.11: Tiempo de procesamiento consumido por las variantes de PSO, durante la optimización de F03, como función del número de partículas

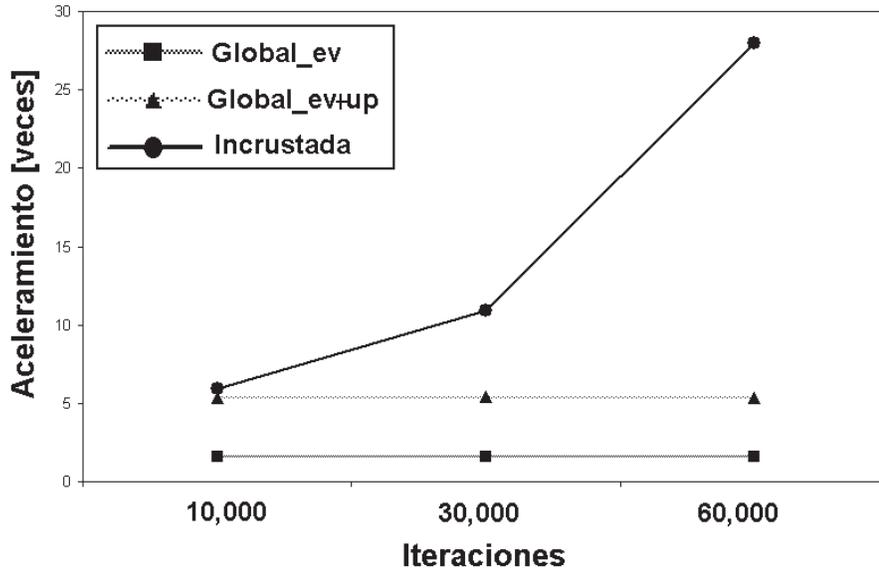


Figura 7.12: Aceleramiento para las variantes paralelas de PSO como función del número de iteraciones

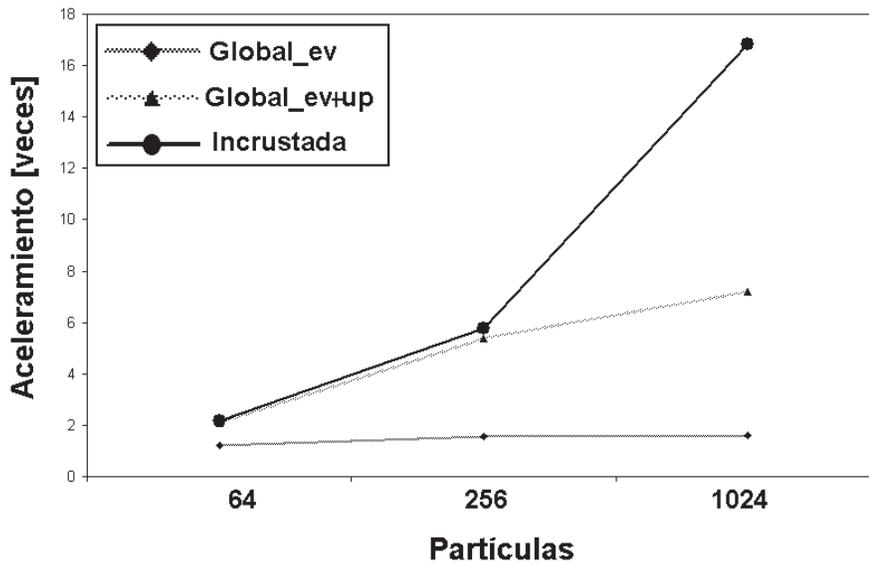


Figura 7.13: Aceleramiento para las variantes paralelas de PSO como función del número de partículas

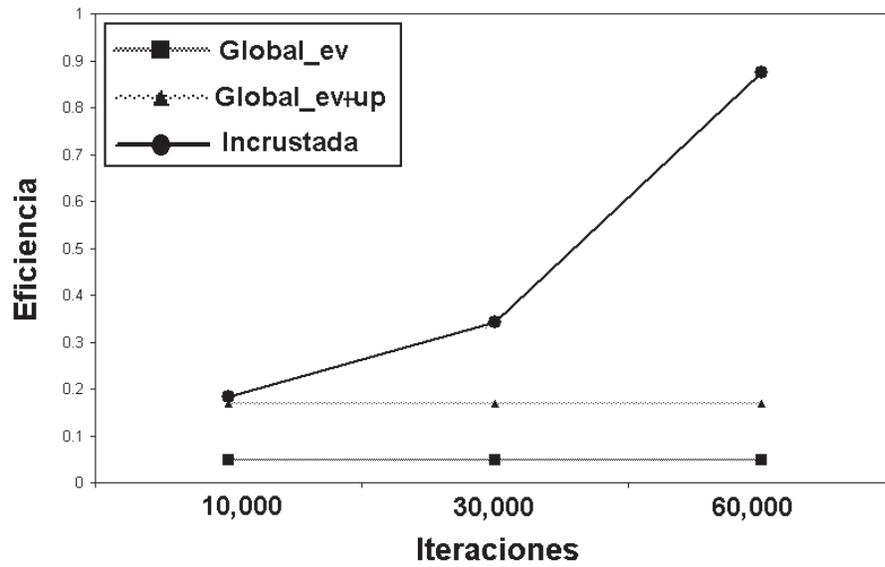


Figura 7.14: Eficiencia para las variantes paralelas de PSO como función del número de iteraciones

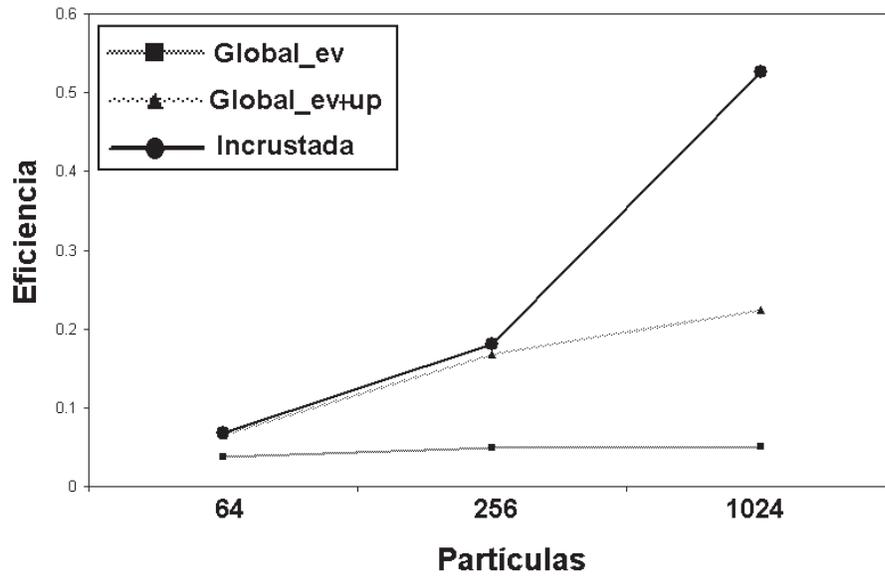


Figura 7.15: Eficiencia para las variantes paralelas de PSO como función de número de partículas

traste, la variante incrustada aumenta su eficiencia ya sea que aumenta el número de las partículas o el número de las iteraciones.

7.7. Conclusiones del capítulo

En este capítulo se presentó un estudio empírico relativo al desempeño de tres variantes paralelas para el algoritmo de optimización por cúmulo de partículas (PSO) que fueron puestas en práctica en un GPU multi-hilos, usando a CUDA como el más reciente de los modelos de programación paralela. Los resultados experimentales mostraron que con estas herramientas computacionales es posible obtener códigos paralelos, para un algoritmo poblacional dado, con una mejora significativa en el desempeño y mediante una programación paralela simple y directa.

En general, se demuestra que un código paralelo que únicamente aprovecha la característica multi-hilos del GPU (es decir, exclusivamente la capacidad de distribuir el trabajo en hilos y sin usar operaciones paralelas en múltiples datos), por lo menos produce un aceleramiento proporcional al número de núcleos del GPU, comparado con el algoritmo secuencial, con lo que se obtiene el poder de cómputo de un *cluster* en una computadora personal convencional. Se espera que al emplear la totalidad de las capacidades del GPU (es decir, empleando operaciones paralelas sobre múltiples datos además de la capacidad de administrar múltiples hilos), sea posible aumentar aún más el desempeño. También, se demuestra que el desempeño total de GPU mejora cuando aumenta la cantidad de tareas simples distribuidas en los hilos del GPU. Este comportamiento se observa en el desempeño de las tres variantes paralelas propuestas, pero es más evidente en la variante incrustada, donde el desempeño alcanzado es considerablemente mejor.

Finalmente, se prevé que el desarrollo de los GPU y su nuevo modelo de programación paralela, conlleve a que, en el futuro cercano, se disponga de características de super-cómputo científico en las computadoras personales convencionales y en los dispositivos de procesamiento especializados tales como los procesadores digitales de señales (DSP, por sus siglas en inglés), gracias a las nuevas plataformas de hardware inspiradas en los GPU modernos y sus precursores: los procesadores vectoriales.

Capítulo 8

Conclusiones generales y trabajo futuro

En este trabajo se han realizado algunas aportaciones teórico-prácticas con el fin de aumentar el aprovechamiento de la línea de distribución de energía eléctrica doméstica como medio de propagación (canal PLC) mediante el empleo de técnicas avanzadas de procesamiento de señales (en particular la aplicación de la descomposición de ondeletas discretas), la aplicación de heurísticas bio-inspiradas, así como el empleo de nuevas arquitecturas de cómputo masivo. En particular, los métodos propuestos para la estimación del canal son de gran utilidad en aplicaciones a entornos contaminados por ruido impulsivo de gran densidad. Las aportaciones se pueden resumir como sigue:

- Entorno de simulación para el canal PLC y el transreceptor OFDM. Esta herramienta queda a disposición de los interesados para que puedan usarla en sus investigaciones ya que no existe una herramienta similar disponible con código abierto.
- Modelo del canal PLC y estimación del canal con el enfoque de la descomposición de ondeletas, en un entorno afectado por una mezcla de ruido Gaussiano y ruido impulsivo de gran densidad, que es el ambiente de operación para los canales PLC que se encuentra frecuentemente en la práctica. En la literatura, no existe publicado ningún trabajo donde se aplique la descomposición de ondeletas para enfrentar el ruido impulsivo para la estimación del canal PLC.
- Algoritmo de optimización por cúmulo de partículas (PSO) para la estimación semi-ciega y seguimiento de canales PLC. La aplicación de heurísticas bio-inspiradas para la estimación del canal es un campo poco explorado y por lo mismo existen pocas publicaciones al respecto.
- realización paralela de la heurística bio-inspirada PSO en un procesador de gráficos (GPU) y evaluación de su desempeño. Debido a que este es un campo de muy reciente aparición, los resultados producidos por esta vertiente de la investigación han generado gran interés y han sido bien acogidos para su publicación.

El código del programa con el modelo del canal PLC (canal estacionario, ciclo-estacionario y ruido de clase A de Middleton) así como el entorno de simulación Monte Carlo para un transreceptor OFDM/DMT se publica en una versión básica a fin de que sirva como plataforma para evaluar el

rendimiento de los métodos de estimación de canal propuestos en este trabajo, así como las futuras propuestas de otros investigadores.

Las simulaciones realizadas para la estimación del canal soportada por ondeletas, propuesta en esta tesis para su uso en canales PLC plagados de ruido impulsivo, especialmente con una tasa de 6000 impulsos por ciclo de potencia, arrojan resultados prometedores. En particular, estos resultados permiten concluir que la técnica de estimación de canal, con base en la eliminación de ruido, mediante la descomposición de ondeletas a partir de la respuesta al impulso y en conjunto con la regla de umbralado de Donoho y Johnstone (encogimiento) es la mejor alternativa, en términos de la curva de probabilidad de error en los bits (BER), para enfrentar una mezcla de ruido gaussiano e impulsivo, de acuerdo con el modelo de clase A de Middleton, especialmente cuando la SNR es menor a 5dB. Se probaron las técnicas propuestas tanto en un canal estacionario como en uno ciclo-estacionario y se pudo verificar la consistencia de los resultados en ambos casos.

En las simulaciones realizadas para el canal ciclo-estacionario, donde siempre se presentó cierta probabilidad de error en los bits (BER) irreducible, también muestran la importancia de contar con una técnica efectiva para el seguimiento del canal. En este contexto, se propuso una técnica semi-ciega para la estimación y seguimiento del canal PLC, inspirada en la heurística bio-inspirada de optimización por cúmulo de partículas (PSO) y específicamente para enfrentar el problema del seguimiento del canal en un entorno contaminado por ruido impulsivo de alta densidad. Se aprovechó la característica auto-organizativa y de rastreo del cúmulo de partículas PSO para seguir a las clases en la constelación de los símbolos recibidos, que se encuentran en constante movimiento debido a las variaciones ciclo-estacionarias del canal PLC. Las simulaciones arrojaron resultados prometedores en términos de la curva de probabilidad de error en los bits (BER), aunque resultaron costosas en cuanto a memoria y tiempo consumido por la computadora.

Se incursionó en la nueva arquitectura de cómputo denominada como unidad de procesamiento de gráficos (GPU) a fin de explorar su potencial para ejecutar en forma paralela el algoritmo poblacional bio-inspirado que se empleó (PSO) para la propuesta de estimación y seguimiento semi-ciego del canal PLC. Una motivación importante para incursionar en estas nuevas tecnologías es la previsión de que estas se encuentren disponibles, en un futuro no muy lejano, a nivel de los dispositivos de procesamiento digital de señales (DSP) y teniendo en cuenta que ello hará perfectamente realizable la aplicación de las heurísticas bio-inspiradas al campo de las comunicaciones digitales y, en particular, de la tecnología PLC. Esta vertiente de la investigación, a pesar de encontrarse aún en su etapa inicial, arrojó resultados de gran interés que tuvieron buena acogida para su publicación. Los resultados de la evaluación del desempeño del algoritmo PSO programado en forma paralela, con CUDA y ejecutados en el GPU multi-hilos, mostraron que se produce un aceleramiento significativo en comparación con la versión secuencial original.

8.0.1. Trabajo futuro

Como trabajo futuro se resaltan los siguientes tópicos:

- Comparar el desempeño de la estimación soportada por ondeletas respecto a otras técnicas de eliminación del ruido impulsivo.
- Mejorar la técnica de seguimiento del canal ciclo-estacionario para alcanzar los niveles de probabilidad de error, del transreceptor simulado, a niveles más competitivos (del orden de 10^{-3}) según lo requieren las aplicaciones PLC comerciales.
- Aplicar la descomposición de ondeleta para mejorar el seguimiento del canal mediante una mejor predicción con base en un modelo AR.
- Realizar una biblioteca propia de rutinas de descomposición de ondeleta en paralelo, mediante el aprovechamiento de las capacidades de paralelismo que ofrece la arquitectura del GPU.
- Migrar el código de simulación del canal PLC y del transreceptor OFDM a la plataforma GPU multi-hilos.

Parte III

Apéndices

Apéndice A

Notación empleada

Cuadro A.1: Notación empleada

Notación	significado
$L^2(\mathbb{R})$:	Espacio de funciones de cuadrados integrables.
ℓ^2 :	Espacio de secuencias de cuadrados sumables.
$\langle \cdot, \cdot \rangle$:	Producto escalar.
$\Im\{z\}$:	Parte imaginaria de un complejo z .
$\Re\{z\}$:	Parte real de un complejo z .
$(\cdot)^*$:	Conjugado complejo.
\mathbb{Z} :	Espacio de enteros.
\mathbb{N} :	Espacio de los naturales.
\mathbb{R} :	Espacio de los reales.
$\delta_{m,n}$:	Delta de Kronecker.
δ :	Delta de Dirac.

Apéndice B

Glosario

AR. *Autoregressive.* Modelo Auto-regresivo.

AWCN. *Additive White Class A Noise.* Ruido Aditivo Blanco de Clase A, también conocido como modelo de ruido de clase A de Middleton.

BER. *Bit Error Rate.* Proporción de errores en bits, que indica la probabilidad de error en los bits recibidos después del proceso de detección.

BPL. *Broadband Power-line Communication.* Comunicación de banda ancha por la línea eléctrica.

BPSK. *Binary Phase-Shift Keying.* Modulación digital mediante desplazamientos binarios de fase.

CAD. Conversor Analógico a Digital.

Célula. Es la subred formada por un módem central MC o repetidor R y los módems secundarios M abonados al mismo. Una célula M es la que tiene por cabeza a un módem central, mientras que una célula R es aquella que tiene por cabeza a un módem repetidor.

CDA. Conversor Digital a Analógico.

CDMA. *Code-Division multiple Access.* Técnica de acceso al canal de comunicación para múltiples usuarios mediante repartición de códigos.

CIR. *Channel impulse response.* Respuesta al impulso del canal.

CFR. *Channel frequency response.* Respuesta en frecuencia del canal.

COFDM. *Coded OFDM.* OFDM codificado.

CP. *Cyclic prefix.* Prefijo cíclico empleado como intervalo de guardia.

DAB. *Digital Audio Broadcast.* Transmisiones de amplia difusión con audio digital.

DAPSK. *Differential Amplitude and phase shift keying.* Codificación por diferencias en amplitud y fase.

DE. *Differential Evolution.* Técnica bio-inspira de optimización con evolución diferencial.

Despliegue de fase. Traducción para *unwrapping*.

Dispersión. En este contexto es sinónimo de extensión o prolongación.

DFT. *Discrete Fourier Transform.* Transformada Discreta de Fourier.

DMT. *Discrete Multitone.* Multi-tono discreto. Técnica MCM empleada típicamente en sistemas DSL.

DPSK. *Differential phase shift keying.* Codificación por diferencias de fase.

DSL. *Digital Subscriber Line.* Línea de abonado digital, generalmente se trata de una línea de cobre de par trenzado.

DT-CWT. *Dual-tree complex wavelet transform.* Transformada de ondeleta compleja de árbol dual.

DWT. *Discrete Wavelet Transform.* Transformada discreta de ondeleta.

EM. *Expectation-maximization.* Maximización de la esperanza.

Enlace ascendente. Es el flujo de datos que va de un módem secundario M hacia un módem central MC o repetidor R.

Enlace descendente. Es el flujo de datos que va de un módem central MC o repetidor R hacia un módem secundario.

Error Residual. Es el componente del error que no se puede explicar una vez calculados los coeficientes de regresión. Frecuentemente debido a errores de medición u omisión.

Escalonamiento. Operación que consiste en multiplicar una señal de datos por un código o secuencia de escalonamiento cuyos elementos tienen un ancho menor al de los símbolos del mensaje. Esta operación produce una dispersión de los datos en el dominio dual.

Espectro. Traducción para *spectrum*, generalmente empleado como sinónimo para el dominio de la frecuencia, una banda en frecuencia o el perfil de la respuesta en frecuencia de un sistema.

Espectro Disperso. Traducción para la técnica de *Spread Spectrum*.

FDMA. *Frequency-Division multiple Access.* Técnica de acceso al canal de comunicación para múltiples usuarios mediante repartición del espectro de frecuencia.

FEC. *Forward Error Correction.* Corrección de errores en sitio empleada en sistemas de comunicación unidireccionales o con niveles de ruido que hacen impráctica la retransmisión de los mensajes.

FFT. *Fast Fourier Transform.* Transformada rápida de Fourier directa.

FIR. *Finite Impulse Response.* Filtros de respuesta al impulso finita.

Ficha. Traducción para *chip*, que es el nombre que recibe el pulso rectangular que integra la secuencia empleada para multiplicar la señal portadoras de datos en los sistemas **espectro disperso** (*Spread Spectrum*).

Filtro acoplado. Traducción para *matched filter* o filtro de correlación cruzada.

GPU. *Graphics Process Unit.* Unidad de procesamiento de gráficos.

HDD. *Hard Decision decoder.* Decodificador de decisión dura. Tipo de algoritmo de decodificación para corrección de errores. En la variante dura (*hard*) los datos considerados sólo pueden tener valores fijos, por ejemplo 0 y 1, y emplean una distancia de Hamming para detectar errores.

ICI. *Inter-carrier Interferente.* Interferencia entre portadoras.

IDWT. *Inverse Discrete Wavelet Transform.* Transformada inversa de ondeleta discreta.

IFFT. *Inverse Fast Fourier Transform.* Transformada inversa rápida de Fourier.

Igualamiento. Traducción para *equalization*.

i.i.d.. *Independent and identically distributed.* Independiente e idénticamente distribuido.

IIR. *Infinite Impulse Response.* Filtros de respuesta al impulso infinita.

ISI. *Inter-symbol Interferente.* Interferencia entre símbolos.

LS. *Least Squares.* Criterio de mínimos cuadrados.

LLS. *Linear Least Squares.* Recta que mejor aproxima a una serie en el sentido de mínimos cuadrados.

LMMSE. *Linear Minimum Mean Square Error.* Mínimo Error Cuadrático Medio Lineal.

MAP. *Maximum a posteriori.* Máximo *a posteriori*.

MC-CDMA. *Multicarrier CDMA.* Es la combinación de la técnica MCM con la técnica CDMA.

MCM. *Multi Carrier Modulation.* Modulación de múltiples portadoras.

MC-SS. *Multicarrier Spread Spectrum.* Es la combinación de la técnica OFDM con la técnica de dispersión de espectro.

MC-SS-MA. *Multicarrier Spread Spectrum Multiple Access.* Es la técnica multi-bloque que asigna un bloque MC-SS a cada subportadora.

MEM. *Maximum-entropy method.* Método de máxima entropía.

MIMO. *Multiple Inputs Multiple Outputs.* Múltiples entradas y múltiples salidas.

ML. *Maximum Likelihood.* Máxima verosimilitud.

MMSE. *Minimum Mean Square Error.* Mínimo Error Cuadrático Medio.

MSE. *Mean Square Error.* Error Cuadrático Medio.

Módem central, MC. Es el módem que se encuentra a la cabeza de una red (por ejemplo, ubicado en el transformador de distribución de mediana a baja tensión). Hay que notar que el módem repetidor es un módem central respecto de los módems abonados que dependen del mismo.

Módem repetidor, R. Es el módem que sirve como nodo intermediario entre un módem central y módems abonados distantes.

Módem secundario, M. Es el módem abonado o usuario que depende de un módem central. Un módem repetidor es un módem secundario con respecto al módem central del que depende.

ODWT. *Orthogonal Discrete Wavelet Transform.* Transformada de ondeleta ortogonal discreta.

OFDM. *Orthogonal Frequency-Division Multiplexing.* Multiplexado (asignación) por repartición de frecuencias ortogonales. Técnica MCM empleada típicamente en sistemas inalámbricos.

ppm. Partes por millón, que equivale a multiplicar por 10^{-6} .

pdf. *Probability density function.* Función de densidad de probabilidad.

PN. *Pseudo-Noise.* Pseudo-ruido.

PLC. *Power-line Communication.* Comunicación por la línea eléctrica.

PSAM. *Pilot symbol assisted modulation.* Modulación asistida por símbolos pilotos.

PSO. *Particle Swarm Optimization.* Técnica bio-inspirada de optimización por cúmulo de partículas.

QAM. *Quadrature amplitude modulation.* Modulación digital mediante la amplitud de dos portadoras en cuadratura (desfasadas 90°).

QPSK. *Quadrature Phase-Shift Keying.* Modulación digital mediante desplazamientos de fase en cuadratura.

QoS. *Quality of Service.* Calidad del servicio.

Repliegue de fase. Traducción para *wrapping effect*.

SDD. *Soft Decision decoder.* Decodificador de decisión suave. Tipo de algoritmo de decodificación para corrección de errores. En la variante suave (*soft*) los datos considerados pueden tener cualquier valor entre 0 y 1, representando en forma implícita la probabilidad de que sean 0 o 1 mediante una distancia euclidiana.

SINR. *Signal to Interference and Noise Ratio.* Razón de potencias de la señal útil entre la potencia de la señal de interferencia más ruido.

SNR. *Signal to Noise Ratio.* Relación entre la potencia promedio de la señal útil y la potencia promedio del ruido.

SVD. *Singular value decomposition.* Descomposición en valores simples (o singulares).

TDMA. *Time-Division multiple Access.* Técnica de acceso al canal de comunicación para múltiples usuarios mediante repartición del tiempo.

UWB. *Ultra Wideband.* Banda ancha extrema.

VCO. *Voltage Controlled Oscillator.* Oscilador controlado por voltaje.

WOFDM. *Wavelet OFDM.* OFDM de ondeletas.

ZP. *Zero-padding.* Relleno con ceros.

Apéndice C

Efecto del número de niveles de descomposición en la estimación soportada por ondeletas

En las Figuras C.1 a C.5 se puede observar el desempeño de la estimación soportada por ondeletas para diferentes niveles de descomposición. En la rutina de descomposición de ondeletas empleada (`FWT_PO()`), disponible en la biblioteca `Wavelab`) el nivel de descomposición se especifica en forma indirecta indicando el valor \log_2 que corresponde a el número de coeficientes con que se requiere a la secuencia de aproximación.

Se puede observar que el desempeño es prácticamente el mismo cuando la descomposición entrega una aproximación con 1 y con 4 coeficientes, pero sufre un empobrecimiento apreciable a partir de 16 coeficientes de aproximación. La disminución del desempeño es poca para 16 coeficientes, pero es claro que con un mayor número de coeficientes de aproximación (y, por lo tanto, menor número de niveles de descomposición) el efecto es significativo. Por lo tanto, se recomienda emplear el máximo nivel de descomposición (aproximación con un coeficiente), pero se puede interrumpir a un costo aceptable al alcanzar una aproximación de 16 coeficientes.

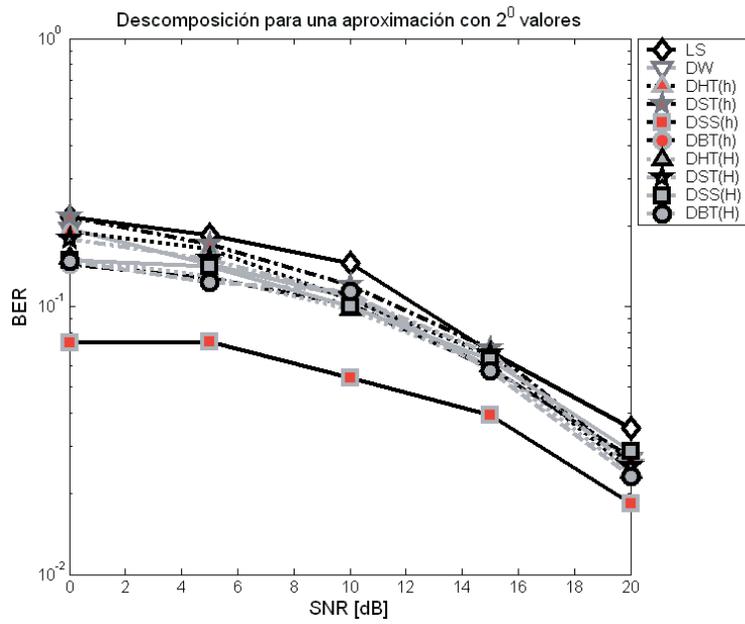


Figura C.1: Desempeño con descomposición hasta obtener aproximación con 1 coeficiente

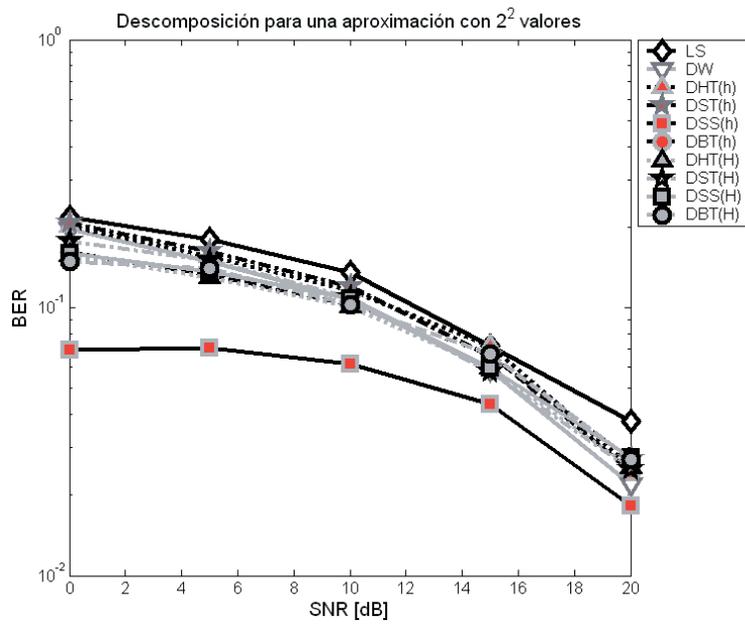


Figura C.2: Desempeño con descomposición hasta obtener aproximación con 4 coeficientes

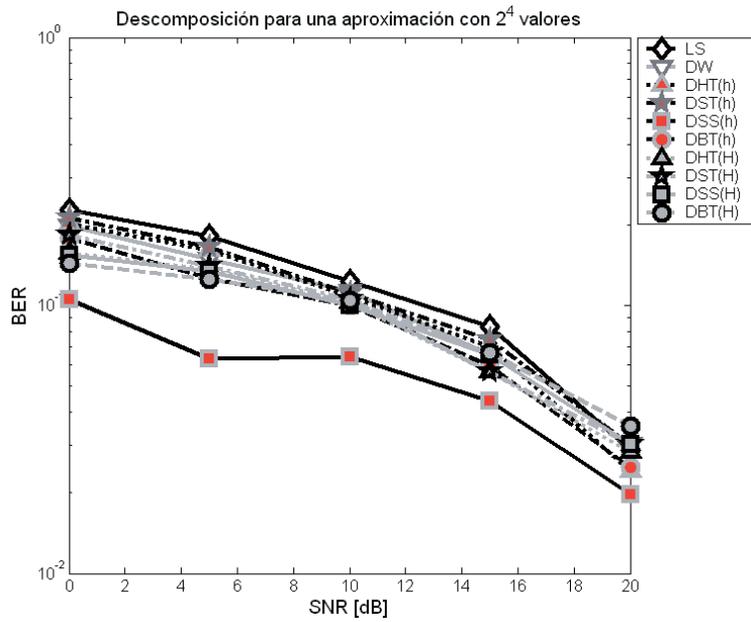


Figura C.3: Desempeño con descomposición hasta obtener aproximación con 16 coeficientes

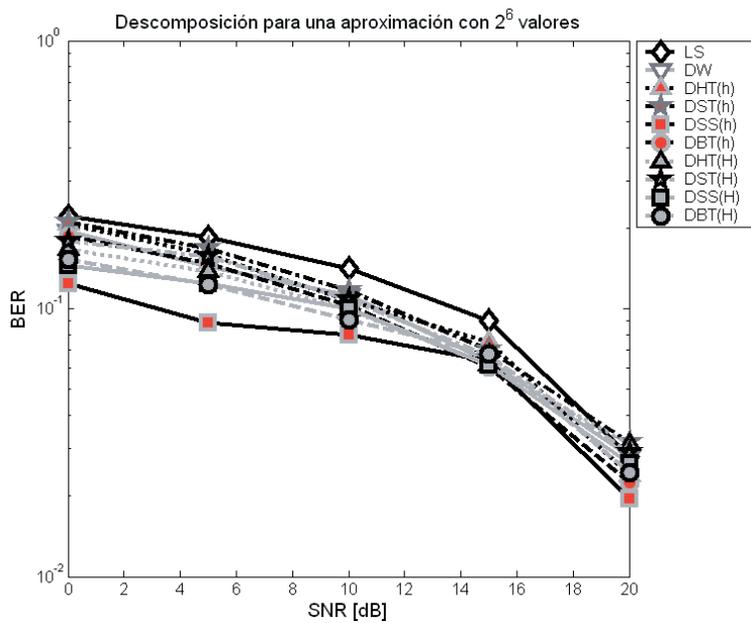


Figura C.4: Desempeño con descomposición hasta obtener aproximación con 64 coeficientes

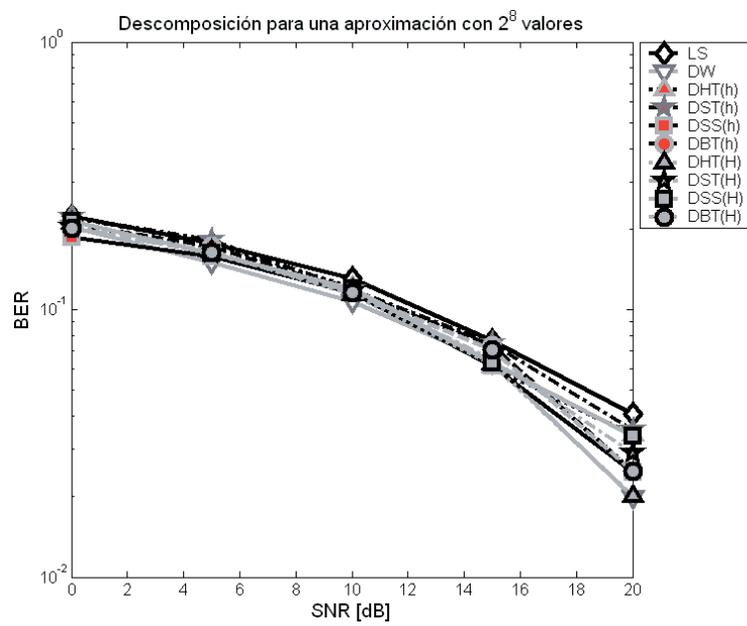


Figura C.5: Desempeño con descomposición hasta obtener aproximación con 256 coeficientes

Apéndice D

Comparación de la estimación soportada por ondeletas contra técnicas clásicas de estimación lineal

En las Figuras D.1 a D.3 se puede observar el desempeño de la estimación soportada por ondeleta contra las más importantes técnicas de estimación lineal clásica, adicionales a la de estimación en el sentido de mínimos cuadrados (LS) y todas asistidas por símbolos pilotos. En primer lugar tenemos la denominada como método de minimización del error lineal cuadrático promedio (*Linear Minimum Mean Square Error Estimator*, LMMSE) y, luego, a sus variantes simplificada y SVD, conforme lo descrito en la sección 3.4 y las recomendaciones prácticas dadas en [129] (ver código en Sección E.3.7).

En estas figuras se observa que, en las condiciones probadas, la estimación soportada por ondeleta (en su variante DSS(h)) tiene el mejor desempeño respecto a la mayoría de las técnicas LMMSE, con excepción de la variante SVD cuando el ruido tiene una alta densidad (6000 impulsos por ciclo de potencia). Sin embargo, se observa que la variante de estimación SVD presenta una probabilidad de error irreducible que es invariante a la densidad del ruido impulsivo. Este comportamiento en la variante SVD es un problema ya que cuando el ruido es prácticamente gaussiano también presenta la probabilidad de error irreducible, pese a que es el más noble de los casos. Esta característica en la variante puede ser explicada por la reducción de rango realizada con base en el máximo tiempo de dispersión considerado.

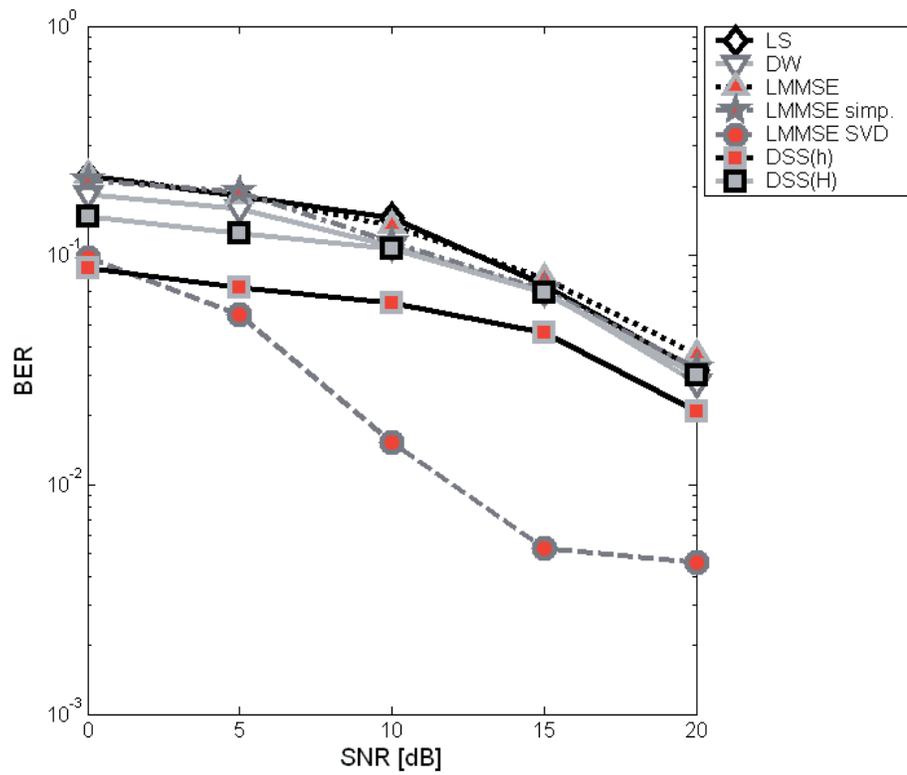


Figura D.1: Escenario estacionario contaminado con ruido Gaussiano más ruido impulsivo con una densidad de 6000 pulsos por ciclo

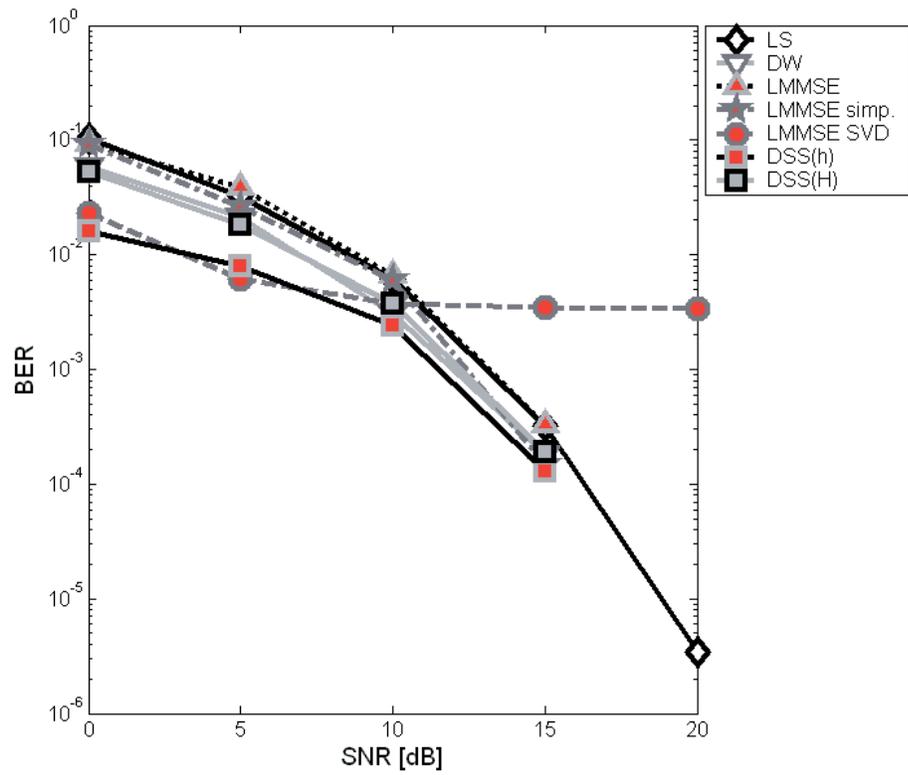


Figura D.2: Escenario estacionario contaminado con ruido Gaussiano más ruido impulsivo con una densidad de 600 pulsos por ciclo

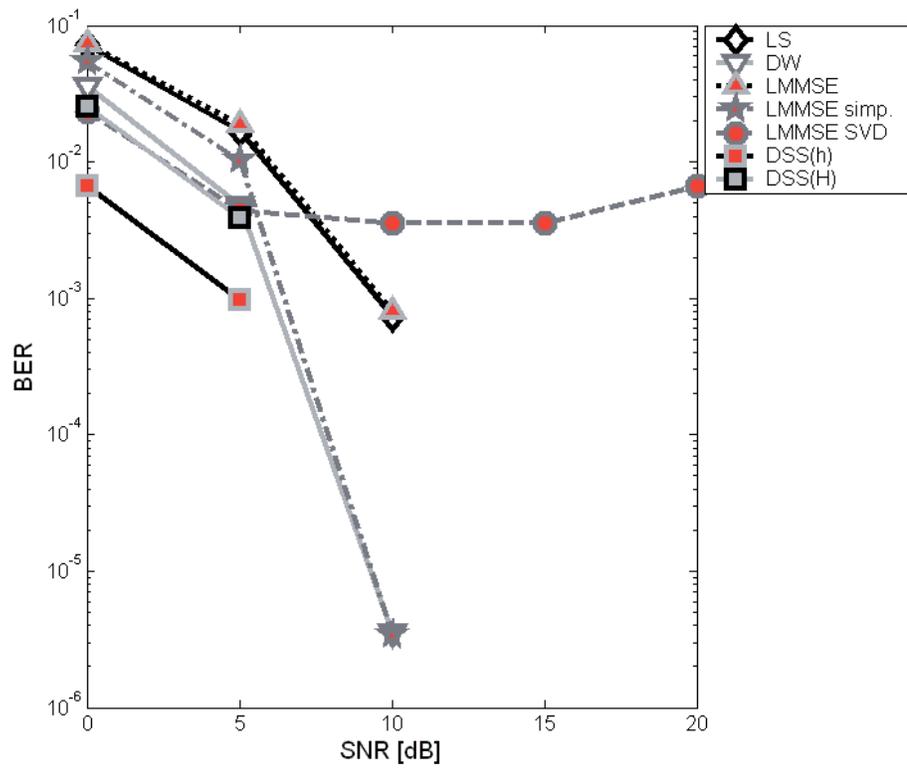


Figura D.3: Escenario estacionario contaminado con ruido prácticamente Gaussiano (ruido impulsivo con una densidad de 10 pulsos por ciclo)

Apéndice E

Código Matlab básico empleado en las simulaciones numéricas para la estimación del canal PLC

A fin de realizar las simulaciones reportadas en esta tesis, se realizó un programa en Matlab que comprende tanto la simulación del canal PLC como del transreceptor OFDM empleado. Aquí se presenta la descripción general de dicho programa de simulación, el listado de las funciones esenciales que lo componen y las instrucciones necesarias para instalarlo y para ejecutarlo.

E.1. Estructura del programa

El programa tiene una estructura jerárquica donde en el más alto nivel se encuentra la rutina que invoca el usuario mediante la línea de mandos de Matlab. Podemos identificar cuatro niveles dentro del código del programa:

- Nivel 1. Inicialización de parámetros y lazo principal de la simulación Monte Carlo. Este es el nivel más alto, en el que se definen las variables globales y se pasan como parámetros de entrada para las funciones invocadas a partir de este punto. En este nivel las salidas de las funciones permiten graficar las curvas de probabilidad de error en los bits (BER) vs. la relación señal a ruido (SNR).
- Nivel 2. Simulación de enlace OFDM/DMT. En este nivel se realiza la simulación, trama por trama, del intercambio unidireccional entre un transmisor y un receptor OFDM en su variante DMT. Se construye el número especificado de tramas OFDM y se genera la señal por transmitir. Las señales transmitidas son sometidas a las transformaciones que resultan de viajar por el canal PLC. Entonces, la señal recibida presenta las distorsiones infringidas por el canal más las perturbaciones adicionales por el ruido. Las señales OFDM son demoduladas, igualadas y detectadas para reconstruir las tramas OFDM y contabilizar el número de errores de todas las tramas.

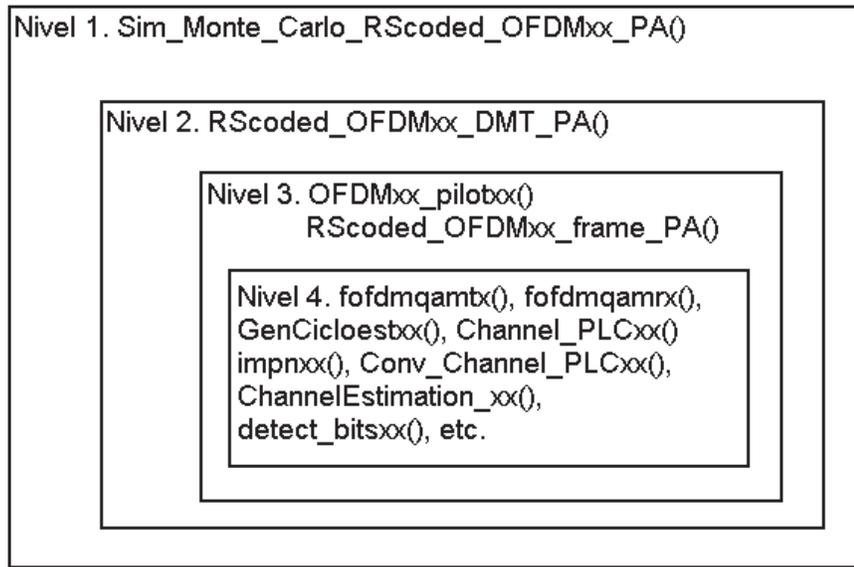


Figura E.1: Estructura del programa de simulación

- Nivel 3. Simulación del enlace de una trama, ya sea de reconocimiento (un solo símbolo piloto) o con datos (símbolo piloto más 255 símbolos OFDM con datos). En este nivel, por una parte, se genera una trama, la señal OFDM y el efecto de su viaje por el canal PLC. Por otro lado, se recibe la señal de la trama distorsionada y contaminada por ruido, se demodula, se iguala y de detecta. A este nivel se realiza el conteo de los errores detectados en la trama recibida.
- Nivel 4. Funciones específicas. Este es el nivel más bajo, donde se encuentran las funciones especializadas para generar los datos fuente, codificación y decodificación, realizar el mapeo QAM, construir la señal analítica, la modulación y demodulación OFDM, simulación del modelo de canal y del modelo de ruido, la estimación del canal, así como el igualamiento y detección de los símbolos de modulación recibidos.

En la Figura E.1 se puede apreciar la estructura jerárquica del programa mientras que en la Figura E.2 se muestra el esquema donde se aprecia el flujo de los datos para el enlace OFDM simulado, en conjunto con el modelo del canal PLC. El flujo de datos mostrado en el esquema de la Figura E.2 es el mismo que en el programa simulador a nivel de cada trama (Nivel 3). Finalmente, en el Cuadro E.1 se proporcionan los nemotécnicos genéricos para las funciones asociadas.

E.2. Instalación y ejecución

Antes de ejecutar el simulador es necesario asegurarse que el sistema cuenta con todo lo necesario. Primero es necesario tener instalado Matlab en su versión 6.5, aunque en principio puede ejecutarse, sin mayor problema, en versiones posteriores. Entonces, debe crearse una carpeta de

Cuadro E.1: Lista de funciones básicas

Tarea	Función asociada	Nivel
Simulación Monte Carlo e impresión de resultados	Sim_Monte_Carlo_RScoded_OFDMxx_PA()	1
Lazo de repeticiones para múltiples tramas	RScoded_OFDMxx_DMT_PA()	2
Trama OFDM de reconocimiento	OFDMxx_pilotxx()	3
Trama OFDM de datos	RScoded_OFDMxx_frame_PA()	3
Inicialización de parámetros OFDM	finiofdmqamx()	4
Generación de bits de datos	bit_simbolAx() bit_source_xx()	4 4
Mapeo QAM	DMT_map_bits()	4
Modulación OFDM	fofdmqamtx()	4
Adición de prefijo cíclico	faddgi()	4
Simulación del canal	GenCicloestxx()	4
	ImpCicloestxx()	4
	HT()	4
	Conv_Channel_PLCxx()	4
	Channel_PLCxx()	4
Simulación de ruido impulsivo	impnxx()	4
	Poisson_interarrivals()	4
	Poisson_N_arrivalsx()	4
Demodulación OFDM	fofdmqamrx()	4
Estimación del canal	ChannelEstimation_xx()	4
	LMMSE_CE_xx()	4
	HardThresholding()	4
	SoftThresholding()	4
	SoftThresholding_D_J()	4
Detección	detect_bitsxx()	4
Des-mapeo QAM	DMT_unmap_bits()	4
Conteo de errores	ErroresRxTxx()	4
Misceláneas	antipodal_2_onoff()	4
	CalculaPotencia()	4
	getmin()	4
	Get_masked_geq()	4
	maped_bit_count()	4
	miatan2()	4
	migaussiana02()	4
	onoff_2_antipodal()	4

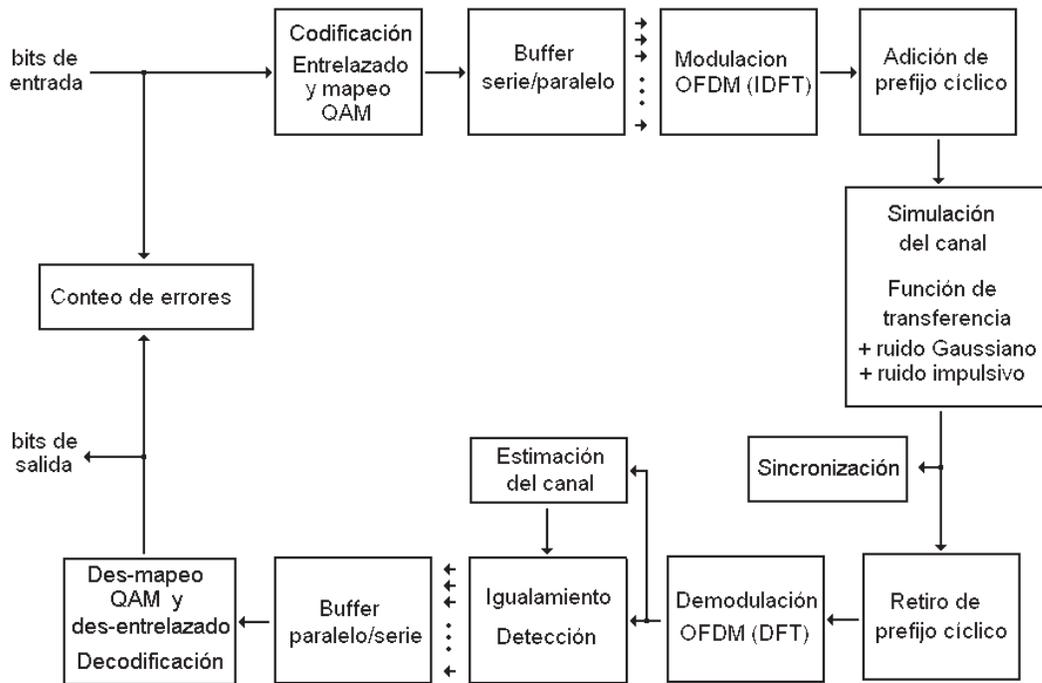


Figura E.2: Flujo de datos en el enlace OFDM simulado

trabajo en una ruta apropiada. Por ejemplo, si Matlab se encuentra instalado para el sistema operativo Windows, debe existir una ruta similar a `C:\MATLAB6p5\`. En este caso, una ruta de trabajo podría ser `C:\MATLAB6p5\work\Sim_OFDM_PLC\`. En esta carpeta de trabajo se deben copiar todas las rutinas listadas en el Cuadro E.1.

También es necesario contar con la biblioteca Wavelab para Matlab [50]. Para instalar esta biblioteca simplemente se descompacta el archivo que la contiene en la carpeta en la carpeta de Matlab denominada `toolbox`. Una vez descompactada la biblioteca debe aparecer la carpeta `Wavelab850` dentro de la carpeta `toolbox`. Por ejemplo, en nuestro caso la ruta para la biblioteca Wavelab debería ser `C:\MATLAB6p5\toolbox\Wavelab850`.

Una vez que se cuenta con todo lo antes expuesto, simplemente se ejecuta la interfaz de usuario de Matlab y en su línea de mandos se introducen las rutas con los archivos de nuestro interés, por ejemplo:

```
>> addpath C:\MATLAB6p5\work\Sim_OFDM_PLC\
>> addpath C:\MATLAB6p5\toolbox\Wavelab850\
```

A continuación, en la línea de mandos de Matlab, se ejecuta una rutina *script* que forma parte del código para la biblioteca Wavelab y que termina de definir las rutas de trabajo para la misma:

```
>> Wavepath
```

Finalmente, para ejecutar el programa de simulación simplemente invoque la rutina principal `Sim_Monte_Carlo_RScoded_OFDMxx_PA()` en la línea de mandos de la siguiente manera:

```
>> Sim_Monte_Carlo_RScoded_OFDMxx_PA
```

Espere a que el programa termine, lo cual puede tomar desde un par de horas hasta algunas decenas de horas, dependiendo de la computadora que lo ejecute y de los parámetros de la simulación. El programa termina cuando aparece en la pantalla la grafica BER vs. SNR correspondiente. Para un usuario principiante, se sugiere modificar únicamente los parámetros de operación que aparecen en el cuerpo de la rutina principal `Sim_Monte_Carlo_RScoded_OFDMxx_PA()`, por ejemplo, la densidad del ruido impulsivo, y el rango y densidad de los valores SNR a probar. Otros parámetros de este programa pueden ser modificados, pero se requiere tener un conocimiento más profundo de todo el código antes de realizar cualquier cambio.

E.3. Listado del código esencial

En esta sección se presenta el listado de las funciones más relevantes dentro de la estructura del programa. Se ofrecen como referencia y pueden sufrir ligeros cambios respecto a la última versión liberada para su distribución.

E.3.1. Cuerpo del programa de simulación Monte Carlo

```

%*****
%*****
%Sim_Monte_Carlo_RScoded_OFDM_PA
%*****
%*****
%Script para simular probabilidad de error contra SNR
% para sistema OFDM-DMT codificado (Reed-Solomon) con canal PLC
% selectivo en frecuencia, con ruido impulsivo,
% impedancia ciclo-estacionaria y prueba de
% diversas técnicas de estimación de canal.
%
%*****
%*****
%Nota: Esta versión emplea la técnica pilot-aided y se ajusta para deshabilitar
%el comportamiento ciclo-estacionario.
% - Se hace data_frames=4 para ser congruentes con un canal estacionario.
%*****
%*****
%Programador: Gerardo Laguna
%Fecha:8/6/10
%*****
data_frames=4; % numero mínimo de tramas de datos a transmitir en cada simulación
QAMsimbolos=512; % número de subportadoras=simbolos QAM por símbolo OFDM
QAMsize=2;% bits por símbolo de modulación QAM.
frec_s=30000000;%frecuencia de muestreo de la señal analítica
DMTlevel=-35; %magnitud, en dBs, de la respuesta en frecuencia a partir del cual
    %transmitir datos
SNRindB=[0:5:20];
Imp_c=600; %numero promedio de impulsos de ruido por ciclo de señal de potencia
%Imp_c=6000; %numero promedio de impulsos de ruido por ciclo de señal de potencia
    %(sensible a nivel de símbolo)

```

```

GIR=0.00001; % Tasa de la potencia del ruido gaussiano contra la del ruido impulsivo.
%*****
%Par'ámetros de impedancia ciclo-estacionaria ZD:
%*****
Fp=60;%frecuencia de la l'inea en Hz
Tp=1/Fp;%periodo de la l'inea en segs
%banda 'util: qamsimbolos= subportadoras para el rango (0,fs]
fz=[(2*pi)/QAMsimbolos:(2*pi)/QAMsimbolos:2*pi];%banda de prueba para la
% impedancia, normalizada a 2pi
total_ticks=100;%marcas de tiempo para actualizar impedancia ciclo-estacionaria
dtz=(Tp/total_ticks)*1000;% delta de t para tiempo de prueba, en milisegundos
tz=[dtz:dtz:Tp*1000]; %tiempo de prueba para la impedancia con 100 muestras de
%un ciclo de potencia, en milisegundos
%Se genera matriz de impedancias que modela el comportamiento
%ciclo-estacionario para el rango de prueba:
ZD=GenCicloest04(fz,tz,250,3,1,500,Tp*1000,0.05,-0.5,-0.6,0.15,
0.6,-0.15,0.5,0,0.75,1,2,4);
%*****

ploting=0; %Las simulaciones en su mayor parte corren sin graficar.
%cicloestacionario=0; %Deshabilita comportamiento ciclo-estacionario.
%cicloestacionario=1; %habilita comportamiento parcialmente ciclo-estacionario.
%cicloestacionario=2; %habilita comportamiento totalmente ciclo-estacionario.
%noisetype=0;% sin ruido
%noisetype=1;% con ruido gaussiano
noisetype=2;% con ruido gaussiano m'as impulsivo

%wavfam='Beylkin';
%wavfam='Coiflet';
wavfam='Daubechies';
%wavfam='Symmlet';
%wavfam='Vaidyanathan';
%wavfam='Battle';
%wavpar=12; %Daubechies: vanishing moments=wavpar/2;
wavpar=8;
%Nivel real de descomposici'on: L=segmentos=log2(n)-wavlev=log2(qamsimbolos)-wavlev
% => Longitud de los coeficientes de aproximaci'on: length(aprox)=length(orig)/2^L:
wavlev=4;
%*****
cicloestacionario=0; %Deshabilita comportamiento ciclo-estacionario.
%*****
L=length(SNRindB);
exp=10;%experimentos

%*****
disp(fix(clock));
%*****
thresh=0;%LS
aver_err_prb=zeros(1,L);
for p=1:exp
    smld_err_prb=zeros(1,L);
    for i=1:L
        [smld_err_prb(1,i),data_bits,Z,H,He]=
RScoded_OFDM_DMT_PA(data_frames,QAMsimbolos,QAMsize,frec_s,DMTlevel,SNRindB(i),
Imp_c,GIR,ploting,cicloestacionario,noisetype,thresh,wavfam,wavpar,wavlev,ZD,
total_ticks,Fp);
    end
    aver_err_prb=aver_err_prb+smld_err_prb;
end
smld_err_prb0=aver_err_prb/exp;
disp('1 de 14');
%*****
thresh=1;%DW
aver_err_prb=zeros(1,L);
for p=1:exp

```

```

smdl_err_prb=zeros(1,L);
for i=1:L
    [smdl_err_prb(1,i),data_bits,Z,H,He]=RScoded_OFDM_DMT_PA(data_frames,
        QAMsimbolos,QAMsize,frec_s,DMTlevel,SNRindB(i),Imp_c,GIR,ploting,
        cicloestacionario,noisetype,thresh,wavfam,wavpar,wavlev,ZD,total_ticks,Fp);
end
aver_err_prb=aver_err_prb+smdl_err_prb;
end
smdl_err_prb1=aver_err_prb/exp;
disp('2 de 14');
%*****
thresh=2;
aver_err_prb=zeros(1,L);
for p=1:exp
    smld_err_prb=zeros(1,L);
    for i=1:L
        [smdl_err_prb(1,i),data_bits,Z,H,He]=RScoded_OFDM_DMT_PA(data_frames,
            QAMsimbolos,QAMsize,frec_s,DMTlevel,SNRindB(i),Imp_c,GIR,ploting,
            cicloestacionario,noisetype,thresh,wavfam,wavpar,wavlev,ZD,total_ticks,Fp);
        end
    aver_err_prb=aver_err_prb+smdl_err_prb;
end
smdl_err_prb2=aver_err_prb/exp;
disp('3 de 14');
%*****
thresh=3;
aver_err_prb=zeros(1,L);
for p=1:exp
    smld_err_prb=zeros(1,L);
    for i=1:L
        [smdl_err_prb(1,i),data_bits,Z,H,He]=RScoded_OFDM_DMT_PA(data_frames,
            QAMsimbolos,QAMsize,frec_s,DMTlevel,SNRindB(i),Imp_c,GIR,ploting,
            cicloestacionario,noisetype,thresh,wavfam,wavpar,wavlev,ZD,total_ticks,Fp);
        end
    aver_err_prb=aver_err_prb+smdl_err_prb;
end
smdl_err_prb3=aver_err_prb/exp;
disp('4 de 14');
%*****
thresh=4;%t:DHT
aver_err_prb=zeros(1,L);
for p=1:exp
    smld_err_prb=zeros(1,L);
    for i=1:L
        [smdl_err_prb(1,i),data_bits,Z,H,He]=RScoded_OFDM_DMT_PA(data_frames,
            QAMsimbolos,QAMsize,frec_s,DMTlevel,SNRindB(i),Imp_c,GIR,ploting,
            cicloestacionario,noisetype,thresh,wavfam,wavpar,wavlev,ZD,total_ticks,Fp);
        end
    aver_err_prb=aver_err_prb+smdl_err_prb;
end
smdl_err_prb4=aver_err_prb/exp;
disp('5 de 14');
%*****
thresh=5;%t:DST
aver_err_prb=zeros(1,L);
for p=1:exp
    smld_err_prb=zeros(1,L);
    for i=1:L
        [smdl_err_prb(1,i),data_bits,Z,H,He]=RScoded_OFDM_DMT_PA(data_frames,
            QAMsimbolos,QAMsize,frec_s,DMTlevel,SNRindB(i),Imp_c,GIR,ploting,
            cicloestacionario,noisetype,thresh,wavfam,wavpar,wavlev,ZD,total_ticks,Fp);
        end
    aver_err_prb=aver_err_prb+smdl_err_prb;
end
smdl_err_prb5=aver_err_prb/exp;

```

```

disp('6 de 14');
%*****
thresh=6;%t:DSS
aver_err_prb=zeros(1,L);
for p=1:exp
    smld_err_prb=zeros(1,L);
    for i=1:L
        [smld_err_prb(1,i),data_bits,Z,H,He]=RScoded_OFDM_DMT_PA(data_frames,
            QAMsimbolos,QAMsize,frec_s,DMTlevel,SNRindB(i),Imp_c,GIR,ploting,
            cicloestacionario,noisetype,thresh,wavfam,wavpar,wavlev,ZD,total_ticks,Fp);
    end
    aver_err_prb=aver_err_prb+smld_err_prb;
end
smld_err_prb6=aver_err_prb/exp;
disp('7 de 14');
%*****
thresh=7;%t:DBT
aver_err_prb=zeros(1,L);
for p=1:exp
    smld_err_prb=zeros(1,L);
    for i=1:L
        [smld_err_prb(1,i),data_bits,Z,H,He]=RScoded_OFDM_DMT_PA(data_frames,
            QAMsimbolos,QAMsize,frec_s,DMTlevel,SNRindB(i),Imp_c,GIR,ploting,
            cicloestacionario,noisetype,thresh,wavfam,wavpar,wavlev,ZD,total_ticks,Fp);
    end
    aver_err_prb=aver_err_prb+smld_err_prb;
end
smld_err_prb7=aver_err_prb/exp;
disp('8 de 14');
%*****
thresh=8;
aver_err_prb=zeros(1,L);
for p=1:exp
    smld_err_prb=zeros(1,L);
    for i=1:L
        [smld_err_prb(1,i),data_bits,Z,H,He]=RScoded_OFDM_DMT_PA(data_frames,
            QAMsimbolos,QAMsize,frec_s,DMTlevel,SNRindB(i),Imp_c,GIR,ploting,
            cicloestacionario,noisetype,thresh,wavfam,wavpar,wavlev,ZD,total_ticks,Fp);
    end
    aver_err_prb=aver_err_prb+smld_err_prb;
end
smld_err_prb8=aver_err_prb/exp;
disp('9 de 14');
%*****
thresh=9;%f:DHT
aver_err_prb=zeros(1,L);
for p=1:exp
    smld_err_prb=zeros(1,L);
    for i=1:L
        [smld_err_prb(1,i),data_bits,Z,H,He]=RScoded_OFDM_DMT_PA(data_frames,
            QAMsimbolos,QAMsize,frec_s,DMTlevel,SNRindB(i),Imp_c,GIR,ploting,
            cicloestacionario,noisetype,thresh,wavfam,wavpar,wavlev,ZD,total_ticks,Fp);
    end
    aver_err_prb=aver_err_prb+smld_err_prb;
end
smld_err_prb9=aver_err_prb/exp;
disp('10 de 14');
%*****
thresh=10;%f:DST
aver_err_prb=zeros(1,L);
for p=1:exp
    smld_err_prb=zeros(1,L);
    for i=1:L
        [smld_err_prb(1,i),data_bits,Z,H,He]=RScoded_OFDM_DMT_PA(data_frames,
            QAMsimbolos,QAMsize,frec_s,DMTlevel,SNRindB(i),Imp_c,GIR,ploting,

```

```

        cicloestacionario,noisetype,thresh,wavfam,wavpar,wavlev,ZD,total_ticks,Fp);
    end
    aver_err_prb=aver_err_prb+smdl_err_prb;
end
smdl_err_prbA=aver_err_prb/exp;
disp('11 de 14');
%*****
thresh=11;%f:DSS
aver_err_prb=zeros(1,L);
for p=1:exp
    smld_err_prb=zeros(1,L);
    for i=1:L
        [smdl_err_prb(1,i),data_bits,Z,H,He]=RScoded_OFDM_DMT_PA(data_frames,
            QAMsimbolos,QAMsize,frec_s,DMTlevel,SNRindB(i),Imp_c,GIR,ploting,
            cicloestacionario,noisetype,thresh,wavfam,wavpar,wavlev,ZD,total_ticks,Fp);
    end
    aver_err_prb=aver_err_prb+smdl_err_prb;
end
smdl_err_prbB=aver_err_prb/exp;
disp('12 de 14');
%*****
thresh=12;%f:DBT
aver_err_prb=zeros(1,L);
for p=1:exp
    smld_err_prb=zeros(1,L);
    for i=1:L
        [smdl_err_prb(1,i),data_bits,Z,H,He]=RScoded_OFDM_DMT_PA(data_frames,
            QAMsimbolos,QAMsize,frec_s,DMTlevel,SNRindB(i),Imp_c,GIR,ploting,
            cicloestacionario,noisetype,thresh,wavfam,wavpar,wavlev,ZD,total_ticks,Fp);
    end
    aver_err_prb=aver_err_prb+smdl_err_prb;
end
smdl_err_prbC=aver_err_prb/exp;
disp('13 de 14');
%*****
thresh=13;
aver_err_prb=zeros(1,L);
for p=1:exp
    smld_err_prb=zeros(1,L);
    for i=1:L
        [smdl_err_prb(1,i),data_bits,Z,H,He]=RScoded_OFDM_DMT_PA(data_frames,
            QAMsimbolos,QAMsize,frec_s,DMTlevel,SNRindB(i),Imp_c,GIR,ploting,
            cicloestacionario,noisetype,thresh,wavfam,wavpar,wavlev,ZD,total_ticks,Fp);
    end
    aver_err_prb=aver_err_prb+smdl_err_prb;
end
smdl_err_prbD=aver_err_prb/exp;
disp('14 de 14');
%*****
disp(fix(clock));
%*****

%*****
%Segue graficaci'on BER vs. SNR:
%*****
%
BER(1,:)=smdl_err_prb0;
BER(2,:)=smdl_err_prb1;
BER(3,:)=smdl_err_prb2;
BER(4,:)=smdl_err_prb3;
BER(5,:)=smdl_err_prb4;
BER(6,:)=smdl_err_prb5;
BER(7,:)=smdl_err_prb6;
BER(8,:)=smdl_err_prb7;

```

```

BER(9,:)=smld_err_prb8;
BER(10,:)=smld_err_prb9;
BER(11,:)=smld_err_prbA;
BER(12,:)=smld_err_prbB;
BER(13,:)=smld_err_prbC;
BER(14,:)=smld_err_prbD;

semilogy(SNRindB,BER);%grafica probabilidad de error.

ylabel('BER');
xlabel('SNR en dB');
title('Gráfica comparativa de técnicas');
%*****
% Fin de script Sim_Monte_Carlo_RScoded_OFDM_PA
%*****

```

E.3.2. Código para la construcción de tramas OFDM

```

%*****
%*****
function [Pe,total_bits,ZD,H,He]=RScoded_OFDM_DMT_PA(nframes,qamsimbolos,ss,fs,
DMTlev,SNR,imp_c,GIR,debug,ciclic,noisetype,thresh,wavfam,wavpar,wavlev,ZD,
total_ticks,Fp)

%*****
%*****
% function Pe=RScoded_OFDM_DMT_PA() para simular un enlace de datos OFDM-DMT
% codificados con código Reed-Solomon.
% y se estima canal con eliminación de ruido con estrategia Pilot-aided (PA)
% incluyendo umbralado en el dominio de la ondeleta.
% !!!Se sugiere su uso para canales estacionarios!!!.
%*****
%
% nframes: el número de tramas que se solicita transmitir
% qamsimbolos: número de símbolo de modulación QAM por símbolo OFDM
% = No. de subportadoras.
% ss: Bits por símbolo de modulación
% fs: frecuencia de muestreo de señal analítica.
% DMTlev: magnitud de la resp. en frecuencia, en dBs, a partir de la cual
% transmitir datos.
% SNR: tasa de señal a ruido (en dBs)
% imp_c: número promedio de impulsos de ruido por ciclo de señal de potencia
% GIR: Gaussian to impulsive power ratio
% debug: - si no es cero, imprime graficas del símbolo A y estado del canal.
%         - si es 2, imprime las Hf generadas durante la transmisión de
%         las tramas.
%ciclic: si es 1 habilita el comportamiento parcialmente ciclo-estacionario
%         (H no cambia durante una trama).
%         si es 2 habilita el comportamiento totalmente ciclo-estacionario
%         (H si cambia durante una trama).
%noisetype: segun el valor agrega ruido a la señal:
% 0: sin ruido
% 1: ruido blanco
% 2: ruido blanco mas ruido impulsivo
%thresh: segun el valor realiza limpieza del estimado de la respuesta en
% frecuencia:
%parametros wavelet:wavfam,wavpar,wavlev (se emplea biblioteca WaveLab850)
%ZD: matriz de impedancias para simular impedancia ciclo-estacionaria en nodo
% terminal D
%total_ticks: marcas de tiempo para actualizar impedancia cicloestacionaria
%Fp: frecuencia de la linea en Hz
%
% Regresa:
% En Pe la probabilidad de error con base a un
% total_bits número de bits transmitidos.
% En ZD la matriz de impedancias para simular impedancia cicloestacionaria.

```

```

% En H la matriz de funciones de transferencia en tiempo-frecuencia empleadas.
% En He la matriz con respuestas en frecuencia estimadas
%*****
%C'odigo para simular un enlace OFDM, en su variante DMT, codificado.
%
% Esta versi'on emplea algoritmo de simulaci'on de canal PLC
% selectivo en frecuencia con ruido impulsivo,
% modulaci'on OFDM, igualaci'on y detecci'on.
% Incorpora simulaci'on de impedancia no lineal variante en el tiempo,
% pero agrega opci'on 'ciclic' para habilitar/deshabilitar comportamiento
% ciclo-estacionario.
%
%*****
%NOTA: -Se RECOMIENDA pasar ciclic=0 para ser congruente con la t'ecnica
%Pilot-aided (PA)
% - Se hace frame_size=256 para ser congruentes con una t'ecnica basada en
% pilotos y canal estacionario.
%*****
%Programador: Gerardo Laguna
%Fecha: 2/12/09
%*****
%*****
%Par'ámetros de trama:
%*****
frame_size=256;%tama'no de trama de datos, en s'imbolos OFDM
%
%*****
%Par'ámetros del canal:
%*****
r_m=0.0084;%reactancia ohms/m
l_m=0.1*10^-6;%inductancia H/m
c_m=3.5*10^-9;%capacitancia F/m
g_m=3*10^-12;%conductancia S/m
L1=15;%longitud de tramo 1 (entre A y B)
L2=15;%longitud de tramo 2 (entre B y C)
L3=9.75;%longitud de la derivaci'on (tramo 3 entre B y D)
ZL=50;%impedancia de carga en el nodo receptor C
Zs=50;%impedancia de fuente en el nodo transmisor A

%*****
%Par'ámetros de comportamiento ciclo-estacionario:
%*****
Tp=1/Fp;%periodo de la l'inea en segs.
%banda util;qamsimbolos= subportadoras para el rango (0,fs]
dtz=(Tp/total_ticks)*1000;% delta de t para tiempo de prueba, en milisegundos
Ts=qamsimbolos*(1/fs);%duraci'on de un s'imbolo OFDM
frac_tick_per_simb=(Ts*1000)/dtz;%duraci'on de un s'imbolo OFDM como fracci'on
%de dtz

%*****
%Par'ámetros del ruido impulsivo a escala de tramas OFDM:
%*****
P=100;%Tama'no de arreglo para generar aleatorios de Poisson
Tf=Ts*frame_size;%duracion de trama de datos
A=(1/imp_c)*(Tp)/(Tf);%tiempo de arribo de pulsos como m'ultiplo de Tf
z=Poisson_interarrivals(P,A);%se genera vector de tiempos de 'inter arribo

T=1;%Periodo de observaci'on (normalizado a tiempo de trama,Tf)
Td=A;%se inicializa tiempo de evento diferido (puede ser cualquier tiempo)
for i=1:nframes+1 %Se genera secuencia de Poisson para total de tramas OFDM
    [Km,Td]=Poisson_N_arrivals2(z,P,T,Td);
    K(1,i)=Km;%secuencia para variable de Poisson
end
%*****

```

```

%*****
%transmitimos tramas de datos:
%*****
current_tick=1; %inicializamos tiempo transcurrido dentro de un ciclo de potencia,
                %en unidades dtz
simbs_per_tick=1/frac_tick_per_simb;%calcula n'umero de s'imbolos OFDM por lapso dtz
total_bits=0;
Pe=0;
j=1; %indice para construir H
for i=1:nframes
    verHfs=0;%De entrada
    if i==nframes%Para la ultima trama, se habilita graficaci'on de conjunto de
                %Hf empleadas:
        if debug==2 %siempre que debug sea 2
            verHfs=2;
        end
    end
end

%*****
%Obtenemos el estado del canal y mascara con trama de reconocimiento
%(Un s'imbolo Piloto A):
%*****
Ka=imp_c*(Ts/Tp);% pulsos arribados, en proporci'on, para s'imbolo A
[lastH_e,Mask,A_errores]=OFDM_pilot_PA(qamsimbolos,ss,fs,DMTlev,SNR,Ka,GIR,
0,r_m,l_m,c_m,g_m,L1,L2,L3,ZL,Zs,ZD,current_tick,noisetype,thresh,wavfam,
                                wavpar,wavlev);
if ciclic>0 %aumenta la duraci'on del s'imbolo A en unidades dtz
    current_tick=current_tick+floor(frac_tick_per_simb);
end

%*****
%Transmitimos trama de datos (Un s'imbolo Piloto A + DATOS UTILES):
%*****
[frame_bits,frame_Pe,Htf,H_e]=RScoded_OFDM_frame_PA(frame_size,qamsimbolos,
ss,fs,DMTlev,SNR,K(1,i),GIR,verHfs,Mask,lastH_e,r_m,l_m,c_m,g_m,L1,L2,L3,
ZL,Zs,ZD,current_tick,simbs_per_tick,ciclic,noisetype,thresh,wavfam,wavpar,
                                wavlev);

[n,m]=size(Htf);
H(j:j+n-1,:)=Htf;
He(i,:)=H_e;
j=j+n;%siguiente espacio para construir H

if ciclic>0
    current_tick=current_tick+floor(Tf*1000/dtz);% se aumenta la duraci'on de
                                                % la trama, en unidades dtz
    if current_tick>total_ticks current_tick=1; end% si se rebasa la duraci'on
                                                %de un ciclo; se vuelve al
                                                %inicio
end

Pe=Pe*total_bits/(total_bits+frame_bits)+
    frame_Pe*frame_bits/(total_bits+frame_bits);
total_bits=total_bits+frame_bits;
end

%*****
% Fin de funci'on RScoded_OFDM_DMT_PA()
%*****

%*****
function [H_e,Mask,errores]=OFDM_pilot_PA(qamsimbolos,ss,fs,DMTlev,SNR,imp_s,GIR,
debug,r_m,l_m,c_m,g_m,L1,L2,L3,ZL,Zs,ZD,tick_ini,noisetype,thresh,wavfam,wavpar,
                                wavlev)

```

```

%*****
%*****
% function OFDM_pilot_PA() para generar trama de reconocimiento
% del estado del canal y estimar canal con diferentes t'ecnicas.
% !!!SE OPTIMIZAN PARAMETROS DE ENTRADA/SALIDA PARA SU USO CON ESTRATEGIA
% PILOT-AIDED (PA)!!!
%
% qamsimbolos: n'umero de s'imbolo de modulaci'on QAM por s'imbolo OFDM
% ss: Bits por s'imbolo de modulaci'on QAM
% fs: frecuencia de muestreo de se'nal anal'itica.
% DMTlev: magnitud de la resp. en frecuencia, en dB, a partir de la cual
%         %transmitir datos.
% SNR: tasa de se'nal a ruido (en dB)
% imp_s: n'umero promedio de impulsos de ruido por s'imbolo A OFDM
% GIR: Gaussian to impulsive power ratio
% debug: - si no es cero, imprime graficas del s'imbolo A y estado del canal.
%         - si es 2, imprime la Hf empleada durante la trasmisi'on del
%         s'imbolo A.
%r_m: reactancia ohms/m
%l_m: inductancia H/m
%c_m: capacitancia F/m
%g_m: conductancia S/m
%L1: longitud de tramo 1 (entre A y B)
%L2: longitud de tramo 2 (entre B y C)
%L3: longitud de la derivaci'on (tramo 3 entre B y D)
%ZL: impedancia de carga en el nodo receptor C
%Zs: impedancia de fuente en el nodo transmisor A
%ZD: matriz de impedancias para simular impedancia ciclo-estacionaria en nodo
%     %terminal D
%tick_ini: representa el tiempo de inicio relativo dentro de un ciclo de potencia,
%           indicando el rengl'on (impedancias en funci'on de la frecuencia) de la
%           matriz ZD en el que se va a emplear.
%
%noisetype: seg'un el valor agrega ruido a la se'nal:
%           0: sin ruido
%           1: ruido blanco
%           2: ruido blanco mas ruido impulsivo
%thresh: segun el valor realiza limpieza del estimado de la respuesta en
%par'ámetros wavelet:wavfam,wavpar,wavlev (se emplea biblioteca WaveLab850)
%
%regresa:
% En Mask regresa la mascara de habilitaci'on de subportadoras
% en errores regresa el n'umero de errores detectados en los bits recibidos.
% En H_e la respuesta en frecuencia estimada
%*****
% Versi'on que prueba algoritmo de simulaci'on de canal PLC
% selectivo en frecuencia con ruido impulsivo e impedancia ciclo-estacionaria.
% Modulaci'on OFDM, igualaci'on, detecci'on y
% estimaci'on de canal mediante s'imbolo A.
% ('unicamente se transmite un s'imbolo A)
%
%*****
%*****
%Programador: Gerardo Laguna
%Fecha:9/10/09
%*****

%*****
%Inicializaci'on del sistema
%*****
[nbits,N,Ts]=finiofdmqamb(qamsimbolos,ss,fs);%inicializa variables del sistema;
%En nbits, el n'umero de bits por s'imbolo OFDM
%En N el tama'no de la FFT= n'umero de s'imbolos QAM de la se'nal anal'itica con
%     %simetria Hermitiana
Dprop=0.15;% proporci'on de la duraci'on del s'imbolo que toma la dispersi'on de la

```

```

%respuesta al impulso
D=floor(Dprop*N);%Calculamos el intervalo de guardia como proporci'on de la
%duracion de un simbolo OFDM
L_spread=floor(D/2);% muestras 'utiles de la respuesta al impulso con base a la
%se'nal anal'itica
TramaSize=1; %Tama'no de trama en s'imbolos OFDM

%Generaci'on de datos para s'imbolo A:
XA=bit_simbolA2(nbits);%simbolo A
[xA_real,XA_analitic]=fodmqamtX02(XA,qamsimbolos,ss); %simbolo A OFDM
xAofdm=faddgi(xA_real,D); %se agrega intervalo de guardia de longitud D

%*****
%Inicielizaci'on de par'ámetros del canal:
%*****
% D
% A___|___C
% B
%Banda de frecuencia 'util, en radianes:
%qamsimbolos= subportadoras para el rango (0,fs]:
B=(2*pi*fs)/qamsimbolos:(2*pi*fs)/qamsimbolos:2*pi*fs;
[ht_c,Hf_c]=Channel_PLC07C(Dprop,B,r_m,l_m,g_m,c_m, L1,L2,L3, ZL,Zs,ZD,tick_ini);
if debug==2
    figure;
    subplot(2,1,1);
    plot(20*log10(abs(Hf_c)));
    title('Hc(f) te'orica inicial');
    ylabel('dB');
    subplot(2,1,2);
    plot(miatan2(imag(Hf_c),real(Hf_c)));
    title('Fase Hc(f) inicial');
end

%*****
%Construye tramas
%*****
%Construimos trama con un s'imbolo A:

datos=[XA];
total_bits=nbits;
s_ofdm=[xAofdm];
%Transmitimos trama:
sn=s_ofdm;

%filtro transmisor:
%s=filter(bf,1,sn);
r=conv(sn,ht_c); %simulamos canal PLC selectivo en frecuencia
%filtro receptor empatao:
%r=filter(bf,1,s);

%*****
%Simulaci'on de ruido:
%*****
rn=r; %De entrada;
if noisetype ~= 0
    %componente gaussiana:
    Pr=CalculaPotencia(abs(r), ZL);%calcula la potencia de la se'nal recibida en watts
    Pg=Pr/(10^(SNR/10));%calcula la potencia del ruido gaussiano, en watts,
    %para cierta SNR, en dBs
    bg=wgn(1,length(r),Pg,ZL,'linear');%genera ruido aditivo gaussiano
    rn=r+bg;%adiciona ruido gaussiano
    if noisetype>1
        %componente impulsiva:
        if imp_s>0 %si hay incidencia de impulsos durante el s'imbolo A:
            Pi=Pg/GIR;%calcula la potencia del ruido impulsivo

```

```

        bi=impn2(imp_s,length(r),Pi,ZL,'linear');% genera ruido impulsivo
        rn=rn+bi;%adiciona ruido impulsivo
    end
end
end
%
if debug~=0
    figure
    subplot(2,2,1);
    plot(abs(r));
    title('Señal OFDM de símbolo A');
    subplot(2,2,2);
    plot(abs(rn));
    title('Símbolo A con ruido');
    subplot(2,2,3);
    plot(abs(rn-r));
    title('Ruido se símbolo A');
    if noisetype>1
        if imp_s>0 %si hay incidencia de impulsos durante el símbolo A:
            subplot(2,2,4);
            plot(bi);
            title('solo Ruido impulsivo');
        end
    end
end
end
r=rn; %señal mas ruido
%*****
%Inicio de trama:
%Se asume que ya se ha sincronizado el inicio de la trama así como
%sincronización fina.
%*****
% inicio de símbolo (En principio el símbolo A):
ix=D+1;% ajuste fino ('índice justo uno después de terminar el intervalo de guardia)
rn=r(ix:ix+N-1); %ventana FFT sincronizada en forma fina

%Se demodula a la señal OFDM, pero ahora
%sincronizada en forma fina:
Yn=fofdmqamrx(rn,qamsimbolos,N); %IFFT de tamaño N y recuperación de señal
%analítica con qamsimbolos

if debug~=0
    figure;
    subplot(2,2,1);
    plot(abs(XA_analitic));
    title('Magnitud símbolo A');
    subplot(2,2,2);
    plot(miatan2(imag(XA_analitic),real(XA_analitic)));
    title('Fase símbolo A');
    subplot(2,2,3);
    plot(abs(Yn));
    title('Magnitud recibida (sic. Fina) antes de igualar');
    subplot(2,2,4);
    plot(miatan2(imag(Yn),real(Yn)));
    title('Fase recibida (sinc. Fina) antes de igualar');
end
%*****
% Estimación de canal PLC con símbolo A
% mediante filtro empatado a símbolo A
%*****
% Estimación de canal PLC con símbolo A
% mediante expresión H=Y/X
%*****
H_en=Yn./XA_analitic;

```

```

if debug==1
    figure
    subplot(2,1,1);
    plot(20*log10(abs(H_en)));
    title('H(f) del canal (Y/X) en bruto');
    ylabel('dB');
    subplot(2,1,2);
    plot(miatan2(imag(H_en),real(H_en)));
    title('Fase H(f) en bruto');
end

%*****
%Estimaci'on del canal:
%*****
H_e=ChannelEstimation(H_en,qamsimbolos,L_spread,XA_analitic,thresh,
                    wavfam,wavpar,wavlev);

%*****
% Obtenci'on de mascara para subportadoras 'utiles
%*****
%como la magnitud m'inima transmitida es 1 en cualquier caso (BPSK, 4QAM, 16QAM):
Vr_min=abs(H_e); %magnitud m'inima recibida
umbral=10^(DMTlev/20);%umbral a partir del cual se habilita la subportadora
Mask=zeros(1,qamsimbolos);%mascara inicial
for i=1:qamsimbolos
    %si despu'es de la atenuaci'on de la subportadora la se'nal rebasa el umbral:
    if Vr_min(i)>umbral
        Mask(i)=1;% entonces se habilita la subportadora
    end
end

%*****
%Igualaci'on de datos de s'imbolo A
%*****
%Hp=diag(Psi_p);
%g=1./Psi_p; %prueba igualaci'on con base a salida de filtro empatado a
%simbolo A
gk=1./H_e; %prueba igualaci'on con base a la divisi'on H=Y/X
G=diag(gk);
Yg=(G*Yn.'.');

if debug~=0
    figure;
    subplot(2,2,1);
    plot(abs(Yg));
    title('Magnitud s'imbolo A igualado');
    subplot(2,2,2);
    plot(miatan2(imag(Yg),real(Yg)));
    title('Fase s'imbolo A igualado');
    subplot(2,2,3);
    plot(abs(Yg.*Mask));
    title('Magnitud s'imbolo A igualado & Mask detectado');
end

%*****
%Detecci'on de datos de s'imbolo A
%*****
Yr=detect_bits(nbits,Yg,ss);
dsix=1;%data source index
Xs=datos(dsix:dsix+nbits-1);
errores=ErroresRxTx(Yr,Xs);

%*****
%Fin de funci'on OFDM_pilot_PA()
%*****

```

```

%*****
%*****
function [total_bits,Pe,Htf,H_e]=RScoded_OFDM_frame_PA(TramaSize,qamsimbolos,ss,
fs,DMTlev,SNR,imp_f,GIR,debug,Mask,lastH_e,r_m,l_m,c_m,g_m,L1,L2,L3,ZL,Zs,ZD,
tick_ini,simbols_in_tick,ciclic,noisetype,thresh,wavfam,wavpar,wavlev)
% function RScoded_OFDM_frame_PA()para generar una trama de datos codificados
% con Reed-Solomon y estimar canal con diferentes t'ecnicas.
%*****
%*****
% !!!SE OPTIMIZAN PARAMETROS DE ENTRADA/SALIDA PARA SU USO CON
% ESTRATEGIA PILOT-AIDED (PA)!!!
%*****
%
%Tramaze: tama'no de trama de datos, en s'imbolos OFDM
% qamsimbolos: n'umero de s'imbolos de modulaci'on QAM por s'imbolo OFDM
% ss: Bits por s'imbolo de modulaci'on QAM
% fs: frecuencia de muestreo de se'nal anal'itica.
% DMTlev: magnitud de la resp. en frecuencia, en dB, a partir de la cual
%transmitir datos.
% SNR: taza de se'nal a ruido (en dBs)
% imp_f: n'umero promedio de impulsos de ruido por trama OFDM
% GIR: Gaussian to impulsive power ratio
% debug: - si es 1, imprime graficas del s'imbolo A y estado del canal.
%         - si es 2, imprime las Hf generadas durante la trasmisi'on de la
%         trama.
% Mask: es un vector de 1s y 0s para habilitar/deshabilitar las subportadoras
% lastH_e: es un vector con la 'ultima respuesta en frecuencia estimada
%r_m: reactancia ohms/m
%l_m: inductancia H/m
%c_m: capacitancia F/m
%g_m: conductancia S/m
%L1: longitud de tramo 1 (entre A y B)
%L2: longitud de tramo 2 (entre B y C)
%L3: longitud de la derivaci'on (tramo 3 entre B y D)
%ZL: impedancia de carga en el nodo receptor C
%Zs: impedancia de fuente en el nodo transmisor A
%ZD: matriz de impedancias para simular impedancia ciclo-estacionaria en nodo
% terminal D.
%tick_ini: representa el tiempo de inicio, relativo dentro de un ciclo de potencia,
%          indicando el rengl'on (impedancias en funci'on de la frecuencia) de la
%          matriz ZD en el que se va a emplear.
%simbols_in_tick: s'imbolos OFDM en un tick
%ciclic: si es 2 habilita el comportamiento ciclo-estacionario.
%noisetype: seg'un el valor agrega ruido a la se'nal:
%          0: sin ruido
%          1: ruido blanco
%          2: ruido blanco mas ruido impulsivo
%thresh: seg'un el valor realiza limpieza de la estimaci'on
%parametros wavelet:wavfam,wavpar,wavlev (se emplea biblioteca WaveLab850)
%
%regresa:
% En total_bits regresa el n'umero de bits 'utiles transmitidos
% En Pe regresa la probabilidad de error
% En Htf la matriz de funciones de transferencia en tiempo-frecuencia empleadas.
% En H_e la respuesta en frecuencia estimada
%*****
%C'odigo para simular sistema OFDM en su variante DMT
%(solo se usan subportadoras con buena SNR) con datos codificados por
%c'odigo Reed-Solomon.
%
% Versi'on que prueba algoritmo de simulaci'on de canal PLC
% selectivo en frecuencia con ruido impulsivo,
% modulaci'on OFDM, igualaci'on, detecci'on e impedancia ciclo-estacionaria,
% pero agrega opci'on 'ciclic' para habilitar/deshabilitar comportamiento
% ciclo-estacionario.

```

```

%*****
%*****
%Programador: Gerardo Laguna
%Fecha:2/12/09
%*****

%*****
%Inicializaci'on del sistema
%*****
[nbits,N,Ts]=finiofdmqamb(qamsimbolos,ss,fs);%inicializa variables del sistema;
%En nbits, el n'umero de bits por s'imbolo OFDM
%En N el tama'no de la FFT= n'umero de s'imbolos QAM de la se'nal anal'itica con
%simetr'ia Hermitiana
Dprop=0.15;% proporci'on de la duraci'on del s'imbolo que toma la dispersi'on de la
%respuesta al impulso
D=floor(Dprop*N);%Calculamos el intervalo de guardia como proporci'on de la
%duraci'on de un s'imbolo OFDM
L_spread=floor(D/2);% muestras 'utiles de la respuesta al impulso con base a la
%se'nal analitica

%Generacion de datos para s'imbolo A:
XA=bit_simbola2(nbits);%s'imbolo A
[xA_real,XA_analitic]=fofdmqamtX02(XA,qamsimbolos,ss); %s'imbolo A OFDM
xAofdm=faddgi(xA_real,D); %se agrega intervalo de guardia de longitud D

%*****
%Par'ametros del ruido impulsivo a escala de s'imbolos OFDM:
%*****
if imp_f>0 %si en esta trama incide alg'un pulso
    P=100; %Tama'no de arreglo para generar aleatorios de Poisson
    A=TramaSize/imp_f;%tiempo de arribo de pulsos como m'ultiplo de Ts
    z=Poisson_interarrivals(P,A); %se genera vector de tiempos de 'inter arribo

    T=1; %Periodo de observaci'on (normalizado a tiempo de simbolo, Ts)
    Td=A; %se inicializa tiempo de evento diferido (puede ser cualquier tiempo)
    for i=1:TramaSize %Se genera secuencia de Poisson para total de s'imbolos OFDM
        [Km,Td]=Poisson_N_arrivals2(z,P,T,Td);
        K(1,i)=Km;%secuencia para variable de Poisson
    end
end%if algun pulso
%*****
%Par'ametros de c'odigo Reed-Solomon (RS):
%*****
m_rs=8; %columnas de matriz de datos (bits por byte)
r_rs=15; %renglones redundantes de la matriz de c'odigo RS
n_rs=(2^m_rs)-1; %renglones totales de la matriz de c'odigo RS
k_rs=n_rs-r_rs; %renglones de datos en la matriz de c'odigo RS
genpoly=rspoly(n_rs,k_rs); %matriz de polinomio generador del c'odigo RS
%*****
%Construye tramas
%*****
%Construimos tramas con un s'imbolo A y TramaSize-1 s'imbolos con c'odigos RS a
%partir de datos aleatorios:
capacidad_simbolo=maped_bit_count(qamsimbolos,ss,Mask);%capacidad de bits
%transmitibles por simbolo OFDM

capacidad_trama=(TramaSize-1)*capacidad_simbolo;
Xrs=zeros(1,capacidad_trama);%de entrada la cadena con datos a transmitir
size_rs=n_rs*m_rs;%tama'no (en bits) de bloque codificado con Reed-Solomon
size_data_rs=k_rs*m_rs;%tama'no (en bits) de datos dentro del bloque codificado
%con Reed-Solomon
nbloques_rs=floor(capacidad_trama/size_rs);%n'umero de bloques Reed-Solomon
Xmsg=[];%cadena con datos 'utiles
ddix=1; %data destination index
for i=1:nbloques_rs

```

```

Xu=bits_source_on_off(size_data_rs);%se generan bits ON/OFF 'utiles en forma
                                %aleatoria

%Xu=ones(1,size_data_rs);
Xmsg=[Xmsg, Xu];%solo datos utiles sin codificar
msg_tx=reshape(Xu,k_rs,m_rs);%arregla la secuencia como arreglo de bytes
Xcode_rs=encode(msg_tx,n_rs,k_rs,'rs/binary',genpoly);
Xcode_rs_s=reshape(Xcode_rs,1,size_rs);%arregla arreglo de bytes como secuencia
Xrs(ddix:ddix+size_rs-1)=Xcode_rs_s;%datos a transmitir codificados,
                                %serializados y en formato on/off
ddix=ddix+size_rs; %actualiza data destination index
end

s_ofdm=[xAofdm];
total_bits=nbits; %de entrada acumulamos los bits del s'imbolo piloto
                                %(todos son utiles)
si=1; %source index
for i=1:(TramaSize-1)
    Xrsp=Xrs(si:si+capacidad_simbolo-1);
    si=si+capacidad_simbolo; %actualiza source index
    Xrspa=onoff_2_antipodal(Xrsp); % pasamos los bits de formato on-off a antipodal
    %Mapea bits 'utiles en secuencia para que ocupen subportadoras habilitadas:
    [Xrspam,sbits]=DMT_map_bits(Xrspa,qamsimbolos,ss,Mask);
    total_bits=total_bits+size_data_rs;%solo consideramos los bits 'utiles de los
                                %c'odigos Reed-Solomon

    [x_real,X_analitic]=fofdmqamtX02(Xrspam,qamsimbolos,ss); %s'imbolo OFDM
    xofdm=faddgi(x_real,D); %se agrega intervalo de guardia de longitud D
    s_ofdm=[s_ofdm, xofdm];
end

%Transmitimos trama:
sn=s_ofdm;

%filtro transmisor:
%s=filter(bf,1,sn);
%*****
% Simulaci'on de canal PLC con impedancia ciclo-estacionaria:
%*****
% D
% A__|__C
% B
%Banda de frecuencia 'util, en radianes:
B=[(2*pi*fs)/qamsimbolos:(2*pi*fs)/qamsimbolos:2*pi*fs]; %qamsimbolos= subportadoras
                                %para el rango (0,fs]
if ciclic==2
    total_ticks=floor(TramaSize/simbols_in_tick);%duraci'on de la trama en ticks
    if total_ticks==0%validacion
        total_ticks=1;
    end
else
    total_ticks=1;
end
%simulamos canal PLC con convoluci'on por partes:
[r,Htf]=Conv_Channel_PLC07C(Dprop,sn,B,r_m,l_m,g_m,c_m, L1,L2,L3,ZL,Zs,ZD,
                                tick_ini,total_ticks,debug);
%*****
%filtro receptor empatado:
%r=filter(bf,1,s);

%*****
%Simulaci'on de ruido:
%*****
errors=0;
rn=r; %De entrada;
if noisetype ~= 0

```

```

%componente gaussiana:
ix=N+D; %indice a inicio aproximado de primer s'imbolo de datos
Pr=CalculaPotencia(abs(r), ZL);%calcula la potencia de la se'nal recibida en watts
Pg=Pr/(10^(SNR/10));%calcula la potencia del ruido gaussiano, en watts,
                %para cierta SNR, en dBs
bg=wgn(1,length(r),Pg,ZL,'linear');%genera ruido aditivo gaussiano
rn=rn+bg;%adiciona ruido
%
if noisetype>1
    %componente impulsiva:
    if imp_f>0 %si hay incidencia de impulsos en la trama:
        Pi=Pg/GIR; %calcula la potencia del ruido impulsivo
        ix=1; %indice a inicio de primer s'imbolo
        for i=1:TramaSize%para cada s'imbolo OFDM
            if K(1,i)>0 %si hay incidencia de impulsos en el s'imbolo en turno:
                bi=impn2(K(1,i),N+D,Pi,ZL,'linear'); % genera ruido impulsivo
                %adiciona ruido impulsivo a las muestras del simbolo:
                rn(ix:ix+N+D-1)=rn(ix:ix+N+D-1)+bi;
            end
            ix=ix+N+D; %inicio de siguiente s'imbolo
        end
    end
end
end
r=rn; %Se'nal con ruido
%
%*****
%Inicio de trama:
%Se asume que ya se ha sincronizado el inicio de la trama as'i como
%sincronizaci'on fina.
%*****
% inicio de simbolo (En principio el simbolo A):
ix=D+1;% ajuste fino ('indice justo uno despu'es de terminar el intervalo de guardia)
rn=r(ix:ix+N-1); %ventana FFT sincronizada en forma fina

%Se demodula a la se'nal OFDM, pero ahora
%sincronizada en forma fina:
Yn=fofmdmqamrx(rn,qamsimbolos,N); %IFFT de tama'no N y recuperaci'on de se'nal
                %anal'itica con qamsimbolos

if debug==1
    figure;
    subplot(2,2,1);
    plot(abs(XA_analitic));
    title('Magnitud simbolo A');
    subplot(2,2,2);
    plot(miatan2(imag(XA_analitic),real(XA_analitic)));
    title('Fase simbolo A');
    subplot(2,2,3);
    plot(abs(Yn));
    title('Magnitud recibida (sic. fina) antes de igualaci'on');
    subplot(2,2,4);
    plot(miatan2(imag(Yn),real(Yn)));
    title('Fase recibida (sinc. fina) antes de igualaci'on');
end
%*****
% Estimaci'on de canal PLC con s'imbolo A
% mediante expresi'on H=Y/X
%*****
H_en=Yn./XA_analitic;

if debug==1
    figure
    subplot(2,1,1);
    plot(20*log10(abs(H_en)));

```

```

        title('H(f) del canal (Y/X) en bruto');
        ylabel('dB');
        subplot(2,1,2);
        plot(miata2(imag(H_en),real(H_en)));
        title('Fase H(f) en bruto');
end

%*****
%Estimaci'on del canal:
%*****
H_e=ChannelEstimation(H_en,qamsimbolos,L_spread,XA_analitic,thresh,
                    wavfam,wavpar,wavlev);

%Atenuacion de cambios bruscos:
epsilon=0.5;
H_e=epsilon*H_e+(1-epsilon)*lastH_e;
%*****
if debug==1
    figure
        subplot(2,1,1);
        plot(20*log10(abs(H_e)));
        title('H(f) estimada del canal');
        ylabel('dB');
        subplot(2,1,2);
        plot(miata2(imag(H_e),real(H_e)));
        title('Fase de H(f) estimada');
end
%*****
%Igualaci'on de datos de s'imbolo A
%*****
g=1./H_e; %prueba igualaci'on con base a la divisi'on H=Y/X
G=diag(g);
Yg=(G*Yn.').';

if debug==1
    figure;
        subplot(2,2,1);
        plot(abs(Yg));
        title('Magnitud s'imbolo A igualado');
        subplot(2,2,2);
        plot(miata2(imag(Yg),real(Yg)));
        title('Fase s'imbolo A igualado');
        subplot(2,2,3);
        plot(abs(Yg.*Mask));
        title('Magnitud s'imbolo A igualado & Mask');
end
%*****
%Detecci'on de datos de s'imbolo A
%*****
Yr=detect_bits(nbits,Yg,ss);
Xs=XA;
A_errors=ErroresRxTx2(Yr,Xs);
%*****
%Recepci'on del resto de la trama:
%*****
Yrs=[];%cadena con datos recibidos
for i=1:(TramaSize-1)
    %Ventaneo:
    ix=ix+N+D;% siguiente ventana
    rn=r(ix:ix+N-1); %ventana FFT
    %*****
    %
    %Demodulacion de la se'nal OFDM:
    Yn=fofdmgamrx(rn,qamsimbolos,N); %IFFT de tama'no N y recuperaci'on de se'nal

```

```

                                %anal'itica con qamsimbolos

%Iguacion:
Yg=(G*Yn.').';
%Deteccion:
[Yrspam,Y_analitic]=detect_bits03(nbits,Yg,ss,Mask);
%para recuperar la secuencia codificada mapeada en las portadoras habilitadas:
Yrspa=DMT_unmap_bits(Yrspam,capacidad_simbolo,qamsimbolos,ss,Mask);
Yrsp=antipodal_2_onoff(Yrspa);%pasamos los datos a valores on-off
Yrs=[Yrs,Yrsp];%El tama'no final de Yrs es la capacidad de la trama.
                                %Contiene bloques completos RS y lo que sobra con ceros
end%for

dsix=1;%data source index
csix=1;%code source index
Ymsg=[];
for i=1:nbloques_rs
    Ycode_rs_s=Yrs(csix:csix+size_rs-1);%datos recibidos codificados,
                                %serializados y en formato on/off
    csix=csix+size_rs;%actualiza data source index
    Ycode_rs=reshape(Ycode_rs_s,n_rs,m_rs);%arregla la secuencia como arreglo
                                %de bytes (Bloque RS)
    msg_rx=decode(Ycode_rs,n_rs,k_rs,'rs/binary');
    Yu=reshape(msg_rx,1,size_data_rs);
    Ymsg=[Ymsg,Yu];
    Xu=Xmsg(dsix:dsix+size_data_rs-1);
    dsix=dsix+size_data_rs;%actualiza data source index
    errors=errors+ErroresRxTx2(Yu,Xu);
end
Pe=errors/total_bits;
%*****
%Fin de funci'on RScoded_OFDM_frame_PA().
%*****

```

E.3.3. Código para la simulación del canal PLC

```

%*****
%*****
function Z=GenCicloest04(f,t,alfa,w,sigma,beta,T,
                        K,r1,r2,r3,r4,r5,r6,r7,Et,Ef1,Ef2,Ef3)
%*****
%*****
%    function Z=GenCicloEst03()
% Genera la matriz Z de impedancias (valores complejos)
% de un dispositivo con comportamiento
% ciclo-estacionario para el rango determinado por el vector t y el vector f

% t: arreglo con valores para la variable temporal en milisegundos.
% f: arreglo con valores para la variable de frecuencia normalizada con
%    un m'aximo en 2Pi
% alfa: impedancia m'axima aproximada en la frecuencia w
% w: frecuencia con m'aximo local normalizada respecto a 2Pi
% sigma: desviaci'on est'andar en el eje de la frecuencia para la envolvente
%    gaussiana para el
% m'aximo local en w.
% beta: impedancia m'axima aproximada en la frecuencia de 0 Hz.
% T: el periodo del ciclo de potencia en mseg.
% K: offset para evitar valores negativos de impedancia (t'ipicamente 0.3)
% r1 a r7: valores entre -0.5 y 0.5 para generar variaciones en la amplitud en:
% - r1 y r2 del primer y segundo pulsos sinc() en el dominio del tiempo.
% - r3 y r5 en el valor de beta para el primer y segundo semiciclo.
% - r4 y r6 en la amplitud de la envolvente gaussiana en frecuencia para el
%    primer y segundo semiciclo.
% - r7 en las fronteras entre semiciclos.
% Et: Constante de escalamiento para envolvente de la funci'on sinc() en
%    el dominio temporal (t'ipicamente 1).

```

```

% Ef1: Constante de escalamiento para envolvente exponencial en el dominio
%   en frecuencia (t'ipicamente 1).
% Ef2: Constante de escalamiento para envolvente sinc() en el dominio
%   en frecuencia (t'ipicamente 1).
% Ef3: Constante de escalamiento para envolvente cos() en el dominio
%   en frecuencia (t'ipicamente 1).
%
%*****
%Programador: Gerardo Laguna
%Fecha:29/09/09
%*****

t_len=length(t);
f_len=length(f);
for i=1:t_len
    for k=1:f_len
        R(i,k)=ImpCicloest04(t(i),f(k),alfa,w,sigma,beta,T,K,
                               r1,r2,r3,r4,r5,r6,r7,Et,Ef1,Ef2,Ef3);
    end
end
%obtenemos impedancia compleja a partir de la resistencia como si se
%tratara de la magnitud:
[realpart,y,x]=getmin(R);
Z=realpart+j*(sqrt((R.*R)-realpart^2));
%*****
% Fin funci'on GenCicloest04()
%*****

%*****
%*****
function Imp=ImpCicloest04(t,f,alfa,w,sigma,beta,T,K,r1,r2,r3,r4,r5,r6,r7,
                           Et,Ef1,Ef2,Ef3)

%*****
%*****
%function Z=ImpCicloEst03()
% Simula la resistencia (unicamente valores reales) de un dispositivo
% con comportamiento ciclo-estacionario.
% t: variable de tiempo en milisegundos.
% f: variable de frecuencia normalizada con un maximo en 2Pi
% alfa: impedancia m'axima aproximada en la frecuencia w
% w: frecuencia con m'aximo local normalizada respecto a 2Pi
% sigma: desviaci'on est'andar en el eje de la frecuencia para la envolvente
% gaussiana para el m'aximo local en w.
% beta: impedancia m'axima aproximada en la frecuencia de 0 Hz.
% T: el periodo del ciclo de potencia en mseg.
% K: offset para evitar valores negativos de impedancia (t'ipicamente 0.3)
% r1 a r7: valores entre -0.5 y 0.5 para generar variaciones en la amplitud en:
% - r1 y r2 del primer y segundo pulsos sinc() en el dominio del tiempo
% - r3 y r5 en el valor de beta para el primer y segundo semiciclo
% - r4 y r6 en la amplitud de la envolvente gaussiana en frecuencia para el
% primer y segundo semiciclo.
% - r7 en las fronteras entre semiciclos.
% Et: Constante de escalamiento para envolvente de la funci'on sinc() en
% el dominio temporal (tipicamente 1).
% Ef1: Constante de escalamiento para envolvente exponencial en el dominio
% en frecuencia (t'ipicamente 1).
% Ef2: Constante de escalamiento para envolvente sinc() en el dominio
% en frecuencia (t'ipicamente 1).
% Ef3: Constante de escalamiento para envolvente cos() en el dominio
% en frecuencia (t'ipicamente 1).
%
%Regresa en Imp la impedancia como n'umero complejo
%imaginaria, respectivamente.
%
%*****

```

```

%Programador: Gerardo Laguna
%Fecha:29/09/09
%*****
gt=alfa*(1+r1)*sinc(Et*(t-T/4))+alfa*(1+r2)*sinc(Et*(t-3/4*T))+alfa*K;
tao=(1+r7)*T/8;
if (0<t) & (t<=tao)
    z=(gt+beta*(1+r3))*exp(-Ef1*f);
else
    if (tao<t) & (t<=(T/2-tao))
        z=gt*(1+r4)*migaussiana02(f,w,sigma)*((1/2)*abs(sinc(Ef2*(f-w)))
            +abs(cos(Ef3*(f-w))))+3);
    else
        if ((T/2-tao)<t) & (t<=(T/2+tao))
            z=(gt+beta*(1+r5))*exp(-Ef1*f);
        else
            if ((T/2+tao)<t) & (t<=(T-tao))
                z=gt*(1+r6)*migaussiana02(f,w,sigma)*((1/2)*abs(sinc(Ef2*(f-w)))
                    +abs(cos(Ef3*(f-w))))+3);
            else
                z=(gt+beta*(1+r3))*exp(-Ef1*f);
            end
        end
    end
end
Imp=z+alfa*K;
%*****
%Fin funci'on ImpCicloest04()
%*****

%*****
%*****
function [xc,Htf]=Conv_Channel_PLC07C(Dprop,x,W,r,l,g,c, L1,L2,L3, ZL,Zs,ZD,t0,
    partes,debug)

%*****
%*****
% Conv_Channel_PLC07C() Convoluciona por partes una secuencia de
% entrada x para simular el efecto de un canal variante debido a una impedancia
% ciclo-estacionaria.
% Se obtiene la respuesta al impulso (ht), en funci'on de ZD(t), empezando
% en t0, y se convoluciona con el segmento de x que le corresponde a cada
% tiempo. El tiempo t corresponde a un rengl'on de ZD y, cada vez que se
% termina la convoluci'on de un segmento, t se incrementa en uno.
% Se asume una red PLC de topolog'ia T
% en el rango de frecuencia del vector W y los par'ámetros de la red.

% Dprop: Proporción, respecto de la duración del símbolo,
%         que toma la dispersión de la respuesta al impulso.
% W: Vector de frecuencia en radianes.
% (la separación entre frecuencias determina el tiempo de muestreo)
% r: resistencia por metro
% l: inductancia por metro
% g: conductancia por metro
% c: capacitancia por metro
% L1: longitud de tramo 1 en metros
% L2: longitud de tramo 2 en metros
% L3: longitud de tramo 3 en metros
% ZL: impedancia de carga (Recetor)
% Zs: Impedancia de fuente (Transmisor)
% ZD: matriz de impedancias para simular impedancia ciclo-estacionaria en
%     nodo terminal D
% t0: el rengl'on inicial de la matriz ZD que se va a emplear
% partes: es el número de segmentos en el que se divide a la señal para
%         convolucionarlos con la respuesta del canal correspondiente al rengl'on
%         de ZD en turno.
% debug: Si su valor es 2 grafica las Hf correspondientes a las respuestas impulso

```

```

%      empleadas a lo largo de la convoluci'on por partes.
%
%Regresa en Htf la matriz de funciones de transferencia en tiempo-frecuencia.
%*****
%*****
%Programador: Gerardo Laguna
%Fecha:5/10/09
%*****
[RZD,CZD]=size(ZD);
x_len=length(x);
part_len=floor(x_len/partes);
ht_len=length(W);%en nuestro caso, la respuesta al impulso tiene tantas muestras
                %como subportadoras.
conv_len=part_len+ht_len-1;
y=zeros(partes+1,conv_len);
Htf=zeros(partes,ht_len);
row=t0;
ix=1;
for i=1:partes
    [ht_c,Hf_c]=Channel_PLC07C(Dprop,W,r,l,g,c, L1,L2,L3, ZL,Zs,ZD,row);
    row=row+1;%siguiente renglon de ZD
    if row>RZD
        row=1;%validacion de row
    end
    sx=x(ix:ix+part_len-1);%segmento en turno
    ix=ix+part_len;%indice a siguiente segmento de x
    y(i,:)=conv(sx,ht_c);
    Htf(i,:)=Hf_c;
end%for

if ix<x_len+1 %si restan datos por convolucionar:
    sx=x(ix:x_len);%ultimo segmento
    c=conv(sx,ht_c);
    y(i+1,1:length(c))=c;
end%si restan datos
%*****
%sumamos las convoluciones parciales:
%*****
%Se asume que el segmento es mayor que la respuesta impulso. En este caso
%que el n'umero de muestras del segmento sea mayor o igual al n'umero de muestras
%de la respuesta al impulso, que aqu'i es del tama'no de un s'imbolo. Es decir,
%que el segmento corresponda a por lo menos un s'imbolo OFDM.
xc=y(1,:);
for i=2:partes+1
    xc_len=length(xc);
    xc=[xc(1:xc_len-ht_len+1),xc(xc_len-ht_len+2:xc_len)+y(i,1:ht_len-1),
        y(i,ht_len:conv_len)];
end%for

if debug==2
    figure;
    if partes>0
        subplot(2,1,1);
        plot(20*log10(abs(Htf(1,:))));
        title('Hc(f) teorica');
        ylabel('dB');
        hold;
    end
    if partes>1
        for i=2:partes
            plot(20*log10(abs(Htf(i,:))));
        end%for
    end%if partes>1
    if partes>0
        subplot(2,1,2);

```

```

        plot(miatan2(imag(Htf(1,:)),real(Htf(1,:))));
        title('Fase Hc(f)');
        hold;
    end
    if partes>1
        for i=2:partes
            plot(miatan2(imag(Htf(i,:)),real(Htf(i,:))));
        end%for
    end%if partes>1

end%if debug
%*****
% Fin funci'on Conv_Channel_PLC07C()
%*****

```

E.3.4. Código para la simulación del ruido impulsivo

```

%*****
%*****
function bi=impn2(imp_s,n,pi,Z,powertype)
%*****
%*****
%funci'on impn2() genera una secuencia de ruido impulsivo para las
% muestras de un s'imbolo.
%
% imp_s: n'umero promedio de impulsos de ruido por s'imbolo OFDM
% n: n'umero de muestras por s'imbolo OFDM
% pi: potencia promedio del ruido impulsivo
% Z: es la impedancia de carga
% powertype: cadena que determina las unidades de pi. Ver wgn() de matlab
%Regresa en b la secuencia de valores de ruido.
%*****
%Programador: Gerardo Laguna
%Fecha:4/09/09
%*****
%*****
%Par'ámetros del ruido impulsivo a escala de s'imbolos OFDM:
%*****
P=100; %Tama'no de arreglo para generar aleatorios de Poisson
A=(n/imp_s); %tiempo de arribo de pulsos en muestras
z=Poisson_interarrivals(P,A); %se genera vector de tiempos de inter-arribo

T=1; %Periodo de observaci'on (en muestras)
Td=A; %se inicializa tiempo de evento diferido (puede ser cualquier tiempo)
for i=1:n %Se genera secuencia de Poisson para s'imbolo OFDM
    [Km,Td]=Poisson_N_arrivals2(z,P,T,Td);
    K(1,i)=Km;%secuencia para variable de Poisson
end
%*****
%*****
%genera ruido impulsivo (de acuerdo a la literatura [Kay 2006]):
bi=sqrt(K).*wgn(1,n,pi/A,Z,'linear');

%*****
% Fin function bi=impn2(imp_s,n,pi,Z,powertype)
%*****

%*****
%*****
function z=Poisson_interarrivals(l,lambda)
%*****
%*****
% Km=Poisson_interarrivals()
% Obtiene el vector z de longitud l con los tiempos de
% 'inter arribo con media lambda.

```

```

%
%*****
%Programador: Gerardo Laguna
%Fecha: 18/05/09
%*****
%rand('state',0); %restablece generador aleatorio.
%Se habilita cuando se desea ver lo mismo en todas las corridas.
for(i=1:len)
    z(1,i)=(lambda)*log(1/(1-rand(1,1)));
end
%*****
%Fin funci'on Poisson_interarrivals()
%*****

%*****
%*****
function [Km,Td]=Poisson_N_arrivals2(z,len,T,Td)
%*****
%      Km=Poisson_N_arrivals2()
% Obtiene el n'umero de arriivos en el intervalo T (normalmente tiempo de muestreo)
% dado un vector z de longitud len con los tiempos de 'inter arribo.
% Td es el tiempo que resta para el arribo de un evento diferido
% Nota: el vector z es de dimensi'on lxl len
%
%*****
%*****
%Programador: Gerardo Laguna
%Fecha:04/09/09
%*****

i=floor(rand(1,1)*len)+1;
Td=Td-T; %se decrementa tiempo de evento diferido
if Td<T
    Km=1;
    t=Td;
else
    Km=0;
    t=0;
end
s=0; %Salir= NO SALIR
while s==0 %mientras no se encuentre break
    t=t+z(1,i); %sumamos el tiempo de 'inter arribo
    if t>T %si el tiempo rebasa el tiempo de observaci'on:
        d=t-T; %se obtiene tiempo para posible evento diferido
        if Td<=0 Td=d; %Si tiempo diferido ya expiro:
            %actualizamos con nuevo evento diferido
        end
        break % terminamos
    end
    Km=Km+1; % si no se rebasa el tiempo de observaci'on: contabilizamos el arribo
    i=i+1;
    if i==(len+1)
        i=1;
    end
end
end
%*****
%Fin funci'on Poisson_N_arrivals2()
%*****

```

E.3.5. Código para la simulación del transreceptor con modulación OFDM

```

%*****
%*****
function [a,utiles]=DMT_map_bits(bits,qamsimbolos,ss,mask)

```

```

%*****
%*****
%funcion DMT_map_bits() mapea los bits de acuerdo con
%la mascara para subportadoras dada en mask.
%bits: secuencia de bits calculada por mapped_bit_count()
%quamsimbolos: n'umero de s'imbolos QAM de la se'nal anal'itica
%ss tama'no de s'imbolo QAM en bits
% mask es un vector de 1s y 0s para habilitar/deshabilitar las subportadoras
%Regresa en a la secuencia de datos por transmitir que corresponde a la
%
%      mascara con bits 'utiles transmitidos.
utiles n'umero de bits 'utiles transmitidos.
%*****
%
%Programador: Gerardo Laguna
%Fecha:13/08/09
%*****
b=zeros(qamsimbolos,ss);
utiles=0;
si=1; %source index
for i=1:qamsimbolos
    if mask(i)==1 %si la mascara es de habilitaci'on:
        b(i,:)=bits(si:si+ss-1); %datos para subportadora habilitada
        si=si+ss;%actualizamos 'indice
        utiles=utiles+ss;% se trata de bits 'utiles
    end
end
a=reshape(b,1,qamsimbolos*ss);
%*****
%Fin funcion DMT_map_bits()
%*****

%*****
%*****
function a=DMT_unmap_bits(b,ubits,qamsimbolos,ss,mask)
%*****
%*****
%funcion DMT_unmap_bits() des-mapea los bits de acuerdo con
%la mascara para subportadoras dada en mask.
%b: secuencia de bits obtenida de s'imbolo OFDM
%ubits: bits 'utiles dentro de secuencia b
%quamsimbolos: n'umero de s'imbolos QAM de la se'nal anal'itica
%ss: tama'no de s'imbolo QAM en bits
% mask: es un vector de 1s y 0s para habilitar/deshabilitar las subportadoras
%Regresa en a la secuencia de datos 'utiles que corresponde a la mascara.
%
%
%*****
%
%Programador: Gerardo Laguna
%Fecha:14/08/09
%*****
q=reshape(b,qamsimbolos,ss);
di=1; %destination index
a=zeros(1,ubits);
for i=1:qamsimbolos
    if mask(i)==1 %si la mascara es de habilitaci'on:
        a(di:di+ss-1)=q(i,:); %datos para subportadora habilitada
        di=di+ss;%actualizamos 'indice
    end
end
%*****
%Fin funcion DMT_unmap_bits()
%*****

%*****
%*****

```

```

function [xm,Xa]=fofdmqamt02(a,qamsimbolos,ss)
%*****
%*****
%funcion ofdmqamt02() par modulacion OFDM
%
%*****
%Programador: Gerardo Laguna
%Fecha:27/03/09
%*****
%a : secuencia de bits (1 y -1) de datos
% (el numero de bits n debe ser un multiplo de ss)
%simbolos: Numero de simbolos QAM de la se'nal analitica
%ss tama'no de simbolo QAM en bits
%puede generar los datos en forma aleatoria como sigue:
%a=rand(1,n);
%a=sign(a-0.5);
%en xm regresa la se'nal a transmitir
%en Xa regresa la se'nal analitica
[m,n]=size(a); %se asume un arreglo de lxn
simbolos=qamsimbolos;
b=reshape(a,simbolos,ss);
if ss==4
% Genera simbolos 16QAM:
XaT=2*b(:,1)+b(:,2)+j*(2*b(:,3)+b(:,4));
else
if ss==2
% Genera simbolos 4QAM:
XaT=b(:,1)+j*(b(:,2));
else
%Genera n simbolos BPSK
simbolos=n;
XaT=a.';
end
end
Xa=XaT.';%se'nal analitica
X=[0 Xa 0 conj(Xa(simbolos:-1:1))]; %se'nal analitica con simetr'ia Hermitiana

xm=real(iff(X));%se'nal real (la parte imaginaria es pr'acticamente cero)

%*****
%Fin funcion ofdmqamt02()
%*****

%*****
%*****
function Ym=fofdmqamrx(xm,qamsimbolos,N)
%*****
%*****
%funcion ofdmqamrx() para demodulacion OFDM
%*****
%Programador: Gerardo Laguna
%Fecha:27/03/09
%*****

%xm : secuencia de muestras temporales del simbolo OFDM
%simbolos: Numero de simbolos QAM de la se'nal analitica
%N: Tama'no de la FFT para construir la se'nal real

%en Ym regresa la se'nal analitica recuperada

YY=fft(xm(1:N));
Ym=YY(2:qamsimbolos+1); %se recupera se'nal analitica de se'nal con simetr'ia Hermitiana
%*****
%Fin funcion ofdmqamrx()
%*****

```

```

%*****
%*****
function y=detect_bits(n,Y,ss)
%*****
%*****
%Detecta secuencia de n bits de vector (complejo) Y con ss bits codificados
%por cada valor complejo.
%
%*****
%Programador: Gerardo Laguna
%Fecha: 20/05/09
%*****
simbolos=n/ss;
if ss==4 %16QAM
    y=zeros(simbolos,ss);
    for i=1:simbolos
        if real(Y(i))<-2 %caso 1 real
            y(i,1)=-1;
            y(i,2)=-1;
        else
            if real(Y(i))<0 %caso 2 real
                y(i,1)=-1;
                y(i,2)=1;
            else
                if real(Y(i))<2 %caso 3 real
                    y(i,1)=1;
                    y(i,2)=-1;
                else%caso 4 real
                    y(i,1)=1;
                    y(i,2)=1;
                end%caso 3 real
            end%caso 2 real
        end%caso1 real

        if imag(Y(i))<-2%caso1 imag
            y(i,3)=-1;
            y(i,4)=-1;
        else
            if imag(Y(i))<0%caso2 imag
                y(i,3)=-1;
                y(i,4)=1;
            else
                if imag(Y(i))<2 %caso3 imag
                    y(i,3)=1;
                    y(i,4)=-1;
                else
                    %caso 4 imag
                    y(i,3)=1;
                    y(i,4)=1;
                end%caso 3 imag
            end%caso 2 imag
        end%caso 1 imag
    end %end for
    y=reshape(y,1,n);
else %no es 16QAM
    if ss==2 %4QAM
        y=zeros(simbolos,ss);
        for i=1:simbolos
            if real(Y(i))<0 y(i,1)=-1;
            else
                y(i,1)=1;
            end
            if imag(Y(i))<0 y(i,2)=-1;
            else
                y(i,2)=1;
            end
        end
    end
end

```

```

        end
    end%for
    y=reshape(y,1,n);

    else %se asume BPSK
        y=zeros(1,n);
        for i=1:n
            if real(Y(i))<0 y(1,i)=-1;
            else
                y(i)=1;
            end
        end%for
    end%Si 4QAM otro
end%Si 16QAM otro
%*****
%Fin funci'on detect_bits()
%*****

%*****
%*****
function [y,Yd]=detect_bits03(n,Y,ss,mask)
%*****
%*****
%function y=detect_bits03() detecta secuencia de n bits considerando mascara.
%El vector complejo Y contiene a la se'nal anal'itica.
% En ss bits por s'imbolo de modulaci'on QAM.
% mask es un vector de 1s y 0s para habilitar/deshabilitar las
% subportadoras.
%Regresa:
% En y la cadena de bits correspondiente
% En Yd los valores de la se'nal anal'itica
%*****
%*****
%Programador: Gerardo Laguna
%Fecha: 20/08/09
%*****
if ss==4 | ss==2 | ss==1 %valores validos para ss
    simbolos=n/ss;
    validss=ss;
else % o por defecto:
    simbolos=n;
    validss=1;
end

if validss==4 %16QAM
    y=zeros(simbolos,validss);
    for i=1:simbolos
        if real(Y(i))<-2 %caso 1 real
            y(i,1)=-1;
            y(i,2)=-1;
            Yd(i)=-3;
        else
            if real(Y(i))<0 %caso 2 real
                y(i,1)=-1;
                y(i,2)=1;
                Yd(i)=-1;
            else
                if real(Y(i))<2 %caso 3 real
                    y(i,1)=1;
                    y(i,2)=-1;
                    Yd(i)=1;
                else%caso 4 real
                    y(i,1)=1;
                    y(i,2)=1;
                    Yd(i)=3;
                end
            end
        end
    end
end

```

```

        end%caso 3 real
    end%caso 2 real
end%caso1 real

if imag(Y(i))<-2%caso1 imag
    y(i,3)=-1;
    y(i,4)=-1;
    Yd(i)=Yd(i)-3j;
else
    if imag(Y(i))<0%caso2 imag
        y(i,3)=-1;
        y(i,4)=1;
        Yd(i)=Yd(i)-1j;
    else
        if imag(Y(i))<2 %caso3 imag
            y(i,3)=1;
            y(i,4)=-1;
            Yd(i)=Yd(i)+j;
        else %caso 4 imag
            y(i,3)=1;
            y(i,4)=1;
            Yd(i)=Yd(i)+3j;
        end%caso 3 imag
    end%caso 2 imag
end%caso 1 imag
end %end for
y=reshape(y,1,n);
else %no es 16QAM
    if validss==2 %4QAM
        y=zeros(simbolos,validss);
        for i=1:simbolos
            if real(Y(i))<0
                y(i,1)=-1;
                Yd(i)=-1;
            else
                y(i,1)=1;
                Yd(i)=1;
            end
            if imag(Y(i))<0
                y(i,2)=-1;
                Yd(i)=Yd(i)-j;
            else
                y(i,2)=1;
                Yd(i)=Yd(i)+j;
            end
        end%for
        y=reshape(y,1,n);
    else %se asume BPSK
        y=zeros(1,simbolos);
        for i=1:simbolos
            if real(Y(i))<0
                y(1,i)=-1;
                Yd(i)=-1;
            else
                y(1,i)=1;
                Yd(i)=1;
            end
        end%for
    end%Si 4QAM otro
end%Si 16QAM otro

%Enmascaramos resultado seg'un mask:
b=reshape(y,simbolos,validss);
ceros=zeros(1,validss);

```

```

for i=1:simbolos
    if mask(i)==0 %si la mascara es de des-habilitaci'on:
        b(i,:)=ceros; %enmascara subportadora
    end
end
y=reshape(b,1,n);
%*****
%Fin funci'on detect_bits03()
%*****

```

E.3.6. Código para la estimación del canal

```

%*****
%*****
function H_e=ChannelEstimation(H_en,qamsimbolos,L_spread,X_analitic,thresh,
                               wavfam,wavpar,wavlev)
%*****
%*****
%function ChannelEstimation() para estimar los coeficientes del canal
%subyacentes en una observaci'on ruidosa.
%
%*****
% H_en la observaci'on de la respuesta en frecuencia
% qamsimbolos: n'umero de s'imbolos de modulaci'on QAM por s'imbolo OFDM
% L_spread: muestras 'utiles de la respuesta al impulso
% X_analitic: Si se usa opci'on 2 o 3, se'nal anal'itica con patr'on de referencia
%             (s'imbolo piloto o trasmitido)
% thresh: seg'un el valor realiza limpieza del estimado de la respuesta en
%         frecuencia:
%         1: por ventaneo de respuesta al impulso en el dominio temporal
%         2: Estimaci'on LMMSE
%         3: Estimaci'on LMMSE simplificada
%         4: por umbralado duro en dominio ondeleta
%             partiendo de la resp. al impulso.
%         5: por umbralado suave en dominio ondeleta
%             partiendo de la resp. al impulso.
%         6: por umbralado suave D&J en dominio ondeleta
%             partiendo de la resp. al impulso.
%         7: por umbralado bayesiano en dominio ondeleta
%             partiendo de la resp. al impulso.
%         8: por umbralado bayesiano en cada nivel de descomposici'on
%             de la resp. al impulso
%         9: por umbralado duro en dominio ondeleta
%             partiendo de la resp. en frecuencia.
%         10: por umbralado suave en dominio ondeleta
%             partiendo de la resp. en frecuencia.
%         11: por umbralado suave D&J en dominio ondeleta
%             partiendo de la resp. en frecuencia.
%         12: por umbralado bayesiano en dominio ondeleta
%             partiendo de la resp. en frecuencia.
%         13: por umbralado bayesiano en cada nivel de descomposici'on
%             de la resp. en frecuencia
%         otro: sin limpieza alguna (Estimacion LS).
% Par'ámetros wavelet: wavfam,wavpar,wavlev (se emplea biblioteca WaveLab850)
%Regresa:
% En H_e la respuesta en frecuencia estimada
%*****
%*****
%Programador: Gerardo Laguna
%Fecha:28/04/10
%*****
%*****
% Eliminaci'on de ruido de H
%*****
%Primeramente, nos cambiamos al dominio temporal:

```

```

h_tn=ifft(H_en);

%*****
% Eliminacion de ruido en el dominio temporal
%*****
h_len=length(h_tn);
%eliminaci'on de ruido por supuesto de longitud de dispersi'on m'axima:
h_t_u=zeros(1,h_len);
%primera parte 'util:
h_t_u(1:L_spread)=h_tn(1:L_spread);
%'ultima parte 'util:
h_t_u(h_len-L_spread:end)=h_tn(h_len-L_spread:end);
%Determinaci'on de ruido en dominio temporal por supuesto de longitud de
%dispersi'on m'axima:
nt=h_tn(L_spread+1:h_len-L_spread-1);
%
%Calculo de las desv. estd. en cada dominio con base a la potencia del ruido:
P_nt=CalculaPotencia(nt,1);%respecto de ZL=1 porque esto lo ocupamos para calcular
%la desv. std.
desv_nt=sqrt(P_nt);%calculamos la desv. std del ruido en el dominio temporal
P_nf=P_nt*qamsimbolos;% por principio de conservaci'on de la energ'ia de Parseval
%fft<->ifft
desv_nf=sqrt(P_nf);%calculamos la desv. std del ruido en el dominio de la
%frecuencia
% calculo del umbral universal (Donoho & Johnstone):
n_samples=length(nt);
lambda_t=desv_nt*sqrt(2*log10(n_samples));
lambda_f=desv_nf*sqrt(2*log10(n_samples));
%*****
% Retorno al dominio de la frecuencia
%*****
H_e_u=fft(h_t_u);

%*****
% Eliminaci'on de ruido en el dominio de ondeleta
%*****
qmfm=MakeONFilter(wavfam,wavpar);% ondeleta con wavpar/2
%Nivel real de Descomposici'on: L=segmentos=log2(n)-wavlev=log2(qamsimbolos)-wavlev
Lev=log2(qamsimbolos)-wavlev;
%Longitud de los coeficientes de aproximacion: length(aprox)=length(orig)/2^L:
aprox_len=length(H_en)/2^(Lev);

%Pasamos del dominio temporal al dominio de la ondeleta:
h_w = FWT_PO(h_tn,wavlev,qmfm);
h_a=h_w(1:aprox_len);%aproximaci'on
h_d=h_w(aprox_len+1:end);%todo el detalle

%Pasamos del dominio en frecuencia al dominio de la ondeleta:
H_w = FWT_PO(H_en,wavlev,qmfm);
H_a=H_w(1:aprox_len);
H_d=H_w(aprox_len+1:end);

%*****
%Procesamiento seg'un thresh:
%*****
%Por defecto:(thresh<1 o thresh>13) sin eliminaci'on de ruido
% (estimaci'on en bruto o LS)
H_e=H_en;
if thresh==1 %con eliminaci'on de ruido por ventana temporal
H_e=H_e_u;
else % (1)
if thresh==2 % estimaci'on LMMSE
[H_e,C,Rcc]=LMMSE_CE(H_en,P_nf,X_analitic);
else % (2)

```

```

if thresh==3 % estimaci'on LMMSE simplificada
    [H_e,C,Rcc]=LMMSE_CE_fast(H_en,P_nf,X_analitic);
else%(3)
    if thresh==4 %con umbralado duro de la resp. al imp.
        %Umbralado de los detalles:
        h_d_ht=HardThresholding(h_d,lambda_t);%umbralado duro
        %Descomposici'on despu'es de umbralado:
        h_w_ht=[h_a,h_d_ht];
        %Pasamos del dominio de la ondeleta al dominio en frecuencia:
        h_t_hs = IWT_PO(h_w_ht,wavlev,qmf);
        H_e=fft(h_t_hs);
    else%(4)
        if thresh==5 % con umbralado suave de resp. al imp.
            %Umbralado de los detalles:
            h_d_st=SoftThresholding(h_d,lambda_t);%umbralado suave
            %Descomposici'on despu'es de umbralado:
            h_w_st=[h_a,h_d_st];
            %Pasamos del dominio de la ondeleta al dominio en frecuencia:
            h_t_st = IWT_PO(h_w_st,wavlev,qmf);
            H_e=fft(h_t_st);
        else%(5)
            if thresh==6 % con umbralado suave D&J de resp. al imp.
                %Umbralado de los detalles:
                h_d_ss=SoftThresholding_D_J(h_d,lambda_t);%umbralado suave
                %Descomposici'on despu'es de umbralado:
                h_w_ss=[h_a,h_d_ss];
                %Pasamos del dominio de la ondeleta al dominio en frecuencia:
                h_t_ss = IWT_PO(h_w_ss,wavlev,qmf);
                H_e=fft(h_t_ss);
            else %(6)
                if thresh==7 % con umbralado bayesiano de resp. al imp.
                    %Umbralado de los detalles:
                    %eliminaci'on de ruido por m'etodo bayesiano de Silverman,
                    h_d_bt=ebayesthresh(h_d,'laplace', 0.5, 0,[], 0, 'median');
                    %Descomposici'on despu'es de umbralado:
                    h_w_bt=[h_a,h_d_bt.'];
                    %Pasamos del dominio de la ondeleta al dominio en frecuencia:
                    h_t_bt = IWT_PO(h_w_bt,wavlev,qmf);
                    H_e=fft(h_t_bt);
                else%(7)
                    if thresh==8 % con umbralado bayesiano en cada nivel de descomp.
                        %de resp. al impulso.
                        %detalle al nivel L (NOTA: SE ASUME wavlev=4!!!):
                        detail_ini=aprox_len+1;
                        detail_len=aprox_len;
                        h_d5=h_w(detail_ini:detail_ini+detail_len-1);
                        %detalle al nivel L-1:
                        detail_ini=detail_ini+detail_len;
                        detail_len=2*detail_len;
                        h_d4=h_w(detail_ini:detail_ini+detail_len-1);
                        %detalle al nivel L-2:
                        detail_ini=detail_ini+detail_len;
                        detail_len=2*detail_len;
                        h_d3=h_w(detail_ini:detail_ini+detail_len-1);
                        %detalle al nivel L-3:
                        detail_ini=detail_ini+detail_len;
                        detail_len=2*detail_len;
                        h_d2=h_w(detail_ini:detail_ini+detail_len-1);
                        %detalle al nivel L-4:
                        detail_ini=detail_ini+detail_len;
                        detail_len=2*detail_len;
                        h_d1=h_w(detail_ini:detail_ini+detail_len-1);
                        %Umbralado de los detalles:
                        %eliminaci'on de ruido por m'etodo bayesiano de Silverman,
                        h_d5_bt=ebayesthresh(h_d5,'laplace', 0.5, 0,[], 0, 'median');
                    end
                end
            end
        end
    end
end

```

```

h_d4_bt=ebayesthresh(h_d4,'laplace', 0.5, 0,[], 0, 'median');
h_d3_bt=ebayesthresh(h_d3,'laplace', 0.5, 0,[], 0, 'median');
h_d2_bt=ebayesthresh(h_d2,'laplace', 0.5, 0,[], 0, 'median');
h_d1_bt=ebayesthresh(h_d1,'laplace', 0.5, 0,[], 0, 'median');
%Descomposici'on despu'es de umbralado:
h_w_bt=[h_a,h_d5_bt.',h_d4_bt.',h_d3_bt.',h_d2_bt.',h_d1_bt.'];
%Pasamos del dominio de la ondeleta al dominio en frecuencia:
h_t_bt = IWT_PO(h_w_bt,wavlev,qmf);
H_e=fft(h_t_bt);
else%(8)
if thresh==9 % con umbralado duro de resp. en frec.
%Umbralado de los detalles:
H_d_ht=HardThresholding(H_d,lambda_f);%umbralado duro
%Descomposici'on despu'es de umbralado:
H_w_ht=[H_a,H_d_ht];
%Pasamos del dominio de la ondeleta
%al dominio en frecuencia:
H_e = IWT_PO(H_w_ht,wavlev,qmf);
else%(9)
if thresh==10 % con umbralado suave de resp. en frec.
%Umbralado de los detalles:
H_d_st=SoftThresholding(H_d,lambda_f);%umbralado suave
%Descomposici'on despu'es de umbralado:
H_w_st=[H_a,H_d_st];
%Pasamos del dominio de la ondeleta
%al dominio en frecuencia:
H_e = IWT_PO(H_w_st,wavlev,qmf);
else%(10)
if thresh==11 % con umbralado suave D&J
%de resp. en frec.
%Umbralado de los detalles:
%umbralado suave,
H_d_ss=SoftThresholding_D_J(H_d,lambda_f);
%Descomposici'on despu'es de umbralado:
H_w_ss=[H_a,H_d_ss];
%Pasamos del dominio de la ondeleta
%al dominio en frecuencia:
H_e= IWT_PO(H_w_ss,wavlev,qmf);
else%(11)
if thresh==12 % con umbralado bayesiano
%de resp. en frec.
%Umbralado de los detalles:
%eliminaci'on de ruido por m'etodo
%bayesiano de Silverman,
H_d_bt=ebayesthresh(H_d,'laplace',
0.5,0,[],0,'median');
%Descomposici'on despu'es de umbralado:
H_w_bt=[H_a,H_d_bt.'];
%Pasamos del dominio de la ondeleta al
%dominio en frecuencia:
H_e= IWT_PO(H_w_bt,wavlev,qmf);
else%(11)
if thresh==13
%con umbralado bayesiano en cada nivel
%de descomp. de resp. en frec.
%detalle al nivel L
%(NOTA: SE ASUME wavlev=4!!!):
detail_ini=aprox_len+1;
detail_len=aprox_len;
H_d5=H_w(detail_ini:detail_ini+detail_len-1);
%detalle al nivel L-1:
detail_ini=detail_ini+detail_len;
detail_len=2*detail_len;
H_d4=H_w(detail_ini:detail_ini+detail_len-1);
%detalle al nivel L-2:

```

```

detail_ini=detail_ini+detail_len;
detail_len=2*detail_len;
H_d3=H_w(detail_ini:detail_ini+detail_len-1);
%detalle al nivel L-3:
detail_ini=detail_ini+detail_len;
detail_len=2*detail_len;
H_d2=H_w(detail_ini:detail_ini+detail_len-1);
%detalle al nivel L-4:
detail_ini=detail_ini+detail_len;
detail_len=2*detail_len;
H_d1=H_w(detail_ini:detail_ini+detail_len-1);
%Umbralado de los detalles:
%eliminaci'on de ruido por m'etodo
%bayesiano de Silverman,
H_d5_bt=ebayesthresh(H_d5,'laplace',0.5,
                    0,[],0,'median');
H_d4_bt=ebayesthresh(H_d4,'laplace',0.5,
                    0,[],0,'median');
H_d3_bt=ebayesthresh(H_d3,'laplace',0.5,
                    0,[],0,'median');
H_d2_bt=ebayesthresh(H_d2,'laplace',0.5,
                    0,[],0,'median');
H_d1_bt=ebayesthresh(H_d1,'laplace',0.5,
                    0,[],0,'median');
%Descomposici'on despu'es de umbralado:
H_w_bt=[H_a, H_d5_bt.',H_d4_bt.',
        H_d3_bt.',H_d2_bt.',H_d1_bt.'];
%Pasamos del dominio de la ondeleta
%al dominio en frecuencia:
H_e= IWT_PO(H_w_bt,wavlev,qmf);
end%(13)
end%(12)
end%(11)
end%(10)
end%(9)
end%(8)
end%(7)
end%(6)
end%(5)
end%(4)
end%(3)
end%(2)
end%(1)

```

E.3.7. Código para técnicas clásicas de estimación lineal

```

%*****
%*****
function [H_LMMSE,C,Rcc]=LMMSE_CE(H_LS,Pn,X)
%*****
%*****
%Rutina para obtener el vector con los coeficientes de atenuaci'on del canal
%con el criterio de LMMSE.
%
%H_LS: vector con datos de entrada (estimaci'on de canal con criterio de LS)
%Pn: es la potencia del ruido (varianza del ruido gaussiano)
%X: es el vector con los s'imbolos de modulaci'on transmitidos
%
%Regresa
%en H_LMMSE el vector de coeficientes estimados
%en C la matriz de datos para generar matriz de autocorrelaci'on Rcc
%en Rcc la matriz de autocorrelaci'on
%
%*****
%Programador: Gerardo Laguna

```

```

%Fecha:9/10/09
%*****
echo off
% Crea matriz de datos X para generar matriz de autocorrelaci'on R
[C,Rcc] = corrmtx(H_LS,length(H_LS)-1);
%Estimaci'on LMMSE:
H_LMMSE=(Rcc*inv(Rcc+Pn*inv(X*X'))*H_LS.').';
%*****
%Fin function [H_LMMSE,C,Rcc]=LMMSE_CE(H_LS,Pn,X)
%*****

%*****
%*****
function [H_LMMSE_f,C,Rcc]=LMMSE_CE_fast(H_LS,Pn,X)
%*****
%*****
%Rutina r'apida (simplificada) para obtener el vector con los coeficientes de
atenuaci'on del canal
%con el criterio de LMMSE.
%
%H_LS: vector con datos de entrada (estimaci'on de canal con criterio de LS)
%Pn: es la potencia del ruido (varianza del ruido gaussiano)
%X: es el vector con los s'imbolos de modulaci'on transmitidos
%
%Regresa
%en H_LMMSE_f el vector de coeficientes estimados
%en C la matriz de datos para generar matriz de autocorrelaci'on Rcc
%en Rcc la matriz de autocorrelaci'on
%
%*****
%Programador: Gerardo Laguna
%Fecha:9/10/09
%*****
echo off
% Crea matriz de datos X para generar matriz de autocorrelaci'on R
[C,Rcc] = corrmtx(H_LS,length(H_LS)-1);
% Estimaci'on LMMSE simplificada:
% Primero, se emplea carga unitaria para normalizar
beta=CalculaPotencia(X,1)*CalculaPotencia(1./X,1);
snr=CalculaPotencia(X,1)/Pn;
I=eye(size(Rcc));
%Finalmente, calculamos la estimaci'on:
H_LMMSE_f=(Rcc*inv(Rcc+(beta/snr)*I)*H_LS.').';

%*****
%Fin function [H_LMMSE_f,C,Rcc]=LMMSE_CE_fast(H_LS,Pn,X)
%*****

%*****
%*****
function [H_LMMSE,C,Rcc]=LMMSE_CE_SVD(p,H_LS,Pn,X)
%*****
%*****
%function [H_LMMSE,C,R]=LMMSE_CE_SVD()
%Rutina por descomposici'on de valores singulares (SVD) para obtener el vector
con los coeficientes de atenuaci'on del canal
%con el criterio de LMMSE.
%
%p: rango de la estimaci'on
%H_LS: vector con datos de entrada (estimaci'on de canal con criterio de LS)
%Pn: es la potencia del ruido (varianza del ruido gaussiano)
%X: es el vector con los s'imbolos de modulaci'on transmitidos
%
%Regresa

```

```

%en H_LMMSE_f el vector de coeficientes estimados
%en C la matriz de datos para generar matriz de autocorrelaci'on Rcc
%en Rcc la matriz de autocorrelaci'on
%
%*****
%Programador: Gerardo Laguna
%Fecha:15/10/09
%*****
echo off
% Crea matriz de datos X para generar matriz de autocorrelaci'on R
[C,Rcc] = corrmatrix(H_LS,length(H_LS)-1);
%Primero, se emplea carga unitaria para normalizar:
beta=CalculaPotencia(X,1)*CalculaPotencia(1./X,1);
snr=CalculaPotencia(X,1)/Pn;
[U,S,V] = svd(Rcc);
[m,n]=size(S);
%se asume que m=n:
Delta=zeros(m,n);
for i=1:p
    Delta(i,i)=S(i,i)/(S(i,i)+(beta/snr));
end
%Finalmente se calcula estimaci'on:
H_LMMSE=((U*Delta*V)*H_LS.').';

%*****
%Fin function [H_LMMSE,C,Rcc]=LMMSE_CE_SVD(p,H_LS,Pn,X)
%*****

```


Apéndice F

Estimación del espectro por el método de máxima entropía

El método de máxima entropía (MEM, por sus siglas en inglés) fue desarrollado por Burg [20, 21] para superar limitaciones fundamentales de los métodos soportados por la transformada de Fourier para la estimación del espectro en potencia de un proceso estocástico estacionario. La idea básica detrás del método MEM consiste en elegir particularmente al espectro que corresponde que corresponde a la serie de tiempo más aleatoria o más impredecible cuya función de autocorrelación concuerda con un conjunto de valores conocidos. Esto equivale a extrapolar la función de autocorrelación de la serie de tiempo disponible mediante la maximización de su contenido promedio de información (o Entropía, según la teoría de la información de Shannon). Así el método MEM da la vuelta al problema que surge con las funciones de ventaneo, lo cual es común en las técnicas de análisis del espectro soportadas por la transformada de Fourier.

F.1. Espectro de máxima entropía

Supóngase que son dados $2M + 1$ valores de la función de autocorrelación de un proceso estocástico estacionario $u(n)$ de media cero. Se desea obtener el valor específico del espectro en potencia que corresponde a la serie de tiempo más aleatoria cuya función de autocorrelación es consistente con el conjunto de los $2M + 1$ valores conocidos, lo cual se conoce como *principio de máxima entropía* dentro de la teoría de la información.

En el caso de un conjunto de variables aleatorias con distribución Gaussiana y media cero, la entropía está dada por [62]:

$$H = \frac{1}{2} \ln[\det(\mathbf{R})] \quad (\text{F.1})$$

donde \mathbf{R} es la matriz de correlación del proceso. De cualquier forma, cuando el proceso es de duración infinita, se tiene que la entropía H diverge y no puede ser usada como medida del contenido de la información. A fin de superar el problema de la divergencia mejor se emplea la *tasa de*

entropía definido por

$$\begin{aligned} h &= \lim_{M \rightarrow \infty} \frac{H}{M+1} \\ &= \lim_{M \rightarrow \infty} \frac{1}{2} \ln[\det(\mathbf{R})]^{1/(M+1)} \end{aligned} \quad (\text{F.2})$$

Sea $S(\omega)$ el espectro en potencia del proceso $u(n)$. La forma límite del determinante de la matriz de correlación \mathbf{R} se relaciona con el espectro en potencia $S(\omega)$ como sigue:

$$\lim_{M \rightarrow \infty} [\det(\mathbf{R})]^{1/(M+1)} = \exp \left\{ \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln[S(\omega)] d\omega \right\} \quad (\text{F.3})$$

donde sustituyendo (F.3) en (F.2) tenemos

$$h = \frac{1}{4\pi} \int_{-\pi}^{\pi} \ln[S(\omega)] d\omega \quad (\text{F.4})$$

Aunque esta relación fue derivada suponiendo que el proceso $u(n)$ es Gaussiano, la forma de la relación es válida para cualquier proceso. Se puede replantear el problema MEM en términos de la tasa de entropía. Se desea encontrar un espectro en potencia de valor real positivo mediante la tasa de entropía h si se satisfacen simultáneamente dos requerimientos:

1. La tasa de entropía h es *estacionaria* con respecto a los valores *no conocidos* de la función de auto-correlación del proceso.
2. El espectro en potencia es *consistente* con respecto a los valores *conocidos* de la función de auto-correlación del proceso.

Se verá cada uno de estos requerimientos en su oportunidad. Ya que la secuencia de auto-correlación $r(m)$ y espectro en potencia $S(\omega)$ de un proceso estacionario $u(n)$ forman una pareja de correspondencia de la transformada de Fourier de tiempo discreto, se puede escribir:

$$S(\omega) = \sum_{m=-\infty}^{\infty} r(m) \exp(-jm\omega) \quad (\text{F.5})$$

La ecuación (F.5) supone que el periodo de muestreo del proceso $u(n)$ es normalizado a la unidad. Sustituyendo (F.5) en (F.4) se tiene

$$h = \frac{1}{4\pi} \int_{-\pi}^{\pi} \ln \left[\sum_{m=-\infty}^{\infty} r(m) \exp(-jm\omega) \right] d\omega \quad (\text{F.6})$$

Se extrapola la secuencia de auto-correlación $r(m)$ fuera del rango de valores conocidos $-M \leq m \leq M$, eligiendo los valores no conocidos de la función de auto-correlación de tal manera que no se agrega información o entropía al proceso. Esto es, se impone la condición

$$\frac{\partial h}{\partial r(m)} = 0, \quad |m| \geq M+1 \quad (\text{F.7})$$

Así, diferenciando (F.6) con respecto de $r(m)$ y fijando el resultado a cero, se encuentra que las condiciones para maximizar la entropía son las siguientes:

$$\int_{-\pi}^{\pi} \frac{\exp(-jm\omega)}{S_{\text{MEM}}(\omega)} d\omega = 0, \quad |m| \geq M + 1 \quad (\text{F.8})$$

donde $S_{\text{MEM}}(\omega)$ es el valor especial del espectro en potencia que resulta de la condición (F.7). La expresión (F.8) implica que el espectro de la potencia $S_{\text{MEM}}(\omega)$ se puede expresar en la forma de una serie de Fourier truncada:

$$\frac{1}{S_{\text{MEM}}(\omega)} = \sum_{k=-M}^M c_k \exp(-jk\omega) \quad (\text{F.9})$$

Los coeficientes de Fourier complejos c_k de la expansión satisfacen la condición Hermitiana

$$c_k^* = c_{-k}$$

a fin de asegurar que $S_{\text{MEM}}(\omega)$ es real para todo ω .

El otro requerimiento es hacer consistente al espectro en potencia $S_{\text{MEM}}(\omega)$ con el conjunto de valores conocidos de la función de auto-correlación $r(m)$ para el intervalo $-M \leq m \leq M$. Ya que $r(m)$ debe ser igual a la transformada de Fourier inversa de tiempo discreto de $S_{\text{MEM}}(\omega)$ para $0 \leq m \leq M$, expresado como

$$r(m) = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_{\text{MEM}}(\omega) \exp(jm\omega) d\omega, \quad 0 \leq m \leq M \quad (\text{F.10})$$

Incluso, sustituyendo (F.9) en (F.10), se tiene

$$r(m) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{\exp(jm\omega)}{\sum_{k=-M}^M c_k \exp(-jk\omega)} d\omega, \quad 0 \leq m \leq M \quad (\text{F.11})$$

Claramente, en el conjunto de coeficientes de Fourier complejos c_k , se cuenta con los grados de libertad necesarios para satisfacer las condiciones de (F.11).

Continuando con el análisis, resulta conveniente emplear la notación de la transformada z mediante el cambio de variable de ω a z . Definiendo

$$z = \exp(j\omega)$$

entonces

$$d\omega = \frac{1}{j} \frac{dz}{z}$$

y podemos re-escribir (F.11) en términos de la variable z como una integral de contorno

$$r(m) = \frac{1}{j2\pi} \oint \frac{z^{m-1}}{\sum_{k=-M}^M c_k z^{-k}} dz, \quad 0 \leq m \leq M \quad (\text{F.12})$$

La integración de contorno en (F.12) se realiza sobre el círculo unitario en el plano z en sentido contrario a las manecillas del reloj. Ya que los coeficientes complejos c_k de Fourier satisfacen la condición Hermitiana, podemos expresar la sumatoria en el denominador de la integral en (F.12) como el producto de dos polinomios, como sigue:

$$\sum_{k=-M}^M c_k z^{-k} = G(z)G^*\left(\frac{1}{z^*}\right) \quad (\text{F.13})$$

donde

$$G(z) = \sum_{k=0}^M g_k z^{-k} \quad (\text{F.14})$$

y

$$G^*\left(\frac{1}{z^*}\right) = \sum_{k=0}^M g_k^* z^k \quad (\text{F.15})$$

Se elige el primer polinomio $G(z)$ para que sea de fase mínima, es decir que sus ceros están todos ubicados dentro del círculo unitario en el plano z . En forma complementaria, se elige el segundo polinomio $G^*(1/z^*)$ para que sea de fase máxima, es decir que sus ceros se encuentren ubicados fuera del círculo unitario en el plano z . Aún más, los ceros de estos dos polinomios son el inverso uno del otro con respecto al círculo unitario. Así, sustituyendo (F.13) en (F.12), se obtiene

$$r(m) = \frac{1}{j2\pi} \oint \frac{z^{m-1}}{G(z)G^*(1/z^*)} dz, \quad 0 \leq m \leq M \quad (\text{F.16})$$

A continuación se construye la siguiente sumatoria

$$\begin{aligned} \sum_{k=0}^M g_k r(m-k) &= \frac{1}{j2\pi} \oint \frac{z^{m-1} \sum_{k=0}^M g_k z^{-k}}{G(z)G^*(1/z^*)} dz \\ &= \frac{1}{j2\pi} \oint \frac{z^{m-1}}{G^*(1/z^*)} dz, \quad 0 \leq m \leq M \end{aligned} \quad (\text{F.17})$$

que se obtuvo partiendo de (F.16) y, luego, empleando (F.14).

Para evaluar la integral de contorno en (F.17), se emplea el *teorema del residuo de Cauchy* de la teoría de las variables complejas. De acuerdo con este teorema, la integral de contorno es igual a $2\pi j$ veces la suma de los *residuos* de los polos de la integral de $z^{m-1}/G^*(1/z^*)$ que caen dentro del círculo unitario empleado como contorno de integración. Ya que el polinomio $G^*(1/z^*)$ se elige para tener todos sus ceros fuera del círculo unitario, entonces la integral en (F.17) es analítica sobre y dentro del círculo unitario para $m \geq 1$. Para $m = 0$ la integral tiene sólo un polo en $z = 0$ con un *residuo* igual a $1/g_0^*$. Así, aplicando el teorema del residuo de Cauchy resulta en

$$\oint \frac{z^{m-1}}{G^*(1/z^*)} dz = \begin{cases} \frac{2\pi j}{g_0^*}, & m = 0 \\ 0, & m = 1, 2, \dots, M \end{cases} \quad (\text{F.18})$$

Entonces, sustituyendo (F.18) en (F.17)

$$\sum_{k=0}^M g_k r(m-k) = \begin{cases} \frac{1}{g_0^*}, & m = 0 \\ 0, & m = 1, 2, \dots, M \end{cases} \quad (\text{F.19})$$

Se puede identificar en el conjunto de $(M + 1)$ ecuaciones en (F.19) tienen similitud con forma matemática con las ecuaciones Wiener-Hopf para predicción hacia adelante de orden M , es decir [62]:

$$\sum_{k=0}^M a_{M,k} r(k-i) = \begin{cases} \mathbf{P}_M, & i = 0 \\ 0, & i = 1, 2, \dots, M \end{cases} \quad (\text{F.20})$$

En particular, comparando (F.19) con (F.20), se deduce que

$$g_k^* = \frac{1}{g_0 P_M} a_{M,k}, \quad 0 \leq k \leq M \quad (\text{F.21})$$

donde $a_{M,k}$, $k = 0, 1, \dots, M$ denota los pesos de las derivaciones del filtro transversal de predicción de error de orden M y \mathbf{P}_M es la potencia promedio del filtro. Ya que, por definición, $a_{M,0} = 1$ para toda M tenemos a partir de (F.21) que para $k = 0$:

$$|g_0|^2 = \frac{1}{P_M} \quad (\text{F.22})$$

Finalmente, sustituyendo (F.13), (F.20) y (F.22) en (F.9) con $z = \exp(j\omega)$, se obtiene

$$S_{\text{MEM}}(\omega) = \frac{P_M}{\left| 1 + \sum_{k=1}^M a_{M,k} \exp(-jk\omega) \right|^2} \quad (\text{F.23})$$

que se referirá como *espectro MEM*.

F.2. Algoritmo eficiente para calcular el espectro MEM

La fórmula (F.23) puede ser re-expresada alternativamente como

$$S_{\text{MEM}}(\omega) = \frac{1}{\sum_{k=-M}^M \psi(k) \exp(-jk\omega)} \quad (\text{F.24})$$

donde $\psi(k)$ se define en términos de los coeficientes del filtro de error de predicción como sigue:

$$\psi(k) = \begin{cases} \frac{1}{P_M} \sum_{i=0}^{M-k} a_{M,i} a_{M,i+k}^* & \text{para } k = 0, 1, \dots, M \\ \psi^*(-k) & \text{para } k = -M, \dots, -1 \end{cases} \quad (\text{F.25})$$

El parámetro $\psi(k)$ puede ser visto como una clase de correlación para los coeficientes del filtro de error de predicción.

Examinando el polinomio denominador en (F.24) se nota que éste representa la transformada discreta de Fourier de la secuencia $\psi(k)$. De acuerdo con esto, se puede emplear un algoritmo de transformada rápida de Fourier para un cálculo eficiente de dicho polinomio denominador y, por ello, del espectro MEM. Dada la secuencia de auto-correlación $r(0), r(1), \dots, r(M)$, perteneciente a un proceso estocástico $u(n)$, se puede resumir un algoritmo eficiente para calcular el espectro MEM como [62]:

1. **Recursión Levinson-Durbing.** Inicializar el algoritmo haciendo

$$a_{0,0} = 1$$

$$P_0 = r(0)$$

Para $m = 1, 2, \dots, M$, calcular

$$k_m = -\frac{1}{P_{m-1}} \sum_{i=0}^{M-1} r(i-m)a_{m-1,i}$$

$$a_{m,i} = \begin{cases} 1 & \text{para } i = 0 \\ a_{m-1,i} + k_m a_{m-1,m-i}^* & \text{para } i = 1, 2, \dots, m-1 \\ k_m & \text{para } i = m \end{cases}$$

$$P_m = P_{m-1}(1 - |k_m|^2)$$

2. **Correlación para coeficientes del filtro de error de predicción.** Calcular los coeficientes de correlación,

$$\psi(k) = \begin{cases} \frac{1}{P_M} \sum_{i=0}^{M-k} a_{M,i} a_{M,i+k}^* & \text{para } k = 0, 1, \dots, M \\ \psi^*(-k) & \text{para } k = -M, \dots, -1 \end{cases}$$

3. **Espectro MEM.** Emplear transformada rápida de Fourier para calcular el espectro MEM con

$$S_{\text{MEM}}(\omega) = \frac{1}{\sum_{k=-M}^M \psi(k) \exp(-jk\omega)}$$

F.3. Código en Matlab para calcular el espectro MEM

```
function [S,a,H,Pe]=MEM_Spectrum(x,p)
%*****
%Rutina para obtener el Espectro con el m'etodo de M'axima Entrop'ia.
%x: vector con datos de entrada
%p: orden del modelo AR
%Regresa
%en S el vector del Espectro AR
%en a el vector con los coeficientes del filtro
%en H el vector con los coeficientes de la funci'on de transferencia
%en Pe la potencia del error
%*****
%Programador: Gerardo Laguna
```

```

%Fecha:06/04/10
%*****
echo off
[a,Pm,X,R]=Levinson_Durbin(x,p);
a_conj=conj(a);
M=p+1;
for k=1:M
    suma=0;
    for i=1:M-k
        suma=suma+a(i)*a_conj(i+k);
    end
    psi(k)=suma/Pm;
end
psi_h=[conj(psi(M:-1:1)),psi];
S=1./fft(psi_h);
%Ahora, estimaci'on de la funci'on de transferencia:
Pe=Pm;
H=sqrt(S./Pm);

```

F.4. Código en Matlab para la recursión Levinson-Durbing

```

function [sol,alfa,X,R]=Levinson_Durbin(x,p)
%*****
%Rutina para obtener el vector de coeficientes AR mediante recursi'on Levinson-Durbin.
%x: vector con datos de entrada
%p: orden del modelo AR
%Regresa
%en sol el vector de coeficientes AR
%en alfa el valor con la potencia promedio de salida del filtro
%en X matriz de datos para generar matriz de auto-correlaci'on R
%en R matriz de auto-correlaci'on
%*****
%Programador: Gerardo Laguna
%Fecha:6/04/10
%*****
echo off
[X,R] = corrmtx(x,p);% Crea matriz de datos X para generar matriz de auto-correlacion R
M=p+1;
a=zeros(M,2);
a(1,1)=1;
r=R(:,1);
alfa= r(1);
c=1;
d=2;
for k=2:M
    a(k,c)=0;
    a(1,d)=1;
    gamma=0;
    for j=1:k
        gamma=gamma+r(k+1-j)*a(j,c);
    end
    beta=gamma/alfa;
    for j=1:k
        a(j,d)=a(j,c)-beta*conj(a(k-j+1,c));
    end
    alfa=alfa*(1-(abs(beta))^2);
    temp=c;
    c=d;
    d=temp;
end
sol=a(:,c);

```


Apéndice G

Algoritmo de optimización por cúmulo de partículas (PSO)

El algoritmo de optimización por cúmulo de partícula (PSO, por sus siglas en inglés) tiene su fundamento en el movimiento de partículas o individuos que “vuelan” en un espacio n -dimensional en busca del óptimo global de una manera cooperativa. En cada iteración del algoritmo, la posición de las partículas se actualiza con una regla simple, donde el movimiento del individuo, aunque esencialmente arbitrario, se ve influenciado por su propia experiencia (aprendizaje individual) y por el ambiente (influencia social) [79].

El algoritmo original fue propuesto por Kennedy y Heberhart [78], en 1995, y se enfocaba en la posición x y el cambio de la posición v (que fue llamado velocidad, por la analogía) para cada partícula. Más tarde, el algoritmo fue mejorado por Shi y Heberhart [130], en 1998, introduciendo el concepto de inercia w . Se denota pbx como la mejor aptitud encontrada hasta el momento por la partícula, y gbx a la mejor aptitud global encontrada hasta el momento dentro de la población. El algoritmo PSO típico, en su versión global, puede ser descrito como sigue:

Algoritmo 1:

1. **Inicialización de la población.** Inicializar cada partícula de la población en forma aleatoria, obteniendo los valores para los vectores n -dimensionales de posición x y velocidad v .
2. **Evaluación de aptitud.** Calcular la aptitud de la posición de cada partícula x_{yo} . Si la aptitud actual es mejor que pbx , entonces pbx es actualizado.
3. **Comparación.** Determinar la posición de la partícula con la aptitud más alta y actualizar en su caso gbx .
4. **Actualización.** Para cada dimensión, de cada partícula, la velocidad es actualizada según la ecuación siguiente:

$$v_{i,d}(t+1) = w \times v_{i,d}(t) + c_1 \times r_1 \times (pbx_{i,d} - x_{i,d}(t)) + c_2 \times r_2 \times (gbx_{i,d} - x_{i,d}(t))$$

donde c_1 y c_2 son constantes que ponderan la influencia del aprendizaje individual y la influencia social, respectivamente; r_1 y r_2 son variables aleatorias, entre 0 y 1, que representan el

libre movimiento de cada partícula, y w es la inercia del sistema calculado como

$$w = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} \times iter$$

donde w_{max} , w_{min} , $iter_{max}$, e $iter$ son los valores para la inercia máxima, inercia mínima, iteraciones máximas, e iteración actual, respectivamente.

5. Actualizar la posición x de cada partícula según la ecuación siguiente:

$$x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1)$$

6. Repetir los pasos 2-5 hasta alcanzar la condición de terminación (número de iteraciones o precisión).

Finalmente, en la realización práctica de PSO es necesario delimitar la velocidad de las partículas dentro del rango $[V_{min}, V_{max}]$ a fin de evitar un comportamiento “explosivo”.

Apéndice H

Artículos publicados y sometidos

H.1. Artículos en revistas indexadas

- **Science Citation Index Expanded:**

Laguna G., Olguín M., Cruz N., Barrón R. and Ivarez J.: Comparative Study of Parallel Variants for a Particle Swarm Optimization Algorithm Implemented on a Multithreading GPU. *Journal of Applied Research and Technology (JART)*. ISSN: 1665-6423. Vol.7 No.3, p.p. 292–309, 2009

H.2. Artículos en conferencias indexadas por IEEE

- Laguna G. and Barrón R.: Blind channel estimation for power-line communications by a pso-inspired algorithm. In *IEEE LATINCOM 2009*, Medellín, Colombia, Septiembre 2009.
- Laguna G. and Barrón R.: Survey on Indoor Power Line Communication Channel Modeling, *Proceedings IEEE Electronics, Robotics and Automotive Mechanics Conference 2008, CERMA 08*, México, 2008. pp. 163–168

H.3. Artículos en conferencias arbitradas

- Laguna G. y Barrón R.: Estudio Teórico Experimental para la Estimación de la Función de Transferencia de un Canal PLC, en memorias del Simposio de Metrología 2008, México, 2008.
- Laguna G. y Barrón R.: Circuito de acoplamiento para la comunicación por la línea eléctrica doméstica, en memorias del 9o. Congreso de Ingeniería Eléctrica y Electrónica Aplicada, CIEEA 2008, Instituto Tecnológico de Durango, pp. 1–4, Marzo, México, 2008.

H.4. Reportes técnicos

- Laguna G. y Barrón R.: Especificación de referencia para sistema de comunicación por la línea eléctrica (PLC). Serie Azul No. 242, Centro de Investigación en Computación, CIC-IPN, Zacatenco, México, D.F., 2009. ISBN 978-607-414-048-4

H.5. Artículos de divulgación

- Laguna G. y Barrón R.: Comunicación por la red eléctrica?: Un acercamiento a la tecnología BPL. Boletín UPIITA, No. 19, 30 de Noviembre del 2009. En línea <http://www.boletin.upiita.ipn.mx/>

H.6. Otras participaciones

- Cruz B., Barrón R., Sossa H. and Laguna G.: New Geometric Associative Memory Model with Application to Pattern Classification, Proceedings 3rd International Conference on Applications of Geometric Algebras in Computer Science and Engineering AGACSE 2008, Germany, 2008.
- Barrón R., Cruz B. and Laguna G.: Conformal Geometric Algebra for Spherical Convex Hull Optimization, Proceedings 3rd International Conference on Applications of Geometric Algebras in Computer Science and Engineering AGACSE 2008, Germany, 2008.

H.7. Artículos sometidos en revisión

- Laguna G., Barrón R. and Prieto A.: On Wavelet-based Channel Estimation for Power-line Communication Environments, sometido a Revista de la Facultad de Ingeniería de la Universidad de Antioquia.
- Laguna G., Barrón R. and Prieto A.: Wavelet-based Improvement for Channel Estimation in a Power-line Communication Environment Impaired by Impulsive Noise, sometido a Conferencia IEEE CERMA 2010.
- Laguna G., Olguín M., Cruz, N. and Barrón R.: Parallel Implementation of Particle Swarm Optimization and Differential Evolution Algorithms on a Multithreading GPU, sometido a Computers & Electrical Engineering de Elsevier.

Bibliografía

- [1] ABAD, J., BADENES, A., BLASCO, J., CARRERAS, J., DOMINGUEZ, V., GOMEZ, C., IRANZO, S., RIVEIRO, J., RUIZ, D., TORRES, L., AND COMABELLA, J. Extending the power line lan up to the neighborhood transformer. *IEEE Communications Magazine* 41 (Abril 2003), 64–70.
- [2] ABRAMOVICH, F., SAPATINAS, T., AND SILVERMAN, B. Wavelet thresholding via a bayesian approach. *Journal of the Royal Statistical Society B* 60, 4 (1998), 725–749.
- [3] AKHTMAN, J., AND HANZO, L. Decision directed channel estimation aided ofdm employing sample-spaced and fractionally-spaced cir estimators. *IEEE Transactions on Wireless Communications* 6, 4 (Abril 2007), 1171–1175.
- [4] ALIAS, M., CHEN, S., AND HANZO, L. Multiple-antenna-aided ofdm employing genetic-algorithm-assisted minimum bit error rate multiuser detection vehicular technology. *IEEE Transactions on Vehicular Technology* 54, 5 (Septiembre 2005), 1713–1721.
- [5] ANTONIADIS, A., JANSEN, M., JOHNSTONE, I. M., AND SILVERMAN, B. Ebayesthresh: Matlab software for empirical bayes thresholding. <http://www-lmc.imag.fr/lmc-sms/Anestis.Antoniadis/EBayesThresh>, 2004.
- [6] AYUB, E. Potencial de la red eléctrica como instrumento de conectividad social en méxico. *Investigación Tecnológica*, 6 (2005).
- [7] BABIC, M., BAUSCH, J., KISTNER, T., AND DOSTERT, K. Performance analysis of coded ofdm systems at statistically representative plc channels. In *IEEE International Symposium on Power Line Communications and Its Applications* (2006), pp. 104–109.
- [8] BABIC, M., HAGENAU, M., DOSTERT, K., AND BAUSCH, J. D4: Theoretical postulation of plc channel model. Open PLC European Research Alliance, 2005.
- [9] BAIG, S., AND GOHAR, N. D. A discrete multitone transceiver at the heart of the phy layer of an in-home power line communication local area network. *IEEE Communications Magazine* 41 (Abril 2003), 48–53.
- [10] BAIG, S., AND MUGHAL, M. J. A frequency domain equalizer in discrete wavelet packet multitone transceiver for in-home plc lans. In *IEEE International Symposium on Power Line Communications and Its Applications* (2007), pp. 181–186.

- [11] BANWELL, T. C., AND GALLI, S. A novel approach to the modeling of the indoor power line channel—part i: Circuit analysis and companion model. *IEEE Transactions on Power Delivery* 20, 2 (Abril 2005), 655–663.
- [12] BARMADA, S., MUSOLINO, A., AND RAUGI, M. Innovative model for time-varying power line communication channel response evaluation. *IEEE Journal on Selected Areas in Communications* 24, 7 (Julio 2006), 1317–1326.
- [13] BARRÓN, R., LÓPEZ, V., AND ESPINOSA, O. Algoritmos piramidales para procesamiento digital de seales. *Polibits: Revista de Computación, CIDeTeC,IPN IX-1*, 19 (1998), 29–32.
- [14] BASKAR, S., AND SUGANTHAN, P. A novel concurrent particle swarm optimization. In *IEEE Congress on Evolutionary Computation* (2004). 792–796.
- [15] BAYRAM, I., AND SELESNICK, I. W. A simple construction for the m-band dual-tree complex wavelet transform. In *IEEE Digital Signal Processing Workshop/Signal Processing Education Workshop* (2006), pp. 594–601.
- [16] BELAL, M., AND EL-GHAZAWI, P. Parallel models for particle swarm optimizers. *International Journal of Intelligent Computing and Information Sciences, IJICIS* 4, 1 (2004), 100–111.
- [17] BIGLIERI, E. Coding and modulation for a horrible channel. *IEEE Communications Magazine* 41 (Mayo 2003), 92–98.
- [18] BOLCSKEI, H., DUHAMEL, P., AND HLEISS, R. A systematic approach to detecting ofdm signals in a fading channel. *IEEE Transactions on Signal Processing* 49, 7 (Julio 2001), 1594–1598.
- [19] BULUMULLA, S., KASSAM, S., AND VENKATESH, S. A systematic approach to detecting ofdm signals in a fading channel. *IEEE Transactions on Communications* 48, 5 (Mayo 2000), 725–728.
- [20] BURG, J. Maximum entropy spectral analysis. In *Meeting of the Society of Exploration Geophysicists* (Oklahoma City, Okla., 1967).
- [21] BURG, J. *Maximum Entropy spectral analysis*. PhD thesis, Stanford University, Stanford, California, 1975.
- [22] BURT, P., AND ADELSON, E. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications* 31, 4 (Abril 1983), 532–540.
- [23] CAÑETE, F., CORTÉS, J., DIÉZ, L., ENTRAMBASAGUAS, J., AND CARMONA, J. Fundamentals of the cyclic short-time variation of indoor power-line channels. In *IEEE International Symposium on Power Line Communications and Its Applications* (2005), pp. 157–161.

- [24] CAÑETE, F. J., CORTÉS, J., DIÉZ, L., AND ENTRAMBASAGUAS, J. Modeling and evaluation of the indoor power line transmission medium. *IEEE Communications Magazine* 41 (Abril 2003), 41–47.
- [25] CAÑETE, F. J., CORTÉS, J., DIÉZ, L., AND ENTRAMBASAGUAS, J. Analysis of the cyclic short-term variation of indoor power line channels. *IEEE Journal on Selected Areas in Communications* 24, 7 (Julio 2006), 1327–1338.
- [26] CAI, X., AND AKANSU, A. A subspace method for blind channel identification in ofdm systems. In *IEEE International Conference on Communications* (2000), vol. 2, pp. 929–933.
- [27] CAMITI, CANIETI, AND FMD. Visión México 2020: Políticas públicas en materia de tecnologías de la información y comunicaciones para impulsar la competitividad de México. En línea: http://www.canieti.org/index.asp?_option_id=1118&_option_parent_id=0&_option_level=0, 2006.
- [28] CANTÚ-PAZ, E. *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer, 2000.
- [29] CAVERS, J. An analysis of pilot symbol assisted modulation for rayleigh fading channels. *IEEE Transactions on Vehicular Technology* 40, 4 (Noviembre 1991), 686–693.
- [30] CHANG, M., AND SU, Y. 2d regression channel estimation for equalizing ofdm signals. In *IEEE Vehicular Technology Conference* (Tokyo, Japan, Mayo 2000), vol. 1, pp. 240–244.
- [31] CHANG, R., AND GIBBY, R. A theoretical study of performance of an orthogonal multiplexing data transmission scheme. *IEEE Transactions on Communication Technology* 16, 4 (Agosto 1968), 529–540.
- [32] CHAUX, C. *Analyse en ondelettes M-bandes en arbre dual; application à la restauration d'images*. PhD thesis, Université de Marne-la-Vallée, Francia, 2006.
- [33] CHINI, A., YIYAN, W., EL-TANANY, M., AND MAHMOUD, S. Filtered decision feedback channel estimation for ofdm-based dtv terrestrial broadcasting system. *IEEE Transactions on Broadcasting* 44, 1 (Marzo 1998), 2–11.
- [34] CHOTIKAKAMTHORN, N., AND SUZUKI, H. On identifiability of ofdm blind channel estimation. In *IEEE Vehicular Technology Conference* (Amsterdam, Netherlands, Mayo 1999), vol. 4, pp. 2358–2361.
- [35] COHEN, A., DAUBECHIES, I., AND FEAUVEAU, J.-C. Biorthogonal bases of compactly supported wavelets. *Communications ACM* 45 (1992), 485–560.
- [36] COIFMAN, R. R., MEYER, Y., AND WICKERHAUSER, M. V. Wavelet analysis and signal processing. *Wavelets and their applications* (1992), 153–178.
- [37] COLERI, S., ERGEN, M., PURI, A., AND BAHAI, A. Channel estimation techniques based on pilot arrangement in ofdm systems. *IEEE Transactions on Broadcasting* 48, 3 (Septiembre 2002), 223–229.

- [38] CORTÉS, J., CAÑETE, F., AND DIÉZ, L. ENTRAMBASAGUAS, J. Characterization of the cyclic short-time variation of indoor power-line channels response. In *IEEE International Symposium on Power Line Communications and Its Applications* (2005), pp. 326 – 330.
- [39] CORTÉS, J., TONELLO, A., AND DIÉZ, L. Comparative analysis of pilot-based channel estimators for dmt systems over indoor power-line channels. In *IEEE International Symposium on Power Line Communications and Its Applications* (2007), pp. 372–377.
- [40] CRUSSIÈRE, M. *Étude et optimisation de communications à haut-débit sur lignes d'énergie: exploitation de la combinaison OFDM/CDMA*. PhD thesis, Institut National des Sciences Appliquées de Rennes (INSA), Rennes, Francia, 2005.
- [41] DAI, H., AND POOR, V. Advanced signal processing for power line communications. *IEEE Communications Magazine* 41 (Mayo 2003), 100–107.
- [42] DAUBECHIES, I. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory* 36, 5 (1990), 961–1005.
- [43] DAUBECHIES, I. *Ten Lectures on Wavelets*. CBMS-NSF, SIAM Lecture Series, 1992.
- [44] DE CARVALHO, E., AND SLOCK, D. Cramer-rao bounds for semi-blind, blind and training sequence based channel estimation. In *IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications* (París, Francia, 1997), pp. 129–132.
- [45] DEMPSTER, A. P., LAIRD, N. M., AND RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. *Royal Statistical Society* 39 (1977), 1–38.
- [46] DONOHO, D. Unconditional bases are optimal bases for data compression and statistical estimation. *Applied and Computational Harmonic Analysis* 1, 1 (Diciembre 1993), 100–115.
- [47] DONOHO, D., AND JOHNSTONE, I. Ideal spatial adaptation via wavelet shrinkage. *Biometrika* 81, 3 (1994), 425–455.
- [48] DONOHO, D., AND JOHNSTONE, I. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association* 90 (1995), 1200–1224.
- [49] DONOHO, D., AND JOHNSTONE, I. Minimax estimation via wavelet shrinkage. *The Annals of Statistics* 26, 3 (1998), 879–921.
- [50] DONOHO, D., MALEKI, A., SHAHRAM, M., BUCKHEIT, J., CLERC, M., KALIFA, J. AND MALLAT, S., AND YU, T. Wavelab 850, home page. <http://www-stat.stanford.edu/~wavelab/>. Consultado el 14 de junio del 2010.
- [51] DORIGO, M., AND STÜTZLE, T. *The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances, Handbook of Metaheuristics*. Kluwer Academic Publishers, 2002, ch. 9, pp. 250–285.

- [52] DUFFIN, R., AND SCHAEFFER, A. A class of non-harmonic fourier series. *Transactions on American Mathematics Society* 72 (1952), 341–366.
- [53] EDFORS, O., SANDELL, M., VAN DE BEEK, J.-J., WILSON, S., AND BORJESSON, P. Ofdm channel estimation by singular value decomposition. *IEEE Transactions on Communications* 46, 7 (1998), 931–939.
- [54] EIBEN, A., AND SMITH, J. *Introduction to Evolutionary Computing*. Natural Computing Series. Springer, 2003.
- [55] ENGELS, V., AND ROHLING, H. Multilevel differential modulation techniques (64-dapsk) for multicarrier transmission systems. *European transactions on telecommunications and related technologies* 6, 6 (1995), 633–640.
- [56] ESMAILIAN, T., KSCHISCHANG, F., AND GULAK, P. An in-building power line channel simulator. In *IEEE International Symposium on Power Line Communications and Its Applications* (2000).
- [57] FRENGER, P., ARNE, N., AND SVENSSON, B. Decision-directed coherent detection in multicarrier systems on rayleigh fading channels. *IEEE Transactions on Vehicular Technology* 48, 2 (Marzo 1999), 490–498.
- [58] FRENGER, P., AND SVENSSON, A. A decision directed coherent detector for ofdm. In *IEEE Vehicular Technology Conference* (Mayo 1996), vol. 3, pp. 1584–1588.
- [59] GUO, D., WANG, X., AND CHEN, R. Wavelet-based sequential monte carlo blind receivers in fading channels with unknown channel statistics. *IEEE Transactions on Signal Processing* 52, 1 (Enero 2004), 227–239.
- [60] HARDING, S., AND BANZHAF, W. Fast genetic programming on gpus. In *European Conference on Genetic Programming* (2007), M. Ebner, Ed., vol. 4445 of *Lecture Notes in Computer Science*, pp. 90–101.
- [61] HÄRING, J., AND VINCK, H. Coding for impulsive noise channels. In *IEEE International Symposium on Power Line Communications and Its Applications* (2001), pp. 103–108.
- [62] HAYKIN, S. *Adaptive filter theory*, 3d ed. Prentice-Hall, 1996.
- [63] HE, H., CHENG, S., ZHANG, Y., AND NGUIMBIS, J. Analysis of reflection of signal transmitted in low-voltage powerline with complex wavelet. In *IEEE International Symposium on Power Line Communications and Its Applications* (2004), pp. 86–91.
- [64] HE, H., CHENG, S., ZHANG, Y., AND NGUIMBIS, J. Home network power-line communication signal processing based on wavelet packet analysis. *IEEE Transactions on Power Delivery* 20, 3 (Julio 2005), 1879–1885.

- [65] HEATH, R. W., J., AND GIANNAKIS, G. Exploiting input cyclostationarity for blind channel identification in ofdm systems. *IEEE Transactions on Signal Processing* 47, 3 (Marzo 1999), 848–856.
- [66] HO EHER, P. Tcm on frequency-selective land-mobile fading channels. In *5th Tirrenia International Workshop on Digital Communications* (1991), pp. 317–328.
- [67] HO EHER, P., KAISER, S., AND ROBERTSON, P. Two-dimensional pilot-symbol-aided channel estimation by wiener filtering. In *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Abril 1997), vol. 3, pp. 1845–1848.
- [68] HOLSCHNEIDER, M. *Wavelets, an analysis tool*. Oxford Science Publications, 1995.
- [69] HOMEPLUG. <http://www.homeplug.org/>.
- [70] HSIEH, M., AND WEI, C. Channel estimation for ofdm systems based on comb-type pilot arrangement in frequency selective fading channels. *IEEE Transactions on Consumer Electronics* 44, 1 (Febrero 1998), 217–225.
- [71] HUANG, A., AND ZHAO, Y. Estimating channel response from pilot subcarrier pairs for ofdm systems. In *Midwest Symposium on Circuits and Systems* (1997), vol. 2, pp. 774–777.
- [72] IZUMI, K., UMEHARA, D., AND DENNO, S. Performance evaluation of wavelet ofdm using ascet. In *IEEE International Symposium on Power Line Communications and Its Applications* (2007), pp. 246–251.
- [73] JAFROT, E. *Estimation de canal très sélectif en temps et en fréquence pour les systèmes OFDM*. PhD thesis, École Nationale Supérieure des Télécommunications (ENST-COMELEC), París, Francia, 2000.
- [74] JIANG, X., NGUIMBIS, J., CHENG, S., HE, H., AND WU, X. A novel scheme for low voltage powerline communication signal processing. *International Journal of Electrical Power & Energy Systems* 25, 4 (2003), 269–274.
- [75] JOHNSTONE, I. M. Wavelets and the theory of non-parametric function estimation. *Philosophical Transactions of the Royal Society A* 357, 1760 (Septiembre 1999), 2475–2493.
- [76] KAY, S. M. *Fundamentals of statistical signal processing: estimation theory*. Prentice Hall PTR, 1993.
- [77] KAY, S. M. *Intuitive probability and random processes using MATLAB*. Springer, 2006.
- [78] KENNEDY, J., AND EBERHART, R. Particle swarm optimization. In *IEEE Conference on Neural Networks* (1995), pp. 1942–1948.
- [79] KENNEDY, J., AND EBERHART, R. *Swarm Intelligence*. Morgan Kaufmann Academic Press, 2001.

- [80] KNUTH, D. *The Art of Computer Programming*, 2d. ed., vol. 2. Addison-Wesley, 1981.
- [81] LAGUNA, G., AND BARRÓN, R. Circuito de acoplamiento para la comunicación por la línea eléctrica doméstica. In *Congreso de Ingeniería Eléctrica y Electrónica Aplicada, CIEEA* (S.P., México, Marzo 2008), pp. 1–4.
- [82] LAGUNA, G., AND BARRÓN, R. Estudio teórico experimental para la estimación de la función de transferencia de un canal plc. In *Simposio de Metrología 2008* (Queretaro, México, Octubre 2008).
- [83] LAGUNA, G., AND BARRÓN, R. Survey on indoor power line communication channel modeling. In *IEEE Electronics, Robotics and Automotive Mechanics Conference, CERMA* (Cuernavaca, México, Septiembre 2008), pp. 163–168.
- [84] LAGUNA, G., AND BARRÓN, R. Blind channel estimation for power-line communications by a pso-inspired algorithm. In *IEEE LATINCOM* (Medellín, Colombia, Septiembre 2009).
- [85] LAGUNA, G., AND BARRÓN, R. Especificación de referencia para sistema de comunicación por la línea eléctrica (plc). Serie Azul 242, Centro de Investigación en Computación, CIC-IPN, Zacatenco, México, D.F., 2009. ISBN 978-607-414-048-4.
- [86] LAN, H., YANG, S., AND TAN, F. Semi-blind channel estimation algorithm based on wavelet de-noising in ofdm system. In *IEEE International Conference on Wavelet Analysis and Pattern Recognition, ICWAPR* (Beijing, China, 2007), vol. 1, pp. 179–182.
- [87] LELE, C., SIOHAN, P., LEGOUABLE, R., AND JAVAUDIN, J.-P. Preamble-based channel estimation techniques for ofdm/oqam over the powerline. In *IEEE International Symposium on Power Line Communications and Its Applications* (2007), pp. 59–64.
- [88] LI, C., AND ROY, S. Subspace-based blind channel estimation for ofdm by exploiting virtual carriers. *IEEE Transactions on Wireless Communications* 2, 1 (Enero 2003), 141 – 150.
- [89] LI, J., WANG, X., HE, R., AND CHI, Z. An efficient fine-grained parallel particle swarm optimization method based on gpu-acceleration. *International Journal of Innovative Computing, Information and Control* 3, 6(B) (2007), 1707.
- [90] LI, Y., CIMINI, L.J., J., AND SOLLENBERGER, N. Robust channel estimation for ofdm systems with rapid dispersive fading channels. *IEEE Transactions on Communications* 46, 7 (Julio 1998), 902 – 915.
- [91] LI, Y., SESHADRI, N., AND ARIYAVISITAKUL, S. An adaptive two-dimensional channel estimator for wireless ofdm with application to mobile dvb-t. *IEEE Transactions on Broadcasting* 17, 3 (Marzo 1999), 461–471.
- [92] LIN, R., AND PETROPULU, A. Linear precoding assisted blind channel estimation for ofdm systems. *IEEE Transactions on Vehicular Technology* 54, 3 (Mayo 2005), 983 – 995.

- [93] LIU, H., XU, G., TONG, L., AND KAILATH, T. Recent developments in blind channel equalization: From cyclostationarity to subspaces. *IEEE Transactions on Signal Processing* 50, 1-2 (Abril 1996), 83 – 99.
- [94] LIU, Z., AND YANG, D. M-band wavelet-based frequency-selective rayleigh-fading channel estimation. In *IEEE International Conference on Mobile Technology, Applications and Systems* (Guangzhou, Noviembre 2005), pp. 3–3.
- [95] MA, H., YE, C., AND S., Z. Research on parallel particle swarm optimization algorithm based on cultural evolution for the multi-level capacitated lot-sizing problem. In *IEEE Control and Decision Conference* (2008), pp. 965–970.
- [96] MALLAT, S. Multiresolution approximation and wavelet orthonormal bases of $l^2(\mathbb{R})$. *Transactions on American Mathematics Society* 315, 1 (Septiembre 1989), 69 – 87.
- [97] MALLAT, S. *A wavelet tour of signal processing*. Academic Press, San Diego, USA, 1998.
- [98] MENG, H., CHEN, S., GUAN, Y., LAW, C., SO, P., GUNAWAN, E., AND LIE, T. Modeling of transfer characteristics for the broadband power line communication channel. *IEEE Transactions on Power Delivery* 19, 3 (Julio 2004), 1057–1064.
- [99] MEYER, Y. *Ondelettes et algorithmes concurrents*. Hermann, París, Francia, 1992.
- [100] MEZURA-MONTES, E., VELÁZQUEZ-REYES, J., AND COELLO-COELLO, C. A comparative study of differential evolution variants for global optimization. In *GECCO* (2006), pp. 485–492.
- [101] MIGNONE, V., AND MORELLO, A. Cd3-ofdm: a novel demodulation scheme for fixed and mobile receivers. *IEEE Transactions on Communications* 44, 9 (Septiembre 1996), 1144–1151.
- [102] MOHER, M., AND LODGE, J. Tcmp-a modulation and coding strategy for rician fading channels. *IEEE Journal on Selected Areas in Communications* 7, 9 (Diciembre 1989), 1347 – 1355.
- [103] MOON, J. K., AND CHOI, S. Performance of channel estimation methods for ofdm systems in amultipath fading channels. *IEEE Transactions on Consumer Electronics* 46, 1 (Febrero 2000), 161 – 170.
- [104] MOULINES, E., DUHAMEL, P., CARDOSO, J.-F., AND MAYRARGUE, S. Subspace methods for the blind identification of multichannel fir filters. *IEEE Transactions on Signal Processing* 43, 2 (Febrero 1995), 516 – 525.
- [105] MUCK, M., DE COURVILLE, M., AND DUHAMEL, P. A pseudorandom postfix ofdm modulator - semi-blind channel estimation and equalization. *IEEE Transactions on Signal Processing* 54, 3 (Marzo 2006), 1005–1017.

- [106] MUQUET, B., AND DE COURVILLE, M. Blind and semi-blind channel identification methods using second order statistics for ofdm systems. In *IEEE International Conference on Acoustics, Speech, and Signal Processing* (1999), vol. 5, pp. 2745 – 2748.
- [107] MUQUET, B., DE COURVILLE, M., AND DUHAMEL, P. Subspace-based blind and semi-blind channel estimation for ofdm systems. *IEEE Transactions on Signal Processing* 50, 7 (Julio 2001), 1699–1712.
- [108] MZOUGH, A., LAFONTAINE, O., AND LITAIZE, D. Performance of the vectorial processor vecsm2* using serial multiport memory. In *International Conference on Supercomputing* (Toulouse Cedex, Francia, 1996), pp. 390–397.
- [109] NASON, G. *Wavelet Methods in Statistics with R*. Springer, 2008.
- [110] NECKER, M., AND STUBER, G. Totally blind channel estimation for ofdm on fast varying mobile radio channels. *IEEE Transactions on Wireless Communications* 3, 5 (Septiembre 2004), 1514–1525.
- [111] NVIDIA CORPORATION. *CUDA: Computer Unified Device Architecture Programming Guide, Version 2.0*. California, USA, 2009.
- [112] OÑEILL, M., AND BRABAZON, A. Self-organising swarm (soswarm). *Soft Computing* 12 (Enero 2008), 1073–1080.
- [113] ONIZAWA, T., MIZOGUCHI, M., SAKATA, T., AND MORIKURA, M. A simple adaptive channel estimation scheme for ofdm systems. In *IEEE Vehicular Technology Conference* (1999), vol. 1, pp. 279–283.
- [114] OPERA. <http://www.ist-opera.org/>.
- [115] OWENS, J., LUEBKE, D., GOVINDARAJU, N., HARRIS, M., KRÜGER, J., LEFOHN, A., AND PURCELL, T. A survey of general-purpose computation on graphics hardware. *Computer Graphics Forum* 26, 1 (2007), 80–113.
- [116] PAUL, C. *Analysis of multiconductor transmission lines*. John Wiley & Sons, 1998.
- [117] PAVLIDOU, N., HAN VINCK, A., YAZDANI, J., AND HONARY, B. Power line communications: State of the art and future trends. *IEEE Communications Magazine* 41 (Abril 2003), 34–40.
- [118] PAY, G., AND SAFAK, M. Performance of dmt systems under impulsive noise. In *IEEE International Symposium on Power Line Communications and Its Applications* (2001), pp. 109–114.
- [119] PHILIPPS, H. Modelling of powerline communication channels. In *IEEE International Symposium on Power Line Communications and Its Applications* (1999), pp. 14–21.
- [120] PROAKIS, J., AND SALEHI, M. *Digital Communications*, 5th ed. McGraw-Hill, 2008.

- [121] PROAKIS, J., SALEHI, M., AND BAUCH, G. *Contemporary Communications Systems Using MATLAB and Simulink*. Thomson Brooks/Cole, 2004.
- [122] RINNE, J., AND RENFORS, M. Pilot spacing in orthogonal frequency division multiplexing systems on practical channels. *IEEE Transactions on Consumer Electronics* 42, 4 (Noviembre 1996), 959–962.
- [123] S.A., T. La sociedad de la información en europa: Presente y perspectivas. España, 2002.
- [124] SADOUGH, S., ICHIR, M., DUHAMEL, P., AND JAFFROT, E. Ultra wideband ofdm channel estimation through a wavelet based em-map algorithm. *European Transactions on Telecommunications* 19 (Septiembre 2008), 761–771.
- [125] SANCHA, S., CAÑETE, F., AND DIÉZ, L. ENTRAMBASAGUAS, J. A channel simulator for indoor power-line communications. In *IEEE International Symposium on Power Line Communications and Its Applications* (2007).
- [126] SANZI, F., AND SPEIDEL, J. An adaptive two-dimensional channel estimator for wireless ofdm with application to mobile dvb-t. *IEEE Transactions on Broadcasting* 46, 2 (Junio 2000), 128–133.
- [127] SCHUTTE, J., REINBOLT, J., FREGLY, B., HAFTKA, R., AND GEORGE, A. Parallel global optimization with the particle swarm algorithm. *International Journal for Numerical Methods in Engineering* 61, 13 (2003), 1–24.
- [128] SELESNICK, I., BARANIUK, R., AND KINGSBURY, N. The dual-tree complex wavelet transform. *IEEE Signal Processing Magazine* 22, 6 (Noviembre 2005), 123–151.
- [129] SHEN, Y., AND MARTINEZ, E. *Channel Estimation in OFDM Systems*. Freescale Semiconductors, 2006. AN3059, on line at http://code.ucsd.edu/~yushen/publications_files/Yushi\%20Shen-Jan06-Freescale-OFDM.pdf.
- [130] SHI, Y., AND EBERHART, R. A modified particle swarm optimizer. In *IEEE International Conference on Evolutionary Computation* (Mayo 1998), pp. 69–73.
- [131] SILVERMAN, B. Empirical bayes thresholding: adapting to sparsity when it advantageous to do so. *Journal of the Korean Statistical Society* 36 (2007), 1–29.
- [132] STEFFEN, P., HELLER, P. N., GOPINATH, R. A., AND BURRUS, C. S. Theory of regular m-band wavelet bases. *IEEE Transactions on Signal Processing* 41, 12 (1993), 3497–3511.
- [133] TUCCI, M., RAUGI, M., MUSOLINO, A., AND BARMADA, S. Blind channel estimation for power-line communications by a kohonen neural network. In *IEEE International Symposium on Power Line Communications and Its Applications* (2007), pp. 35–40.
- [134] TUFVESSON, F., AND MASENG, T. Pilot assisted channel estimation for ofdm in mobile cellular systems. In *IEEE Vehicular Technology Conference* (Mayo 1997), vol. 3, pp. 1639 – 1643.

- [135] VAN DE BEEK, J.-J., EDFORS, O., SANDELL, M., WILSON, S., AND BORJESSON, P. On channel estimation in ofdm systems. In *IEEE Vehicular Technology Conference* (1995), vol. 2, pp. 815–819.
- [136] VIDA KOVIC, B. *Statistical Modeling by Wavelets*. Jhon Wiley & Sons, 1999.
- [137] WALKER, J. *A primer on wavelet and their scientific applications*. Chapman & Hall/CRC, 1999.
- [138] WANG, Z., AND GIANNAKIS, G. B. Linearly precoded or coded ofdm against wireless channel fades? In *IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications* (2001), pp. 267–271.
- [139] WEI, X., LIU, J., AND LI, Y. Simulation and analysis of power transmission line model. In *IEEE International Symposium on Power Line Communications and Its Applications* (2006), pp. 256–260.
- [140] XUE, W., L., Z., AND LI, J. Improved channel estimation using wavelet denoising for ofdm and ofdma systems. In *IEEE International Conference on Advanced Information Networking and Applications Workshops* (Los Alamitos, CA, USA, 2009), pp. 129–133.
- [141] ZAHAROV, V., AND CASCO, F. Discrete wavelet transform application to rayleigh fading channel estimation in radio communications. *Journal of Radioelectronics*, 10 (2002). en línea en http://jre.cplire.ru/iso/oct02/1/text_e.html.
- [142] ZHOU, S., AND GIANNAKIS, G. Finite-alphabet based channel estimation for ofdm and related multicarrier systems. *IEEE Transactions on Communications* 49, 8 (Agosto 2001), 1402–1414.
- [143] ZHOU, S., MUQUET, B., AND GIANNAKIS, G. Subspace-based (semi-) blind channel estimation for block precoded space-time ofdm. *IEEE Transactions on Signal Processing* 50, 5 (Mayo 2002), 1215–1228.
- [144] ZIMMERMANN, M., AND DOSTERT, K. A multi-path signal propagation model for the power line channel in the high frequency range. In *IEEE International Symposium on Power Line Communications and Its Applications* (1999), pp. 45–51.
- [145] ZIMMERMANN, M., AND DOSTERT, K. A multipath model for the powerline channel. *IEEE Transactions on Communications* 50, 4 (Abril 2002), 553–559.